The Islamic University Gaza
Higher Education Deanship
Faculty of Engineering
Computer Department

الجامعة الإسلامية – غزة
عمادة الدراسات العليا
كلية الهندسة
قسم هندسة الحاسوب

Secure Access for Web Educational Recourses across Multiple Domains

"نظام آمن لخدمات الويب التعليمية بين مختلف النطاقات"

Submitted by:

Eng. Osama N. Qunoo

Supervised by:

Prof. Hatem Hammad

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Computer Engineering

١٤٣٢ هـ-٢٠١١ م

المنارة للاستشارات
www.manaraa.com

Abstract

Identity federation is a technology which enables the identity information to be trustily transferred across autonomous security domains. Shibboleth Federation is considered to trust logging process between different web educational resources, Fully Global log-out is not addressed by Shibboleth. In this thesis, we address Fully Global Log-out with a cached version of content as an off-line content, and enforce user to re-login for the new request after global logout. The thesis modifies and utilizes the Shibboleth IdP source code to achieve securing model for web educational resources.

www.manaraa.com

ملخص الرسالة

إن أنظمة اتحاد الهويات هي أنظمة تستخدم تكنولوجيا تمكن من تنقل معلومات الهوية عبر النطاقات المختلفة . ويعتبر أحد هذه الأنظمة Shibboleth والذي يعتمد على عملية تسجيل موثوقة على شبكة الإنترنت بين الموارد والنطاقات المختلفة ، وحيث أن هذا النظام لم يعالج مبادئ الأمان في عمليات الخروج ولم يعد هناك ألية تضمن الخروج الكامل من جميع الموارد بشكل كامل ، سنقوم بطرح نظام تعليمي متعدد يرتبط بنظام ال Shibboleth مع تطبيق الخروج الكامل للموارد من خلال رابط واحد ، و إلزام المستخدم لإعادة تسجيل الدخول في حال أي طلب جديد بعد خروجه .

الأطروحة تعدل وتستخدم Shibboleth لتحقيق نموذج لتأمين الموارد التعليمية على شبكة الإنترنت.

النظام يتميز بواجهته عبر الويب وسهولة الإستخدام من حيث الوصول الي المصادر المحمية بكلمة مرور ، والسماح لمتابعة حالة المستخدم من حيث تواجده داخل النظام وإعلامه بالخدمات التي يتم الخروج منها بناء على طلب منه بالخروج الكامل من النظام.

المنارة للاستشارات
www.manaraa.com

Acknowledgement

This research thesis would not have been possible without the support of my supervisor, Prof. Dr. Hatem Hamad who was abundantly helpful and offered invaluable assistance, support and guidance.

Special thanks to my beloved family members, especially my parents, my wife, and my children for expressing their love, i fully express my gratitude to them; for their understanding and endless love, through the duration of my studies.

Osama N. Qunoo

www.manaraa.com

Table of Contents

المنارة للاستشارات

www.manaraa.com

www.manaraa.com

List of Figures

www.manaraa.com

List of Tables

www.manaraa.com

This page left intentionally blank

www.manaraa.com

Chapter 1 - Introduction

1.1 Thesis Statement

This thesis discusses the securing access to the web educational resources among different domains. The main idea is to provide more security to the logout process in shibboleth Identity federation system as a global logout with forcing to re-login when first request occurs after global logout. The open resources by the user will be cached and be viewed friendly to him, even he will be logout. This in turn will enable the clients to access the educational resources as a one system in logout. The implementations and results are accurate and were obtained by the author.

1.2 Background

1.2.1 Web Educational Resources

This fact, together with the widespread use of Internet, has led many institutions to grant access to educational resources through e-learning systems known as LMSs (Learning Management Systems). These systems deal with the administration, provision and control of educational resources and functionalities. Current LMSs can be considered as complex Web applications. Some of the best-known examples are Moodle (1), Blackboard (2), LRN (3) and Sakai (4). These systems typically provide a centralized environment to supply data (pdf documents, multimedia files, etc.) along with applications or tools to manipulate them.

www.manaraa.com

Resources such as libraries and student services must be confidential, especially if that used among multiple domains. Repeated identity authentications across different organizations not only cause the inconvenience to users, but also take up overfull system resources (5).

1.2.2 Federation System

Over the last few years, ideas for building federations have improved. The Communities are growing and in several countries national federations are established or are being deployed as in (6) (7) (8).

Federation Systems depends on Single Sign-On (SSO), such as Shibboleth users who wish to access services and consume resources, Service Providers (SPs) who offer services to the users and Identity Providers (IdPs), which store the information a user has to submit for the authentication in an SSO environment.

1.3 Research problem

The current research for Federations using the Single Sign-On in (9) (10) (7) (11) (12) have mostly been concentrating on the identity of the users through the core backbone of SSO with a logout vulnerability. Educational Resources in (4) (1) (5) have to be enabled to users in a more secure form with better choices to clients and being familiar to manipulate. The Federation of Educational Resources has to take into its consideration the requirements of SSO model as in (13) (14). This thesis provides the educational resources

www.manaraa.com

federation by shibboleth with more security in logout process which is designed to be meet the SSO requirements and educational requirements in (14).

1.4 Scope of Implementation

The ideas in this thesis can be applied for Universities who shared educational resources for their customers as students or staff. The implementation will provide easily and friendly Logout from the systems.

When the user opens a number of sessions to access confidential educational resources, he is looking forward to the final exit from all sessions by one click/button with a copy retained for the final content of those meetings, so this research is working to apply the final exit or global logout. The modification in Logout process will be in the IdP in Shibboleth Federation system which will be friendly and more secure.

www.manaraa.com

Chapter 2 – Preliminary Discussions

In this chapter, we review the basic concepts behind this thesis. Mainly speaking, we will discuss the concepts of Single Sign-on Models. For each concept, a quick and comprehensive review will be made. This includes the main architectures, the techniques, and finally the security requirements that distinguish their usage in practice.

2.1 Single Sign on Models

These four techniques are: OAuth (15), Delegation Permits (16), Shibboleth (17) and OpenSSO (18). All of them have been proposed to grant authorizations in single sign-on scenarios involving Web applications.

We will talk briefly about OAuth, Delegation Permits, Shibboleth and OpenSSO in the following section.

2.1.1 OAuth

The OAuth initiative has arisen as a method to achieve delegated authorizations. The version 1.0 of the specification has been published in October, 2007. Ever since then, the interest for OAuth has increased to the extent that many companies have adopted or are considering to adopt it (e.g. Google, Yahoo!, MySpace), and a strong developing community has been built around it due to its openness (15).

Using OAuth a user can grant to a Web application tickets to access protected resources hosted in another site, without having to trust any set of credentials. These

www.manaraa.com

protected resources may be data (e.g. pictures, documents), actions (e.g. create a new thread in a forum, send an email) and, in general, any URL with access restrictions. Plenty of information about OAuth can be found in its official Web site (15), and therefore here we will just give a brief summary. The operation of OAuth, depicted in Figure 2.1 as adapted from (15), works on architecture made up by three main actors:

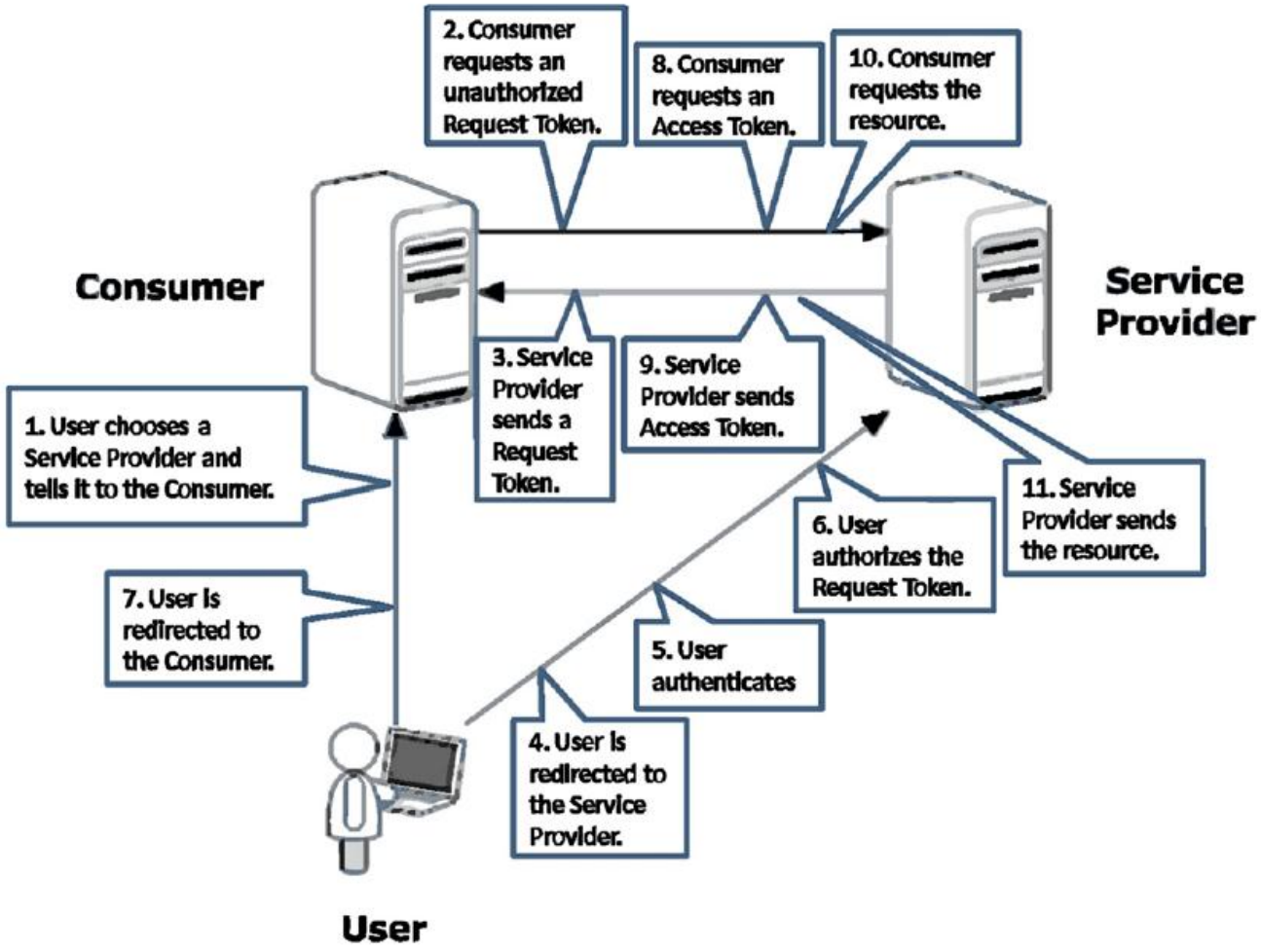

Figure 2.1: OAuth Architecture

www.manaraa.com

- **Service Provider**: Web application that allows the access to protected resources via OAuth.
- **User**: person with an account at the Service Provider.
- **Consumer**: application that uses OAuth to access the Service Provider on behalf of the User.

The internal operation of OAuth is based on the use of single-use tickets, or ‘‘tokens’’ in OAuth’s terminology. For these tokens to be valid for accessing the resource, they must be formerly authorized by the User. OAuth differentiates two kinds of tokens. Access Tokens allow their owner to get the protected resource. Request Tokens, on the other hand, allow their owner to get an Access Token as in (15).

2.1.2 Delegation Permits

Delegation Permits (16) deals with the problem of granting authorizations to mashups in order to access protected resources. Their authors define it as ‘‘a scalable, stateless delegated authorization protocol’’. However, despite the fact that Delegation Permits was originally thought to grant authorizations to mashups, its operation can be generalized to other kinds of systems. The most remarkable differences between Delegation Permits and OAuth involve the actors and messages that take part in the protocol.

