AAI – Authentication and Authorization Infrastructure

System and Interface Specification

Document management

Version/status:	1.0 / final	
Date:	15-JAN-04	
Author(s):	Christoph Graf	SWITCH
	Ueli Kienholz	SWITCH
	Thomas Lenggenhager	SWITCH
	Marc-Alain Steinemann	University of Berne
	André Redard	at rete ag
	Daniela Isch	at rete ag
File name:	AAI_System_Specs_v10.doc	
Replacing:	0.6 / 22-DEC-03	
Approved by:		

Table of Content

Figures

Tables

1. Introduction

A solution to the problem of inter-organizational authentication and authorization is the implementation of an Authentication and Authorization Infrastructure (AAI). The purpose of this document is to define a generic model of the AAI and to give an overview of the implementation based on the Internet2 middleware Shibboleth[1]. It is shown how organizations can integrate their Authentication System and User Directory within the AAI. The document describes three integration methods for resources: by means of Shibboleth (native), the AAI Proxy and the AAI Portal. Further information about the implementation of the AAI can be found in the AAI deployment guides[2].

2. Overview

2.1 AAI Model

The core functionality of an AAI is to tightly couple together the three basic interactions between a user, his or her home organization and a resource during the authentication and authorization process. These three basic interactions are:

1. user authentication, which is always carried out by the user's Home Organization;
2. access request; and
3. delivery of authorization attributes from the Home Organization to the resource.
   The set of authorization attributes which is transmitted to an access control manager has to be configurable and extendible, depending on the needs of the Resource Owner and respecting the restrictions from the data protection law.

In order to describe the functionality of the AAI, the following generic model has been developed:

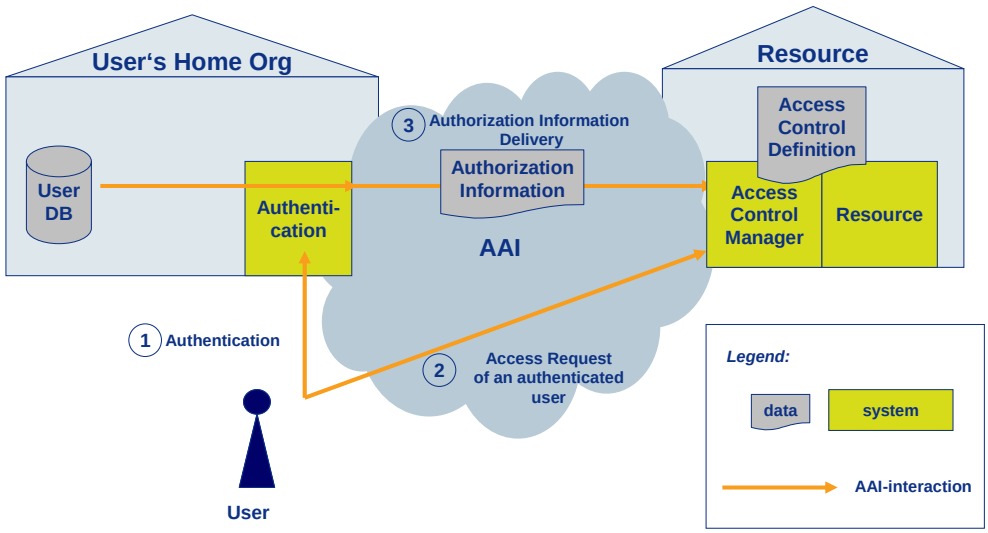

Figure 1: Generic functional model of an AAI

[1] http://shibboleth.internet2.edu
[2] http://www.switch.ch/aai

The terms introduced in Figure 1 are defined as follows:

Name	Description
(User's) Home Organization	Representative of a user community, e.g. universities, libraries, university hospitals etc. • registers its users and stores information about them • is able to authenticate its users
User	Registered member of a Home Organization
User Directory	Directory or database storing information about a registered user, maintained by the Home Organization
Authentication system	System which can authenticate a previously registered user
Resource	Application, web site
Resource Owner	Entity owning a resource and offering resource access to users
Access control manager	Gatekeeper functionality of the resource which grants or denies access to the resource based on the access control definition and the authorization attributes retrieved
Access control definition	Configuration parameters used by the access control manager implementing the access control policy
Authorization attributes	User data needed for access control decisions

After having received the authentication acknowledgement and the authorization attributes from the user's Home Organization, the access control manager, on behalf of the Resource Owner, can decide whether to grant or deny access to the resource.

2.2 Resource Types

The goal of the AAI project is to offer authentication and authorization functionality to a variety of different types of web resources. These resources can be categorized as follows:

Type	Access policy	Examples
Type A unpersonalized web resources	Access control policy based on group membership attributes	• intranet web servers
Type B personalized web resources	Access control policy based on individual and group membership attributes	• discussion forum • web mail • student administration
Type C unpersonalized "black box" web resources with proprietary access control	Access control policy based on group membership attributes	• 3rd party content providers (libraries)
Type D personalized "black box" web resources with proprietary access control and user administration	Access control policy based on individual and group membership attributes	• e-learning platforms • standard applications

Chapter 3.3 describes how these resource types can be integrated within AAI.

2.3 Shibboleth

During the previous phases of the AAI project, Shibboleth[3] has been selected as the preferred framework for the Swiss AAI. Shibboleth is a project of Internet2/MACE (Middleware Architecture Committee for Education)[4]. It aims to develop an architecture for standard-based vendor-independent web access control infrastructure that can operate across institutional boundaries.

The requirements on which Shibboleth was designed are documented in http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-requirements-01.html

2.3.1 Architecture

The primary design principles for Shibboleth are:

- no single central piece of infrastructure required, scalable;
- data protection and privacy are important for Shibboleth; and
- the user is guided by 'HTTP redirect' from the resource to the authentication server and back to the resource for the authorization.

A detailed description of the Shibboleth architecture can be found in *Shibboleth-Architecture Draft v05* [ShibArch].

Shibboleth uses a federated administration; a Resource Owner leaves the administration of user identities and attributes to the user's Home Organization, which is also responsible for providing attributes about a user (possibly but not necessarily including a username) that the Resource Owner can use in making an access control decision when the user attempts to use a resource. Users are registered only at their Home Organizations, and not at each resource.

Shibboleth is a system for securely transferring attributes about a user from his or her Home Organization to the site of the Resource Owner, provided the resources are accessible via standard web browsers. In addition, Shibboleth enables the users to decide which information about them gets released to which site. The users therefore have to balance access and privacy.