Basically, Delegation Permits works over the architecture summarized in Figure 2.2 which adapted from (19), where five main entities are identified:

www.manaraa.com

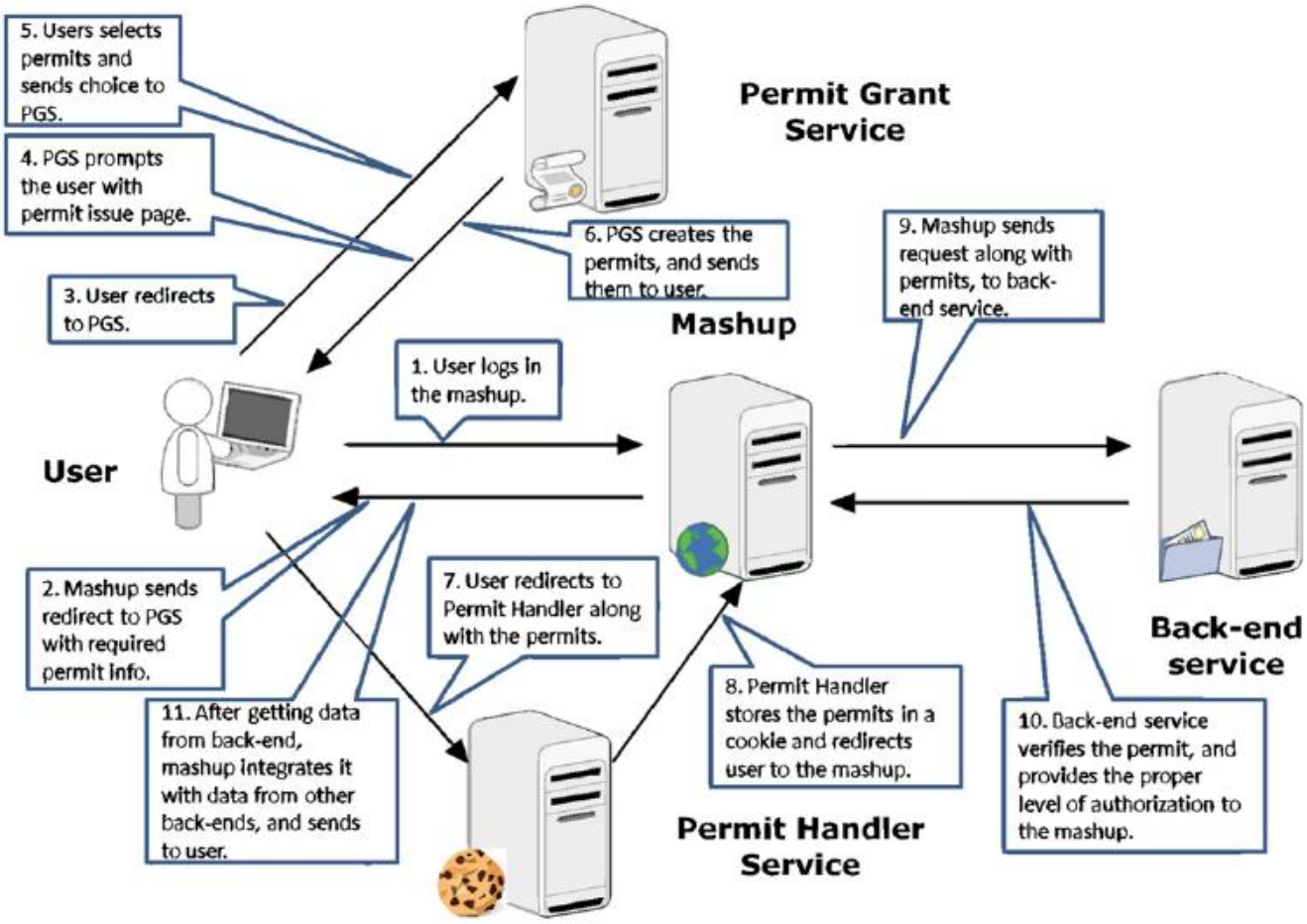

Figure 2.2: Delegation Permits Architecture

- ***Mashup:*** application that wants to access the Back-end Service.
- ***User:*** user of both the Mashup and the Back-end Service.
- ***Back-end Service:*** application that hosts the data the Mashup wants to access.
- ***Permit Grant Service:*** application that deals with the expedition of authorizations to access the Back-end Service.

www.manaraa.com

- ***Permit Handler Service:*** application running at the Mashup, responsible for managing the authorizations once they are received from the Permit Granting Service.

The main architectural difference with OAuth is that in Delegation Permits five entities are involved instead of only three. The entities of OAuth have a rough equivalent in Delegation Permits (User–End User, Consumer–Mashup, and Service Provider–Back-end Service), but there are two more actors: the Permit Handler Service, and the Permit Grant Service. While the Permit Handler Service only provides a further optimization of the performance of the protocol by the use of cookies, the Permit Grant Service supplies the core functionality of the protocol. It provides a centralized environment where the End-User can grant or deny the authorization to the Mashup. However, the Permit Grant Service is a single point of failure, which may not scale well.

2.1.3 Shibboleth

The Shibboleth project (17) is the main effort carried out by the Internet2 consortium (20) to bring a middleware addressing issues in authentication and authorization, in order to make secure inter-institutional services possible and practical.

www.manaraa.com

The primary function of the Shibboleth system is to support transparent access to the resources of multiple sites among which there is a trust relationship, which is known as a federation. Shibboleth pays special attention to enabling independent organizations to federate in order to extend their capabilities. The working of Shibboleth is based on SAML 2.0 (21), an XML-based standard for exchanging security information between partner domains. The information is encoded as attributes, pairs key-value containing information about the User or its role at his/her home institution (e.g. ‘‘profession–teacher’’). SAML has been developed by the OASIS standards working group (21) (22).

Shibboleth works in an architecture made up by four actors as in Figure 2.3 which adapted from (23):

- ***Identity Provider:*** software running at the home institution of the User within the federation. The Identity Provider is responsible for authentication to access the protected resources at the different Service Providers across the federation.
- ***Service Provider:*** institution of the federation that stores the protected resource the User wants access to.
- ***User:*** person with a user account at the Identity Provider and that wants to access a protected resource at the Service Provider.
- ***WAYF (Where Are You From):*** centralized service of the federation that allows the User to choose his/her home institution.

When the User wants to access the protected resource, he/she navigates to its URL using his/her Web browser. The Service Provider redirects the browser to the WAYF, where the

www.manaraa.com

User can select his/her home organization. The browser is sent to the home organization's Web site running a Shibboleth Identity Provider. The User can optionally abandon the WAYF, if the system includes a single institution.

The Identity Provider sends the browser back to the Service Provider. The User has been recognized by the Identity Provider as one of its users, and the Service Provider has been notified of this.

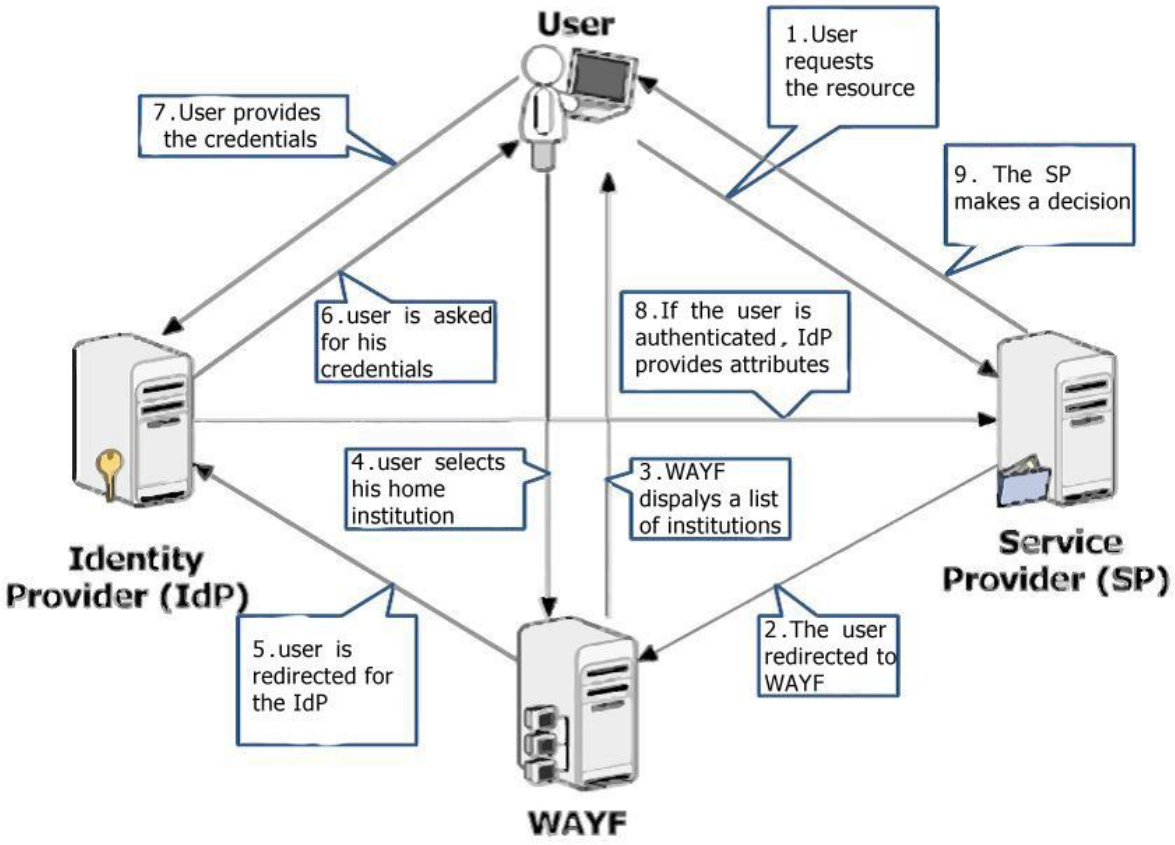

Figure 2.3: Shibboleth Architecture

www.manaraa.com

The process is complete when the Service Provider requests additional information (attributes) about the User to the Identity Provider. These attributes do not necessary involve User's identity, although it could be requested as well. The Service Provider uses these attributes to decide the access policy for the User (e.g. the functionalities granted, the length of the session).

The previous process can be decoupled into two independent stages, depending on the results obtained at each one. On the one hand, the first stage ends when the User has logged in at his/her home institution by means of the WAYF. At this step, the Shibboleth system has achieved a delegated authentication of the User. On the other hand, the second stage involves the request and sending of attributes of the User to determine what he/she is allowed to do. Hence, the second stage provides delegated authorizations.

An important missing part in Shibboleth is the complementary process of single sign-on, generally known as single log-out (24). At the time of this writing not only single log-out is not addressed by Shibboleth, but the requirements are not even clear (25). There are two possible behaviours expected when the User clicks the Logout button: logging out from the concrete Service Provider, or logging out from all the Service Providers of the federation. The only way to ensure the end user's session is to close the browser.

www.manaraa.com

2.1.4 OpenSSO

OpenSSO (18) is the Sun Microsystem's (26) approach to the problems of managing federations with external partners and securely federating legacy applications that do not support federation.

To do so, OpenSSO provides a middleware consisting of core identity services for the implementation of single sign-on in heterogeneous environments. As Shibboleth, OpenSSO is based on the request and sending of attributes concerning the user between an Identity Provider and a Service Provider by the use of SAML 2.0. In general terms OpenSSO deals with the adaptation of identity information transfers from the legacy application into SAML messages that can be understood and processed by all the applications of a federation.

A summarized description of the operation of OpenSSO is shown in Figure 2.4 which adapted from (23), where three actors can be seen:

- ***Identity Provider*:** home institution of the User within the federation. The Identity Provider runs an OpenSSO instance that encapsulates information about the User (e.g. authentication, profile attributes) and sends them to the Service Provider.
- ***Service Provider*:** institution of the same federation of the Identity Provider running another OpenSSO instance. The Service Provider receives the SAML identity information about the User, checks it, and uses it to grant the User access to protected resources according to an access policy.

www.manaraa.com

- ***User*:** person with a user account at the Identity Provider, who wants to access a protected resource at the Service Provider.

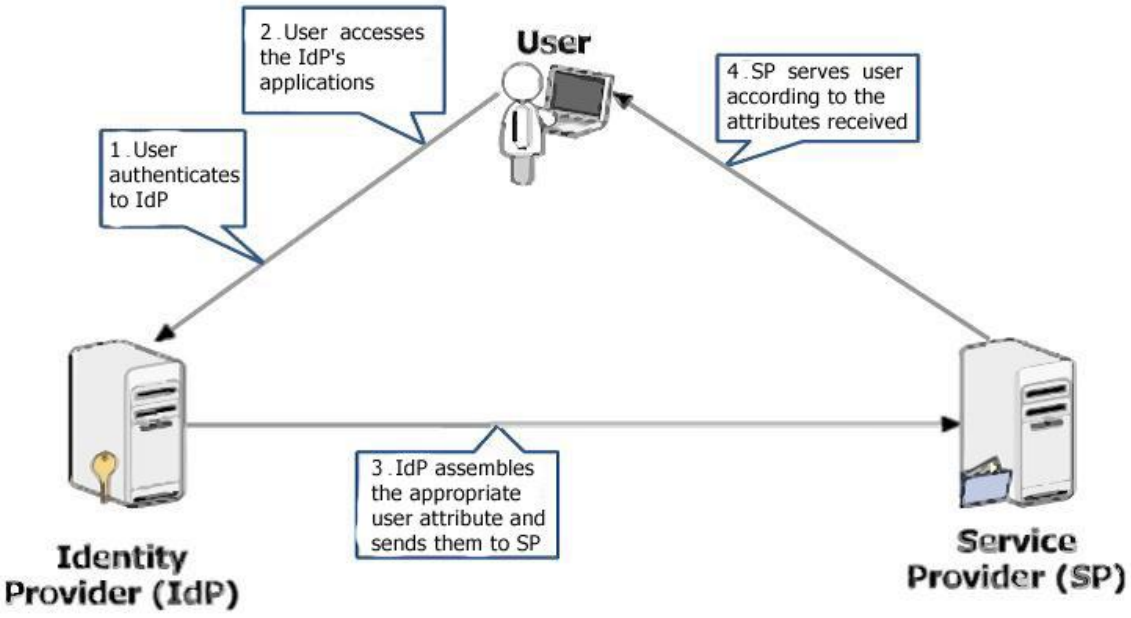

Figure 2.4: OpenSSO Architecture

The process is very similar to Shibboleth with the difference that there is no equivalent to the WAYF. A User authenticates to his/her Identity Provider application. Then, he/she clicks a link that points to a service provided by another application in a different domain (Service Provider's). The Identity Provider's OpenSSO instance assembles the appropriate SAML user attributes (authentication and user data), and encodes, signs and sends them to the Service Provider using SSL. The Service Provider's local OpenSSO instance decodes the attributes and sends them to the Service Provider's application, which compares the

www.manaraa.com

attributes against some access policy. Finally, the application serves the user according to such policy.

2.2 SAML

The Security Assertion Markup Language is a XML encoded framework for exchanging authentication, subject attribute and authorization information. SAML has been designed by the Organization for the Advancement of Structured Information Standards (OASIS) to provide a universal mechanism for conveying security-related information between the various parts of an access control system. It is a very extensible, open standard, which makes it attractive as a basis for further development.

SAML consists of assertions, request-response protocol, binding and summary. Among them, assertion is the core of SAML.

SAML defines three types of assertions:

- Authentication: indicating that a subject was authenticated previously by some means (such as a password, hardware token, or X.509 public key).
- Authorization decision: indicating that a subject should be granted or denied resource access.
- Attribution: indicating that the subject is associated with attributes.

Using a subset of XML, SAML defines the request response protocol by which systems accept or reject subjects based on assertions. A SAML request can either ask for a specific

www.manaraa.com

known assertion or make authentication, attribute, and authorization decision queries, with the SAML response providing back the requested assertions. The language defines a binding to describe of how SAML request/responses are carried within Simple Object Access Protocol (SOAP) exchange messages over Hyper Text Transfer Protocol (HTTP).

Assertion is core of SAML; there are three types of SAML assertion:

a) Authentication assertion

Authentication assertion deals with the particular information relate to the behavior of SAML framework Authentication to a subject. A typical statement can be described as: authority A asserts that it has used a method M to authenticate subject S at a particular time T.

b) Attribute

Attribute assertion deals with specific attributes of a subject associate with assertion. A typical statement can be described as: authority A asserts that subject S has attributes X, Y and Z, and their values are “x”, “y”, “z”.

c) Authorization decision

Authorization decision assertion manages the decision for a given subject’s authority to access resources（e.g. Approval or rejection information for a request). A typical statement can be described as: authority A approves or rejects the accessing request of subject S to resource R in manner of M according to policy P.

www.manaraa.com

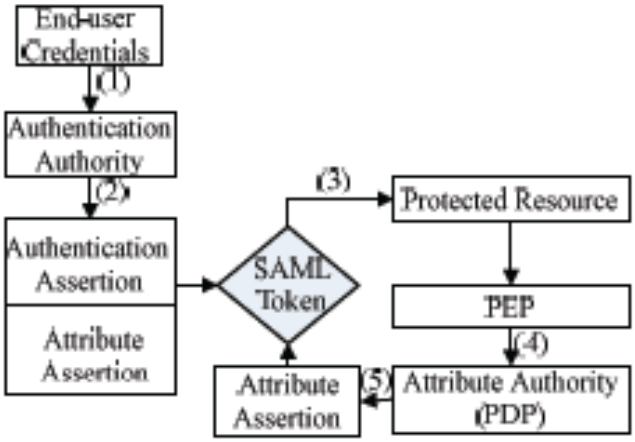

Figure 2.5: Basic principle of SAML

Figure 2.5 - which adapted from (21) - illustrates the basic working process of SAML, Specifically; it can be described as follows:

(1) User submits credentials to Authentication Authority;

(2) Authentication Authority asserts user's credentials and generates an Authentication Assertion together with one or more Attribute Assertions. User is now authenticated and identified by SAML assertions assembled in a token;

(3) User attempts to access a protected resource using the SAML token;

(4) Policy Enforcement Point (PEP) intercepts end-user request to protected resource and submits the end-user's SAML token (Authentication Assertion) to the Attribute Authority;

(5) Attribute Authority or Policy Decision Point (PDP) makes a decision based on its policies. If it authorizes access to the resource, it generates an Attribute Assertion attached to the user's SAML token. The end-user's SAML token can be presented to trusted business partners to implement trust transfer.

www.manaraa.com

We can identify users by Identity authentication server which serves as SAML authority for all the participants. When receiving a soap message from the client, Identity authentication server will treat the message with reverse processes to extract the SAML request. After that, it will generate a SAML token that contains the user's identity and attribute information which is obtained by querying the Subject/Role library.

www.manaraa.com

2.3 Educational Model Requirements

After the description of the architecture under study and the use cases we are now ready to list more formally the main requirements of a solution. A subset of the following requirements has been identified as very urgent by some e-learning commissions (14).

The Requirements (14) are in order interoperability, access transparency, privacy, choosability, granularity, simplicity, dynamic reconfiguration, expiry, awareness, pseudonimity, confidentiality, integrity, authenticity, and single-use authorizations, see Table 1.

Table 1 was adapted from (23) ; The Shibboleth federation system has achieved some security requirements while keeping some of them, so we will work to resolve the security vulnerabilities in the system to meet the other requirements, as Access Transparency, Dynamic Reconfiguration, and Awareness. The definitions of those requirements which mentioned in (14) are:

Fulfillment Requirements By Shibboleth														
	R1	**R2**	R3	R4	R5	R6	**R7**	R8	**R9**	R10	R11	R12	R13	R14
Shibboleth	√	X	x	√	√	x	x	√	X	√	√	√	√	√

Table 1: Fulfillment Requirements by Shibboleth

Access Transparency: It should be possible for the users of the learning system to access the resources without re-enter the password or credentials, especially as he is known to the system in advance.

www.manaraa.com

Dynamic Reconfiguration: It should be possible for a learning system to modify the characteristics of users to access resources. The Learning System may need to modify the status of the users.

Awareness: The learning system should be able to track the activities of each user at an Identity server. The learning system can grant access to additional resources depending on the information it receives from the tools.

2.3.1 Security Vulnerabilities of Educational Resources in Shibboleth

One of the first issues with Global Logout is communicating to the user what will occur if they click "logout". Users have already been taught that doing this will cause them to be logged out of the application that contained the logout button/link they used. They assume that all other applications will remain active. Therefore there must be some means to distinguish between application-level logout and Global Logout.

An example case goes as follows. A student accesses the university's student information system, which participates in the SSO service. The student goes to register for classes and is reviewing possible courses. During this review process the student access some material from the library. Because they have been taught to always logout of applications is done correctly, they click the local logout link in the library application. The session will not end and the student stills authenticated, so the resources are at risk.

www.manaraa.com

This issue is usually addressed by providing a consistent UI component across all participating applications. MS Passport, Virginia Tech's Auth portal, and other systems, do this by placing a branded, distinct, image/button in place of the application-level logout link/button. When a user visits a site that participates in the SSO system the application displays a form of the UI component that indicates that the user may log in. When the user is logged in the UI component is changed to reflect this state.

This approach suffers from two problems. First it requires educating users about this UI component and what it is meant to convey. Second, it requires all SPs display and implement the same UI component. While the first issue can be solved through an education effort but the second problem is practically untenable.

Accessibility and Visibility

The Global Logout of the educational federation system must be accessible for users. The feedback messages must be supported and understandable. One button/link has to do all of the logout process at once, so the solution for the logout problem in the shibboleth federation system must be developed accessible by providing the suitable messages for the users.

www.manaraa.com

Chapter 3 – Related Work

In this chapter, we review some of the significant and recent research papers in the field of federated secure models and its problems. We present these activities and discuss their advantages and the disadvantages. More precisely, we will discuss the reasons of why these solutions do not give the required secure model.

3.1 Secure Model Behavior

In this section, we will mention the federation model and its requirements. We will concern with four requirements: Access Transparency, Dynamic Reconfiguration, Awareness, and Accessibility and Visibility. Those requirements are needed to any Educational Services across multiple domains Model. It is important to us in our research is study the how to construct models in federation environments.

The brief description of meaning of those requirements will be discussed in educational secure model view as follows:

A. Access Transparency

It should be possible for the users to act with multiple applications as a one either in logon or logout phase. That not supported at the logout phase, Shibboleth depends on the Local logout without any guarantee to log out from the IdP as in the specification of Shibboleth (17).

www.manaraa.com

B. Dynamic Reconfiguration

Dynamic Reconfiguration is not supported also, because the Shibboleth architecture has not been designed for scenarios where the authorizations are tied to planning in the manipulating with the Service Provider that model carry out the configuration at the beginning of the authorization process, without any modification to the properties of an ongoing authorization. The Lack of a single log-out mechanism in Shibboleth (24) implies that is not easy to know if the user is still logged in the federation.

C. Awareness

Simply, it is information about the users to track their activities. Then Identity Provider in Shibboleth can track the real opened sessions. In Shibboleth, the IdP may deceive by the existence of open sessions while users are actually outside the system.

Federations are dominated on the models in several fields; education, payments, and learning services. Some of research use a SSO models as described in chapter 2 with a modification in that models to meet the specification of the security.

Gongalez and his group in (23) proposed a modification to one of the federation architectures, OAuth (15) SSO model. The modified model named *Reverse OAuth* which has a fulfillment of the educational security requirments in Table 2 as adapted from (23).

Users have to introduce several kinds of credentials, preventing them from focusing their efforts on their studies and increasing the so-called ‘‘password stress’’. Several initiatives such as OAuth or Delegation Permits have dealt with the problem of delegated authorizations, but their requirements are different from those that arise from an e-learning

www.manaraa.com

environment (23). So Reverse OAuth proposed as 'a protocol to enable the granting of authorizations to access protected resources in educational environments'.

Table 2 – Fulfillment of the requirements by Reverse OAuth.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
OAuth	✓	✓	✓	×	×	✓	×	×	×	×	×	✓	✓	✓
DP	✓	✓	✓	×	✓	×	×	✓	×	×	×	×	×	✓
Shibboleth	✓	×	×	✓	✓	×	×	✓	×	✓	✓	✓	✓	✓
OpenSSO	✓	✓	×	✓	✓	×	×	✓	×	✓	✓	✓	✓	✓
R. OAuth	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	×	✓	✓	✓

Table 2: Fulfillment of the requirements by Reverse OAuth

Gongalez and his group in (23) don't solve the problem of Shibboleth logout and focus on the Reverse OAuth as a solution to his learning model system.

Lutz, and David J (6) also introduce a new federation payment system. That guarantees reliable on-the fly payments without having to contact a payment provider each time a payment is required. They use the abstract architecture of the SSO model.

3.2 Global Logout Problem

The logout operation in shibboleth-enabled systems occurs totally when the user close the browser, all the opened sessions will be closed. Suppose we have three open shibboleth-enabled applications and the user want to logout from shibboleth application 1 and still be in shibboleth application 2 and 3.

www.manaraa.com

Firstly, Identity Provider is responsible for creating, maintaining, and managing user accounts, while Service provider is responsible for controlling the access to the resources by analysing the SAML assertions.

Before we talk about the types of the logout we have to be familiar with the SSO as in Figure 3.1, which show us a sequence diagram for the SSO SAML Operation (22) used by Shibboleth as steps:

1. The client attempts to access the educational resources at the service provider as a web services.
2. Identity provider perform vary methods to identify the user.
3. The service provider SP determines which identity provider IdP to use.
4. Perform permanent redirection to the user to the identity provider including the AuthRequest issued by SP.
5. Identity provider response the SAML Response including the Auth assertions and attribute assertion
6. Based to the Auth assertion the SP determines to return the requested resources.
7. Resources return to the user as a HTTP response, or HTTP error.

www.manaraa.com

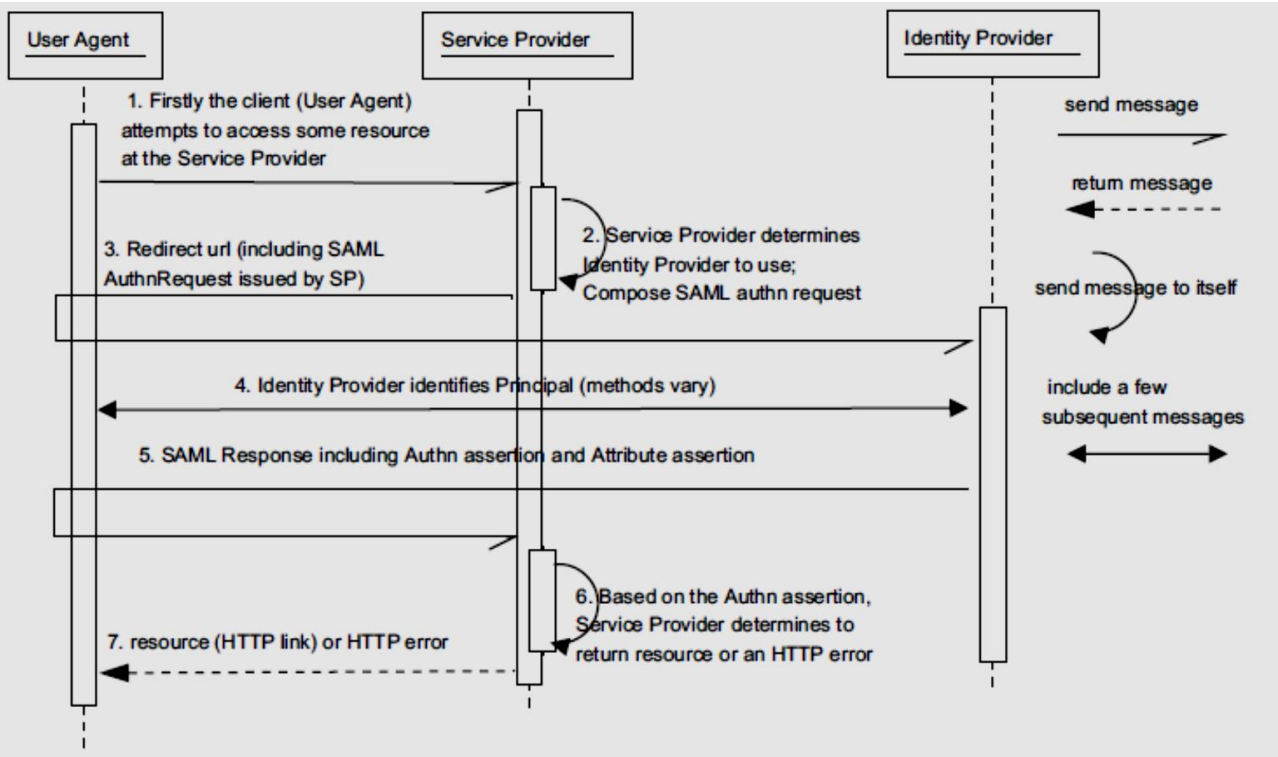

Figure 3.1: SAML authentication in shibboleth login

The logout in shibboleth specification (17) is defined as a local logout. The solution introduced by shibboleth community (27) was the second logout case. Researcher works on that problem and some of them as in (25) introduces the third case with a partial solution to the global logout.

1- Local logout

It called SP logout, which is done by logout the user from the SP only without any guarantee to remove the user session from the IdP (see Figure 3.2). That type is a default logout in the shibboleth implementation.

www.manaraa.com

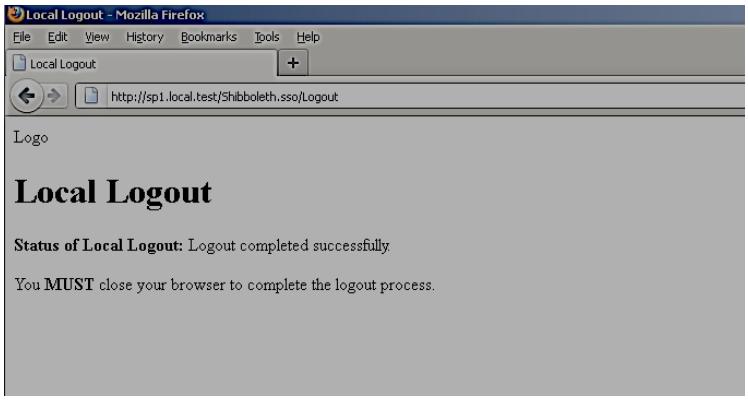

Figure 3.2 : Local logout of the service provider

If the SP is configured on sp2.shibboleth.edu, then the URL *https://sp2.shibboleth.edu/Shibboleth.sso/Logout* can do the SP logout. In SP, we can customize a page to instruct the user to close the application and other custom content. Dangerous situation is that the user session is still in the IdP, and this is a loophole in the security system.

2- Local logout with browser close

The browser is automatically closed by the code implemented in the SP logout page which invokes the browser to close automatically. In this case the user will re-authenticate at the next request if no authenticated session still opened in the same computer. The drawback is the complexity of writing code at every SP, also the Varity of browsers face the capability of the response to the closing code at SP logout page.

3- Local Logout with forcing re-authentication

This is the first type with forcing to authenticate every time the user do the local logout, this can be done by modification the identity provider of Shibboleth.

www.manaraa.com

This Type of logout is not ideal in a "portal" environment, since it requires re-authentication every time the application is accessed from the portal even after initial authentication at the portal.We can summarize the current state of the logout research as comparison between them in the Table 3.

no	Logout	Description	Advantage	Disadvantage
1.	Local logout	• Basic implementation in shibboleth community. • Logout the user from the SP whiteout the IdP.	Simplicity	• Loophole in the security system. • Sessions still alive while users logged out.
2.	Local logout with browser close	• First solution by the shibboleth community. • Depend on the user.	More secure than first case	• The complexity of writing code at every SP. • The Varity of browsers face the capability of the response to the closing code at SP logout page.
3.	Local logout with re-authentication	• Forcing to authenticate every time the user. perform a request	Intense security	• This Type of logout is not ideal in a "portal" environment

Table 3: Comparison between previous logout problems

www.manaraa.com

In our case, the resources are an educational resources and the identity provider implementation is a Shibboleth IdP. The modification part of our research will be in the identity provider of the Shibboleth. We have some security requirements must be applied to the federation educational model as described previously in the chapter 2. So, our modification is appearing in chapter 4 and chapter 5.

www.manaraa.com

Chapter 4 – Secure Educational Resources

This chapter reviews the suggested solution for educational resources in Federation System to be more secure. The chapter begins with an overview of the proposed system, the sequence diagrams of the workflow and the security requirements achieved.

4.1 Overview

The proposed system builds upon the fact that federation system uses Single Sign-on approach which enables users to login in many systems by single profile information, beside the single login we have to achieve the integrity of the system to enable the single logout or global logout. In our approach we suggest the Shibboleth Federation system in the educational system, the logout process in shibboleth is local which means many points of risk in the system, and we will determine the security problems in the model and suggest the solution to them. Figure 4.1 summarizes this process. As the figure illustrates, users can choose the Service Provider (SP) to get the service. Once they finish, the Global Logout will be accessed to them and they can logout from all Service Provider he logged-in before.

www.manaraa.com

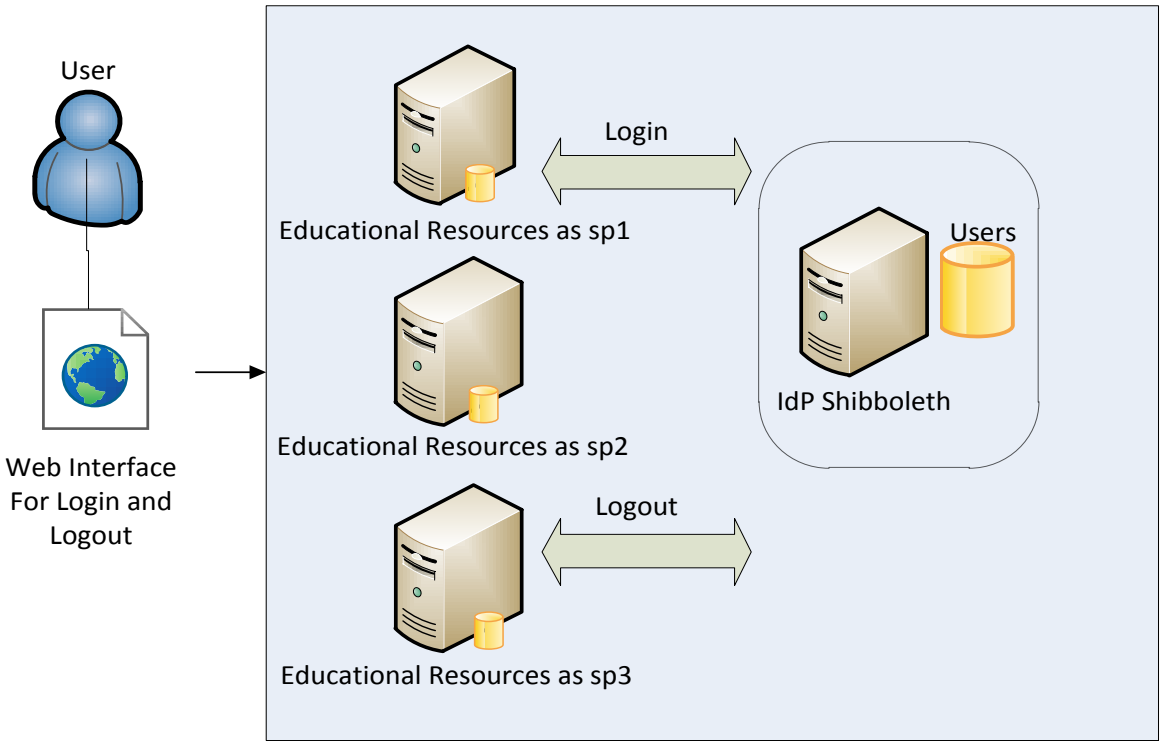

Figure 4.1: Secure model architecture

4.2 Secure E-Resources

The educational resources have many services being provided to students in universities, organization, and camps.

The educational resources have many users from multiple fields or domains to being access, so the demand to the federation system will reduce in time and efforts in the authentication and authorization process in multiple systems, as we know every system has to copy and handle the login information by users in old approach as in Figure 4.2.

www.manaraa.com

In Federation System the handling of login will be a unique for all systems in. as we using Shibboleth in our approach, we will improve the logout process in the systems to get more secure educational resources.

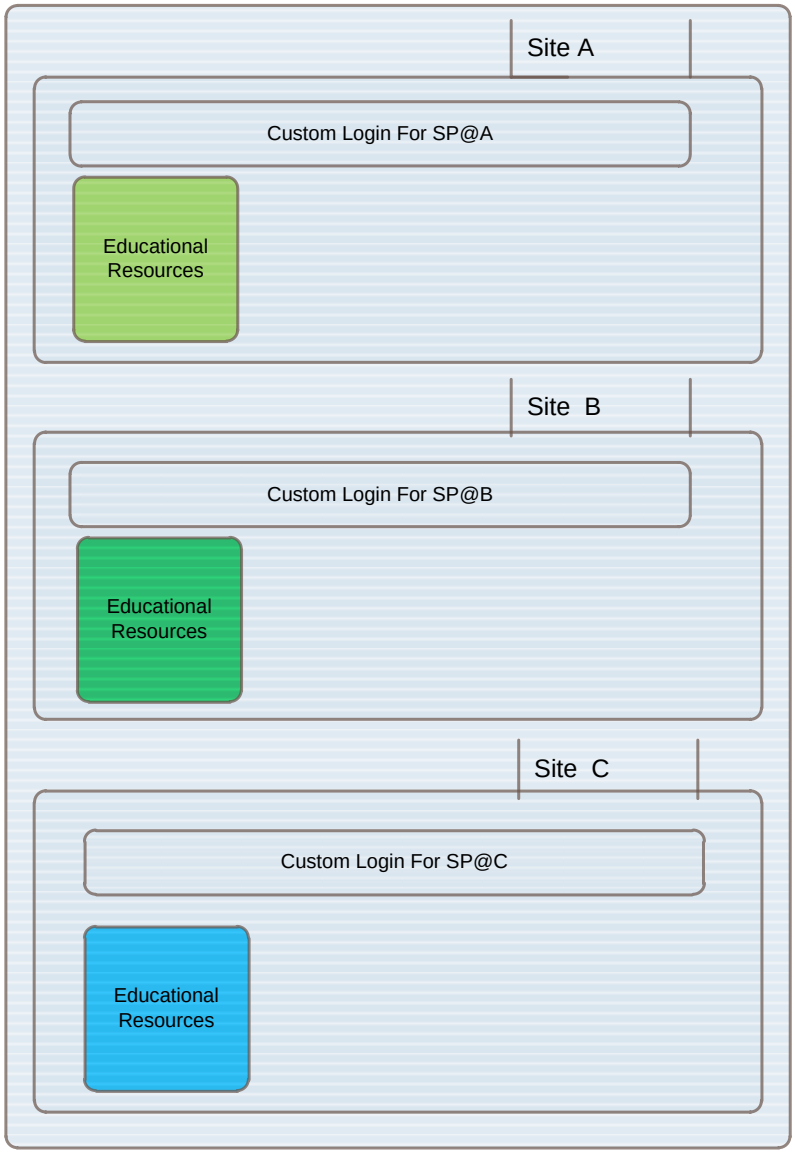

Figure 4.2: Educational resources in multiple sites

www.manaraa.com

4.2.1 Federation Model

In Federation System contains three layers as shown in Figure 4.3, the first layer will be web-based interface to user, who enter the required single login information to get authenticated. The second layer will be the federation system used as discussed in chapter 2 many of them, as OAuth, Delgation Permits, Shibboleth, and OpenSSO. Each of them has different environment and implementation. The third layer will be the educational resources as service provider (SPs) for each resource.

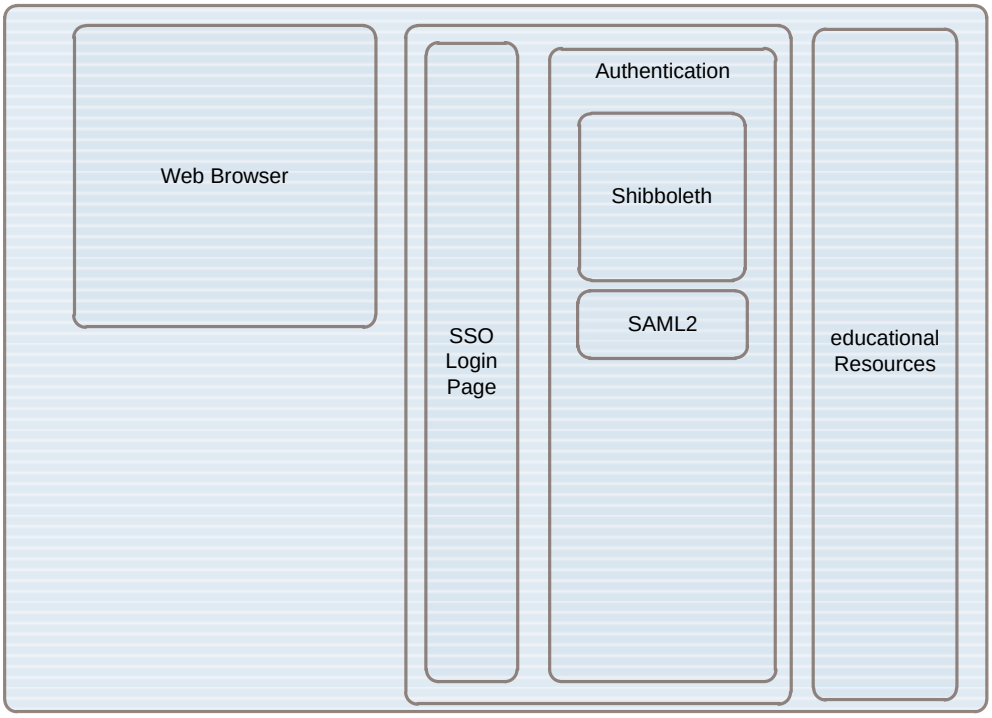

Figure 4.3: Federated educational resources model layers

www.manaraa.com

The Security integrity of our approach is in logout process in the Shibboleth Identity Provider, the logout we suggest is the fully logout of the system as layered in Figure 4.4.

Local logout can be effective if we ensure that the user will close all sessions opened in his browser , if not the risk will be increase to be in attack, So we put the our educational resources in secure model and in secure system.

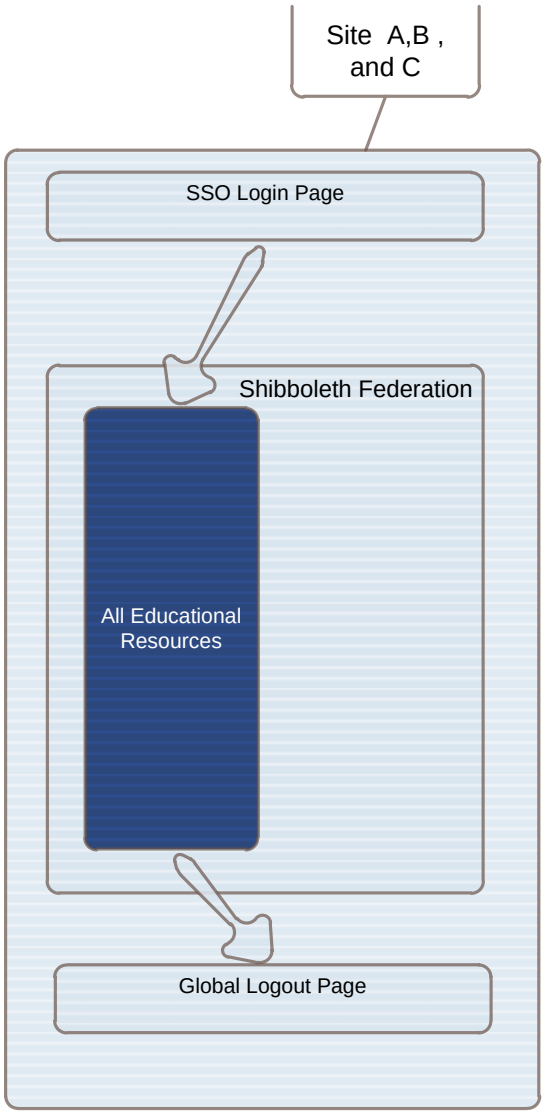

Figure 4.4 : Federation login and logout in educational system

www.manaraa.com

4.2.2 Sessions

In a single sign-on scenario, there are at least four sessions for a user, each created on top of the previous one as in Figure 4.5:

1. IdP authentication session
2. IdP session (active authentications, associated SPs etc)
3. SP session
4. Application session

These sessions are usually implemented with different session stores, therefore they expire or time out independently. For single logout, it becomes a security problem if a session remains usable after the parent session is expired, So we avoid stale sessions, the IdP session should have a limited inactivity timeout ("soft timeout", the maximum time between user activity) but an unlimited session lifetime ("hard timeout", independent from user activity).

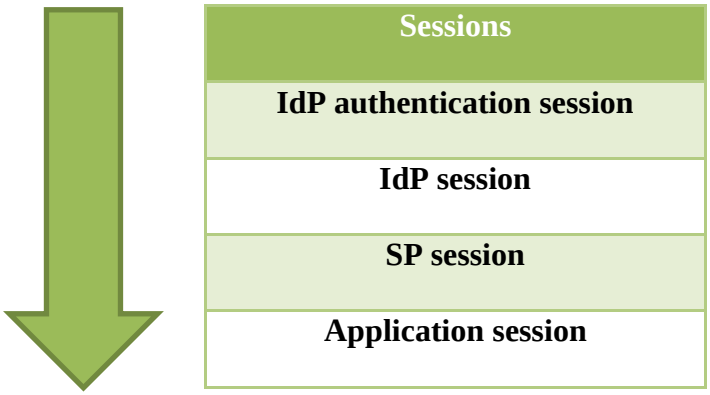

Figure 4.5 : Sessions in shibboleth

www.manaraa.com

The SP sessions must have a session lifetime, which is not longer than the inactivity timeout of the IdP session. The IdP must set the **SessionNotOnOrAfter** attribute in the authenticating assertion according to its session inactivity timeout value. The SP must honor this attribute and set its session lifetime based on this information.

If the SP session lifetime or inactivity timeout is shorter than the IdP session inactivity timeout, then the user would possibly get an error message during single logout because of the expired SP session. Anyway, we inform errors caused by SP inactivity timeout as "Session not found" errors.

4.2.3 Logout operation

We now describe the operations of the global logout, the factor will be.

- User – refers to the federated user who requests and uses the educational resources
- 1st SP – refers to first resources in the shibboleth as a service provider.
- 2nd SP – refers to second resources in the shibboleth as a service provider.
- Shibboleth IdP – refers to Identity server in shibboleth which responsible for mange the sessions and the roles between the users.

We summarize the steps of log out process as the shown in the Figure 4.7 in steps:

1- User get the Educational resources from 1st-SP after a SAML assertion message arrive from the IdP contained the user ID in the login profile as a (*ali*) user in following Figure 4.6.

www.manaraa.com

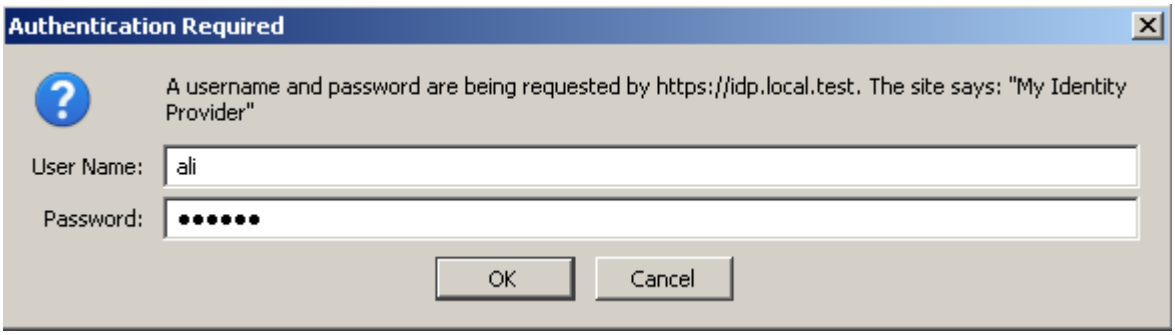

Figure 4.6: Single sign-on login in shibboleth

2- User get the different Educational resources from the 2nd-SP as the 1st-SP.

3- User will leave from all resources in which he was before by a one click, so he will send a request to the SP.

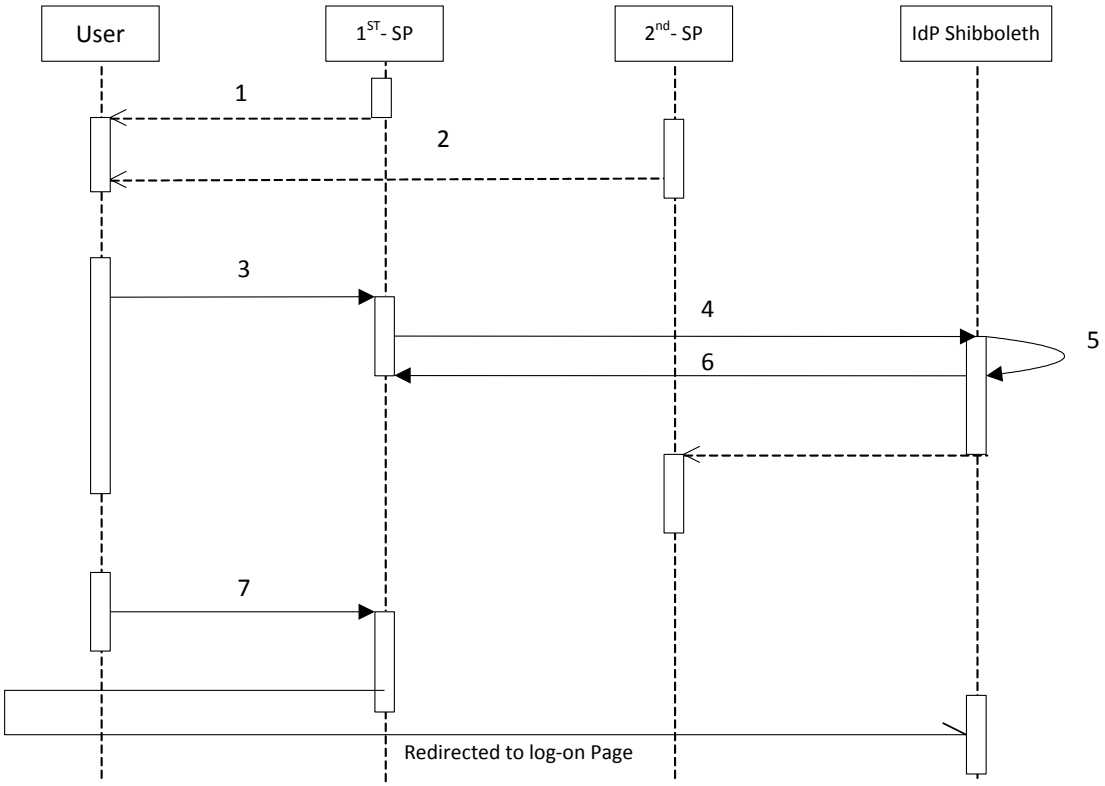

Figure 4.7 : Proposed logout sequence diagram

www.manaraa.com

4- The process of log out does not complete without the IdP, so the 1st-SP will send a SAML Message as a (HTTP-Redirect) to IdP included the user shibboleth ID as a parameter to be identified in a logout profile.
5- IdP will log out the session opened by him.
6- IdP will send a log-out message to 2nd-SP, identified by User ID identified in the shibboleth configuration files or a Database user.
7- Any Request from the user will be directed to the re-log-on page to authenticate the user.

4.2.3.1 Global Logout Impact

The federal system has the specifications in the process of entry and exit processes, particularly in the education systems and educational services, so the problem of the global logout from that systems was the biggest obstacle in the application of federal models on educational services.

A sense of security and protection for user's information and resources through the various web services is the feeling that cannot be achieved through a system of shibboleth, so the proposed solution to the logout problem somewhat met the requirements of the user.

User is looking to logout of the services, at the same time keep a copy of the open information. By caching the content and re-authentication condition achieve that point.

Later in chapter 5 we summarize the advantages of the enhancing the model in educational environment in Table 5.

www.manaraa.com

4.2.3.2 Global Logout Servlet Parameter

We will parameterize the response with no cache and the new request must be revalidate, so the response header of the SAML request of the Logout will be as the following in Figure 4.8:

```
resp.setHeader ("Cache-Control", "no-cache, must-revalidate");

resp.setHeader ("Pragma", "no-cache");
```

Figure 4.8: Response header due to logout request

The cookies in the browser as shown in Figure 4.9 will be deleted as the following:

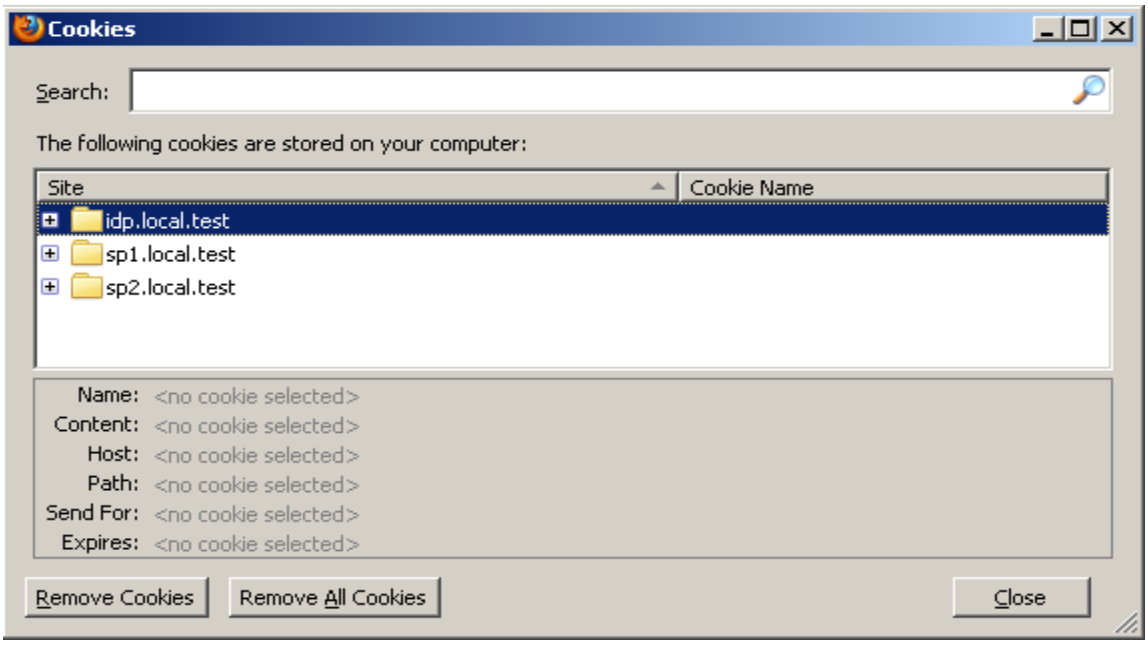

Figure 4.9: Cookies in Firefox

The forward of the request and response is not direct , but in HTTP binding attribute, so the servlet class will clear the cache in the browser for the services required and redirected the request to the logout profile which unbinding the sessions in the IdP as the destroySession function works as in Figure 4.10:

www.manaraa.com

```
Private void destroySession (SingleLogoutContext sloContext) {
    getSessionManager ().destroySession(sloContext.getIdpSessionID ());
  }
```

Figure 4.10 : Destroy sessions in IdP

The fully logout information will be as the following in Figure 4.11.

Figure 4.11: logout status information of the SPs

Then the cookies will be deleted and the sessions in IdP will be destroyed also, Figure 4.12 show the cookies in the browser after the global logout will be achieved.

www.manaraa.com

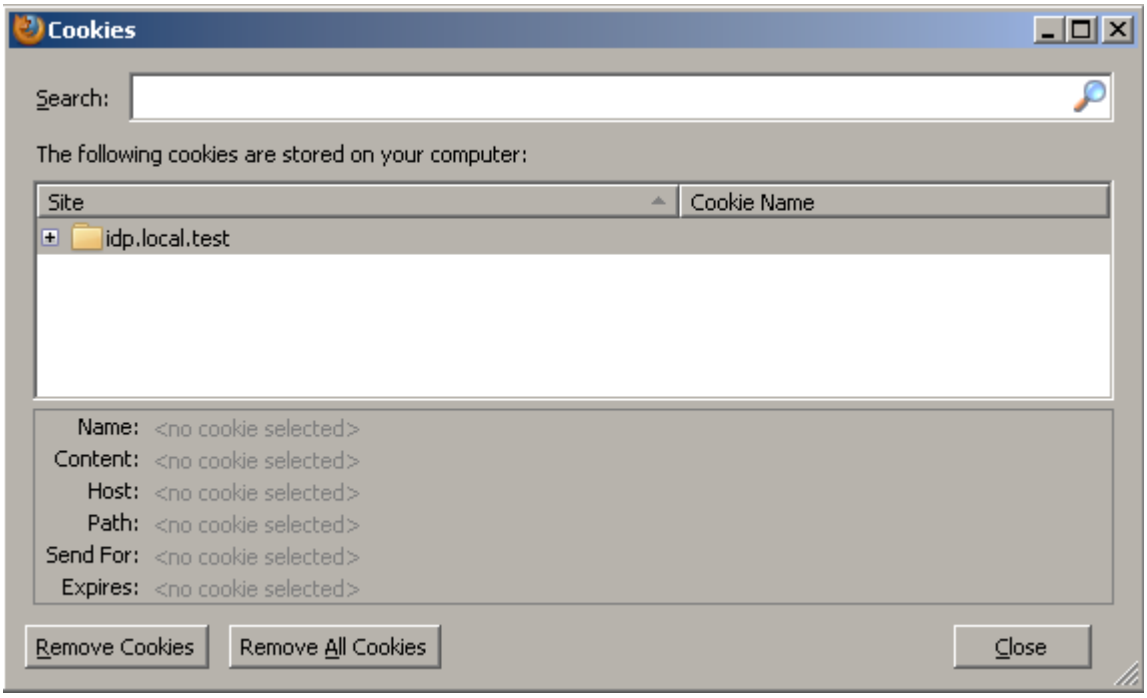

Figure 4.12: Cookies deleted in Firefox browser

The Sessions in the IdP will be invalid Sessions and will request to be revalidate upon the next request as in the Figure 4.13.

Figure 4.13: Sessions in IdP was deleted for both sp1 and sp2

www.manaraa.com

4.2.4 Caching

Order to maintain the requirement of accessibility and visibility of the system; caching is required to maintain the status of the user either in logout status or login status.

User status within the Shibboleth federation system is just logged-in or logged-out, but to apply the visibility requirement a third status must be implemented as 'off-line status'. Off-line status is the status when the user logged out of the Shibboleth federation system, while retaining the open-content sources in advance, see Figure 4.14.

User Logout Status
LOGGED_IN
LOGOUT_SUCCEEDED
LOGOUT_FAILED
LOGOUT_TIMED_OUT
OFF_LINE

Figure 4.14: User Status in Our Enhanced Model

Other Status in the Figure 4.14 will change according to our logout state diagram in Figure 4.15. The status diagram can be explained by four cases; the first case, when the logout successfully occurs by a click from a user, the status will change from **LOGGED_IN** to **LOGOUT_SUCCEEDED**. The second case, when the logout request does not successes for any reason either in the network or application, the status will change from **LOGGED_IN** to

www.manaraa.com

LOGOUT_FAILED. The third and fourth case, when the user logged out by time out T1 setting in the IdP, then the status changes to either **LOGOUT_SUCCEEDED** or **LOGOUT_FAILED** as the same reasons in the first and second case. The two state **LOGOUT_SUCCEEDED** and **LOGOUT_FAILED** go to **OFF_LINE** status.

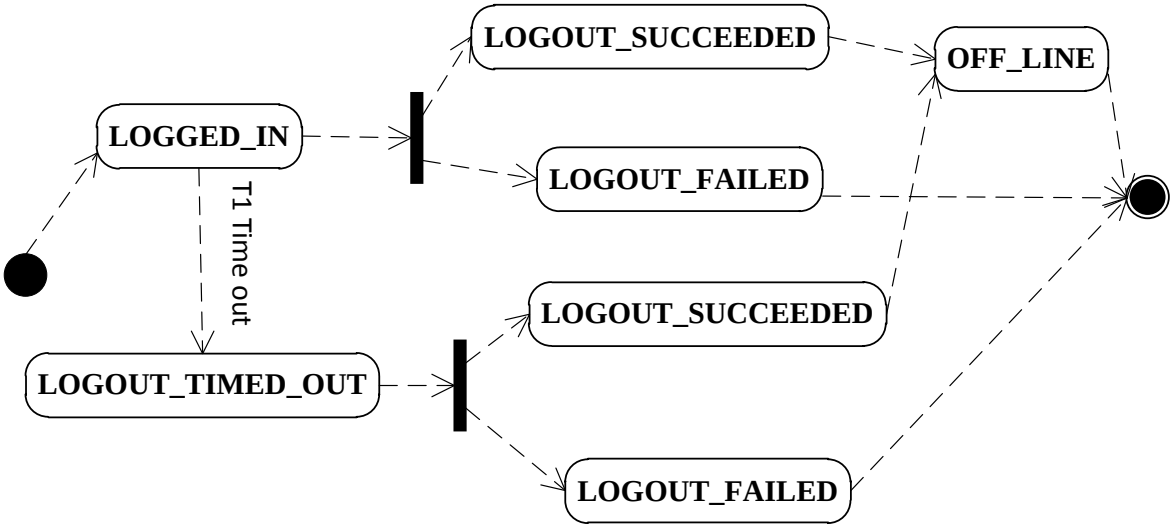

Figure 4.15: state diagram of global logout

We can implement caching in two levels, the first one is in the service provider (sp) level and the second is in the identity provider (IdP) level. The caching in the sp is enabled by the user, so the control will be in the user's browser, while the most effective and guaranteed caching will be in the IdP because the status of the user will be saved at the whole federation model.

The two sections below will define the conceptual view of the two levels of caching; service provider and identity provider caching.

www.manaraa.com

Chapter 5 – Implementation

In this chapter, we review the implementation of the educational resources identity federation. The chapter explains the implementation environment and then the final results are summarized.

5.1 Implementation Environment

The educational resources identity federation was implemented using Shibboleth which consists of Java coded Identity Provider (IdP) was Compiled using *NetBeans 7.1 IDE* (28) installed on *Ubuntu 10.04* (29) Linux distribution and C++ Coded Service Provider (SP) (30) was managed by IIS and installed on *Microsoft Server* 2003 Enterprise Edition (31). The web interface of the educational services was created using Active Server Pages (ASP.NET) (32) technology.

5.2 Educational Web Interfaces

The Interface of Service provider (sp1) will be html interfaces with some links for services for free or authenticated service as in Figure 5.1.

The Global Logout will be direct link to user on sp1 which enable him to be fully logout of the all services opened before as in our case when be two service provider.

www.manaraa.com

Figure 5.1: Educational service web interface sp1

The URL for the first Service Provider is (http://sp1.local.test/), the free service is denoted by the URL (http://sp1.local.test/Services/) and the secure library service can by access by the URL (http://sp1.local.test/secure/Library), also Same URL for Service Provider (sp2), see Figure 5.2.

Figure 5.2: Educational service web interface sp2

www.manaraa.com

5.3 Federation Configuration

The Shibboleth federation system has two sides; one for Identity provider and other for service provider, so the configuration isn't unique for both but match each other. The request will come from the service provider and send to the identity provider as a SAML message for login authentication or global logout.

5.3.1 Identity Provider Configuration

At this section we explain two parts of configuration and its implementation, the first part will talk about the logout configuration and the second one will be for caching configuration.

5.3.1.1 Logout Configuration at IdP

After installing the Shibboleth Identity Provider (27), we can show the folders (bin, log, conf, credentials, metadata, lib, war) as In Figure 5.3.

War folder contains the war version of the compiled java identity provider code, conf folder contains the xml configurations files, credentials contains the authenticated users, metadata contains the metadata of all service providers, log contains the log occurs by the processes, and lib contains the compiled libraries.

www.manaraa.com

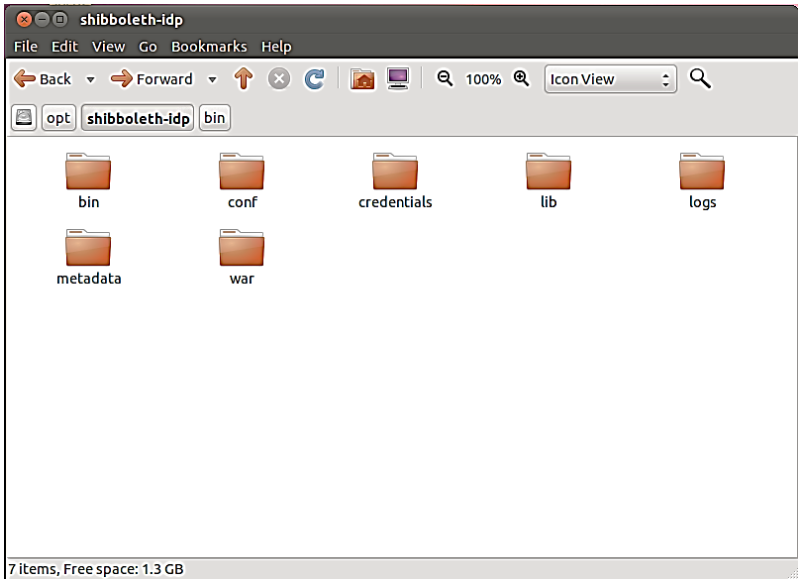

Figure 5.3: Identity Provider installed folder

Login will be either with a default login page connected to database for authenticated user or with authenticated according to file contains the users (users.db) located at the Service Provider (SP) under credentials folder.

Metadata must be identified for all service providers as in Figure 5.4; metadata is defined in each service provider as a URL (http://sp1.local.test/Shibboleth.sso/Metadata), for service provider sp1 and a URL (http://sp2.local.test/Shibboleth.sso/Metadata) for service provider sp2. Metadata located in Identity provider configuration files *conf* in *relying-party.xml* file by a metadata Provider identification tag.

www.manaraa.com

```
        <metadata:MetadataProvider id="SP1_MD" xsi:type="metadata:FileBackedHTTPMetadataProvider"
                                   metadataURL="http://sp1.local.test/Shibboleth.sso/Metadata"
                                   backingFile="/opt/shibboleth-idp/metadata/sp1-metadata.xml">
        </metadata:MetadataProvider>
<metadata:MetadataProvider id="SP2_MD" xsi:type="metadata:FileBackedHTTPMetadataProvider"
                           metadataURL="http://sp2.local.test/Shibboleth.sso/Metadata"
                           backingFile="/opt/shibboleth-idp/metadata/sp2-metadata.xml">
        </metadata:MetadataProvider>
```

Figure 5.4: Metadata for service provider sp1 and sp2

To control the service provider session time -*as we discussed in chapter 4*- we can set it in the *relying-party.xml* file as a *maximumSPSessionLifetime* attribute in the profile login as in Figure 5.5.

```
<rp:ProfileConfiguration xsi:type="saml:SAML2SSOProfile"
                         includeAttributeStatement="true"
                         assertionLifetime="PT5M"
                         assertionProxyCount="0"
                         signResponses="never"
                         signAssertions="always"
                         encryptAssertions="conditional"
                         maximumSPSessionLifetime="30"
                         encryptNameIds="never" />
```

Figure 5.5: Session Service provider lifetime

5.3.1.2 Caching at IdP

The status listed in Figure 4.14 will be implemented as an enumeration object contains status and named LogoutStatus. We set the cache option on the identity provider (IdP) by changing the status of the shibboleth user from LOGOUT_SUCCEEDED to the OFF-LINE status.

www.manaraa.com

```
/**
 * Logout Status for a session participant.
 */
public enum LogoutStatus implements Serializable {
    LOGGED_IN, LOGOUT_SUCCEEDED, LOGOUT_FAILED,
    LOGOUT_TIMED_OUT,OFF_LINE
}
```

Figure 5.6: LogoutStatus enumeration

When User's status became OFF-LINE status, the content of the last session will exist until his new request for any new educational services occurs. For example, when a user logged out of the whole federation system, but he want to get some information back from the last session.

5.3.2 Service Provider Configuration

At this section we explain two parts of configuration and its implementation, the first part will talk about the logout configuration at service provider and the second one will be for caching configuration.

5.3.2.1 Logout Configuration at SP

The educational resources are hosted as service providers. For example if we have two different educational resources, then two service providers are mapped to them. Any request and use of those resources can be done through the service provider.

In our thesis implementation we work on two service provider; first service provider is *sp1.local.test* and the second one is *sp2.local.test*.

www.manaraa.com

The configuration file of service provider is named *shibboleth2.xml* and is located under the path *shibboleth-sp/etc/shibboleth.* We define the services sp1, sp2 in the configuration file as in Figure 5.7 by *Site* tag.

```
<Site id="854781" name="sp1.local.test" />
<Site id="854782" name="sp2.local.test" />
```

Figure 5.7: Define service provider sp1 and sp2

The secure content of the educational resources has to identified in the configuration file and define which authentication type are required as in Figure 5.8.

```
<Host name="sp1.local.test">
    <Path name="secure" authType="shibboleth" requireSession="true"/>
</Host>

<Host name="sp2.local.test" applicationId="sp2">
    <Path name="secure" authType="shibboleth" requireSession="true"/>
</Host>
```

Figure 5.8: Define secure path of the content of sp1 and sp2

Other important part of configuration is to set the parameter of the session initiator of the shibboleth authentication process, we define the URL of the identity server and the type of the exchanging message as a SAML2 as in Figure 5.9.

www.manaraa.com

```
<SessionInitiator type="Chaining" Location="/Login" isDefault="true" id="Login"
                  entityID="https://idp.local.test/idp/shibboleth"  >
    <SessionInitiator type="SAML2" template="bindingTemplate.html" />
    <SessionInitiator type="Shib1"/>
</SessionInitiator>
```

Figure 5.9: Session initiator definition

In Figure 5.10, the logout initiator will be defined in the location is GLogout and the type of the exchange of the request and response between the service provider and the identity provider as *http-redirect*.

```
<LogoutInitiator type="Chaining" Location="/GLogout" relayState="cookie">
  <LogoutInitiator type="SAML2"  outgoingBindings="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" />
  <LogoutInitiator type="Local"/>
</LogoutInitiator>
```

Figure 5.10: The logout initiator definition

5.3.2.2 Caching at SP

Refer to procedure of logout process in Figure 4.7, after the user had logged out globally from the federation system, the content of the open tabs in the browser will be cached by enabling that property at the sp level as in Figure 5.11.

www.manaraa.com

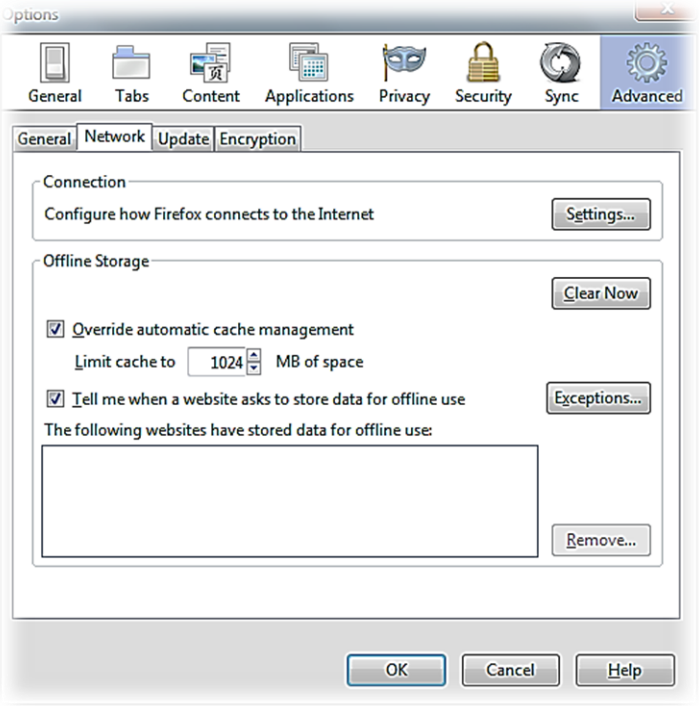

Figure 5.11: Enabling cache property in Firefox browser

This option is a client-based setting, so we have not any guarantee in the client side to enable browser caching. In addition, the variety of the browsers specifications differs for each other.

5.4 Achievement Requirement

In this section, the use cases based on our approach will help to clarify the concepts involved in the model architecture and emphasize the potential advantage of the global logout.

Use case 1. When the user wants to attempt the library services and student registration services, he/she will open two tabs with two sessions. During that he/she can use any or both resources, according to his/her needs. When he/she wants to leave all of them, he will click on a button named *'logout'* in library portal. As a rule of logout thinking

www.manaraa.com

in the mind of the user is to leave all resources at once. So the global logout link/button is a suitable link in that case with a full meaning of the atomic logout from both library and registration portal.

Use case 2. When the user wants to save his/her last state of the previous session, the identity provider has to saves his/her state. In the shibboleth, the users are stateless. Our enhanced IdP of shibboleth can save the state of users as the one of; OFF-LINE, LOGGED_IN, LOGOUT_SUCCEEDED, LOGOUT_FAILED, LOGOUT_TIMED_OUT.

Use case 3. When the user wants to leave all resources, he/she needs feedback information confirmed that logout process which makes the user convinced to go out of the federation system.

We will list which requirements of the e-learning system (14) achieved in the model used for educational resources, so we get improvement in the requirements which are access transparency, dynamic reconfiguration, awareness, and visibility which located in the Table 4 and explained later in subsections.

Fulfillment of Requirements By Educational Model By Enhanced Shibboleth														
	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14
E-Shibboleth	√	√	X	√	√	x	√	√	√	√	√	√	√	√

Table 4: Requirements achieved by enhanced model

www.manaraa.com

5.4.1 Access Transparency Requirement

The definition for that requirement "it should be possible for the users to acting with multiple applications as a one either in logon or logout phase", the login are accomplished in single sign-on approach which achieve the first part of the requirement, but the second part was not in the previous model for the federation models as we mention before in the chapter 3.

The modification in the logout process and make it as a global logout either in the identity provider side or in the service provider side. The single link in each service provider will be support the access transparency as in Figure 4.11 which display to the users who service providers are logged out.

The *use case 1* can explain the achievement of the access transparency, when the user leaves all resources by one click/button in either the library or registration portal.

5.4.2 Dynamic Reconfiguration Requirement

The Lack of a single log-out mechanism in Shibboleth (17) implies that is not easy to know if the user is still logged in the federation. The session status can provide the information required for the users to get be confirmed. For example, The URL of Status of service provider sp 1 is *http://sp1.local.test/Shibboleth.sso/Status* which returns the status of the logout of the services providers.

www.manaraa.com

The *use case 2* can explain the achievement of the Dynamic Reconfiguration requirement, when the user's status was tracked and modified by the IdP according to his/her actions.

5.4.3 Awareness Requirement

In Shibboleth the IdP may be deceived by the sessions opened while its users were log out from application. We work on that to destroy the sessions of the service providers of the users. After that the IdP will be awareness of the users tracking lifetime.

The *use case 2* can explain the achievement of the Awareness requirement, when IdP be know which user are really connected and which are not. The modification in the status implementation in the IdP helps the tracking of the users in the IdP.

5.4.4 Accessibility and Visibility Requirement

We work on the web interface which support the usability of the model, the visibility is accomplished when the user can see what he opened while he logged-out until the new request, which is more useful and familiar to users as in Figure 4.11.

The status of the user will be saved for him by a cached copy of the current opened resources in the model after enabling the caching property of the browser as in Figure 5.11.

www.manaraa.com

So we can summarize that our Educational Federation Model has fulfilled that requirement; Users can manipulate the session in order to adopt them to have been fully logout.

No	Requirements	Shibboleth	Enhanced Model with Shibboleth
1	Access Transparency	Not Qualified	Qualified
2	Dynamic Reconfiguration	Not Qualified	Qualified
3	Awareness	Not Qualified	Qualified
4	Accessibility	Not Qualified	Qualified

Table 5: Achieved requirements of our the secure model

www.manaraa.com

Chapter 6 – Conclusion

In this thesis, we have designed and implemented the secure educational resources model based on shibboleth identity provider which was enhanced to be very effective solution to the problem of fulfillment of the security requirements. The system which consisted of web interfaces to the educational resources, hosted as a service provider and identity server which responsible for authentication in login and logout.

Educational federation model has some specification differs than others in implementations according to the nature of the consumers. Previous federated SSO systems were in payment operations and nation elections are in single session, but in the educational environment where the user can open many parallel sessions at one time and serve the resources he authenticated to access. Nowadays, the integrity of the systems under the services in the federations requires infrastructure for mange the sessions and users.

Our model consists of three layers as the following; first layer is responsible for contacting with users with more flexibility and accessibility in friendly feedback web interface, the second layer is the Identity Provider of shibboleth which is the responsible for login and logout process, and the third layer is for storing the required resources and services by the users. Our model brings together all of the elements of the security mix to achieve the e-learning objectives by delivering to users what they want and need.

The web interface is the interface for the educational resources; users can login from the login window when they access the secure content. Logout occurs by users when they

www.manaraa.com

click the global logout link. Finally the last message is a logout feedback about the successful logged out resources by the user.

The Identity Server which has a modification for handling the sessions by accessing and destroying the SessionManager object. Also setting the session's lifetime of the user and initiating the login parameter to be with cookies.

A global logout have to be in the future work as an administrative logout with enabled tool to monitoring the federation systems. If a user’s session is compromised or for some other reason an individual user needs to be kicked off, administrative logout could be used to terminate IdP and SP sessions without restarting or otherwise clearing all of the IdP sessions. However, none of the implementations offer administrative logout, so it is an important point of research.

As a final conclusion, the proposed model has been a successful educational web model for manage the user's requests and provide it as a secure resource. The model is very scalable and extremely effective in memory utilization, and supports large volumes of traffic.

www.manaraa.com

References

1. Website of the Moodle project. [Online] http://moodle.org/.

2. Web site of the Blackboard project. [Online] http://www.els.qut.edu.au/blendedlearning/blackboard/upgrade2011/.

3. Web site of the dotLRN project. [Online] http://openacs.org/projects/dotlrn/.

4. Web site of the Sakai project. [Online] http://sakaiproject.org/.

5. *SAML Based Unified Access Control Model for Inter-Platform Educational Resources.* Shang, Chaowang, et al., et al. s.l. : IEEE International Conference on Computer Science and software Engineering, 2008. 978-0-7695-3336-0/08.

6. *Federation Paments using SAML Tokens with Trusted Platform Modules.* Lutz, David J. 2009 : IEEE. 1-4244-1521-7/07.

7. *Interoperability between heterogeneous federation architectures: Illustration with SAML and WS-Federation.* Mika¨el Ates, Christophe Gravier, Jeremy Lardon, Jacques Fayolle, Bruno Sauviac. s.l. : Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, 2008. 978-0-7695-3122-9/08.

8. Trusted Computing Group. [Online] http://www.trustcomputinggroup.org.

9. *Proposal of Delegeation Using Electronic Certificates on Single Sign-On System with SAML-Protocol.* Komura, Takaaki, et al., et al. s.l. : IEEE Ninth Annual International Symposium on Applications and the Internet, 2009. 978-0-7695-3700-9/09.

10. *An SSO-capable Distributed RBAC Model with High Availability across Administrative Domain.* Juntapremjitt, Sekpon, Fugkeaw, Somchart and Manpanpanich, Piyawit. s.l. : IEEE 22nd Intenational Conference on Advanced Information Networking and Applications , 2008. 978-0-7695-3096-3/08.

11. SANS Institue 2005. *Secure Implementation of Entrprise single sign-on product in an organization.* s.l. : Ravikanth Ponnapalli, 2004.

12. *A model of Unit-Authentication Single Sign-On Based on SAML underlaying Web.* kaixing, Wu and xiaolin, Yu. China : IEEE International Conference on Information and Computing Science, 2009. 978-0-7695-3634-7/09.

13. yale university - Technology & Planning. *Secure Single Sign-On.* s.l. : Yale University, 2011.

14. T, klobucar. Requirments collection and analysis with focus on privecy and security issues. [Online] 2008. http://isotc.iso.org/livelink/livelink?func=ll&objId=5906025&objAction=browse.

www.manaraa.com

15. OAuth Community Site. [Online] http://oauth.net/.

16. Delegation Permit. [Online] www.epa.gov/region9/air/permit/permitdelegation.html.

17. Internet 2. Shibboleth web site. [Online] http://shibboleth.internet2.edu.

18. OpenSSO Web site. [Online] http://java.net/projects/opensso/.

19. *Please permit me: stateless delegated authorization in mashups.* Hasan R, Conlan R, Slesinsky B,Ramani N,Winslett M. s.l. : Annual computer security applications conference (ACSAC), 2008.

20. Internet2 web site. [Online] http://www.internet2.edu.

21. Security Assertion Markup Language. [Online] http://www.oasisopen.org.

22. Security Assertion Markup Langauge(SAML) . [Online] Cover Pages Technology Reports, 2010. http://xml.coverpages.org.

23. *Reverse OAuth:Asolution to achive delegated authorization in single sign-on on e-learning systems.* Gonzalez, Jorge Fontenla, Rodriguez, Manuel Caeiro and Nistal, Nartin Llamas. 28, spain : Elsevier - Computers & Security, 2009, Vols. 843-856.

24. The difficulties of Single Sign-On . [Online] http://spaces.internet2.edu/display/SHIB2/SLOIssues.

25. Cramton, James. Shibboleth and Application Logout Best Practices. [Online] https://wiki.brown.edu/confluence/display/CISDOC/Shibboleth+and+Application+Logout+Best+Practices.

26. Sun Microsystems Web site. [Online] http://www.oracle.com/us/sun/index.htm.

27. Shibboleth Instalation. [Online] https://wiki.shibboleth.net/confluence/display/SHIB2/Installation.

28. Oracle Corporation. *NetBeans.* [Online] http://netbeans.org/community/releases/71.

29. Canonical Ltd. *Ubuntu Linux.* [Online] http://www.ubuntu.com/.

30. Sun Microsystems. *Sun Developer Network (SDN).* [Online] http://java.sun.com/products/jsp/.

31. Microsoft Server 2003 Enterrise Edition. [Online] http://www.microsoft.com/.

32. ASP.NET. [Online] http://www.asp.net/.

33. Netbeans. Netbeans. [Online] http://netbeans.org/community/releases/71.

34. *Webroot Security.* [Online] http://www.webroot.com/.

www.manaraa.com

35. Wadner, B and Pfitzmann, B. *Token-based web Single Sign-On with Enabled Clients.* s.l. : IBM Research Report, 2009. RZ 3458(93844).

36. *Certificate-based access control for widely distributed resources.* W. Thompson, W. Johnston, S. Mudumbai, G. Hoo, K. Jackson, and A. Essiari. USA : USENIX, 1999. Proc. of the 8th USENIX Security Symposium.

37. *Secure web services using two-way authentication and three-party key establishment for service delivery.* Song Han, Tharam Dillon, Elizabeth Chang, Biming Tian. 55 (2009), s.l. : Journal of Systems Architecture, 2009, Vols. 233-242.

38. Shubho, Al-Farooque. Single Sign on (SSO) FOR Cross-Domain asp.net Applications. *The Code Project - Web Development.* [Online] 2010. http://www.codeproject.com/kb/aspnet/CrossDomainSSOModel.aspx.

39. Scambray, J. *Hacking Exposed: Network Security Sercets & Solution.* s.l. : McGraw-Hill, 2001.

40. *Secure Delegation Model based on SAML in Ubiquitous Environments.* Kim, Kyu II, Lee, Hae Kyung and Kim, Ung Mo. s.l. : IEEE International Conference on Information Security and Assurance, 2008. 978-0-7695-3126-7/08.

41. *Analysis and Design of Duplex Signature Based on SAML Token.* Kim, K, Lee, H and kim, U. s.l. : IEEE, 2009.

42. Kearney, Paul. *Message level security for web services.* s.l. : Information Security Technical Report, 2005.

43. *An SSO-caplable Distributed RBAC Model with High Availability across Administrative Domain.* Juntapremjitt, Sekpon, Fugkeaw, Somchart and Manpanpanich, Piyawit. s.l. : IEEE International Conference on Advanced Information Networking and Applications, 2008. 978-0-7695-3096-3/08.

44. *Authorization- Authentication Using XACML and SAML.* Jake Wu, Panos Periorellis. s.l. : School of Computing Science, Newcastle University, 2005.

45. *Intelligent security and access control framework for service-oriented architecture.* Hany F. EL Yamany, Miriam A.M. Capretz, David S. Allison. 52, s.l. : Information and Software Technology 52, 2010, Vols. 220–236.

46. *Adding SAML to Two-Factor Authentication and Single Sign-On Model for Dynamic Access Control.* Fugkeaw, Somchart, Manpanpanich, Piyawit and Juntapremjitt, Sekpon. s.l. : IEEE International Conference on Information Systems, 2011. 1-4244-0983-7/07.

47. *Using SAML and XACML for Complex Authorisation Scenarios in Dynamic Resource Provisioning.* Demchenko, Yuri, Gommans, Leon and Laat, Cees de. s.l. : Reliability and Security (ARES'07), 2007. Second International Conference on Availability.

www.manaraa.com

48. Beznosov, Konstantin. *Introduction to web swervices and their security.* s.l. : Information Security Technical Report, 2005.

49. Web Service Development using SOAP. [Online] http://www.soapuser.com.

50. Introduction to single sign-on. [Online] http://www.opengroup.org/security/sso/sso_intro.htm.

51. *An Extended XACML model to ensure secure information access for web services.* 83 (2010), s.l. : Journal of Systems and Software, 2010, Vols. 77-84.

52. Amazon. *Amazon Elastic Compute Cloud (Amazon EC2).* [Online] Amazon. http://aws.amazon.com/ec2/.

53. *SimpleSAMLphp.* [Online] http://simplesamlphp.org/.

www.manaraa.com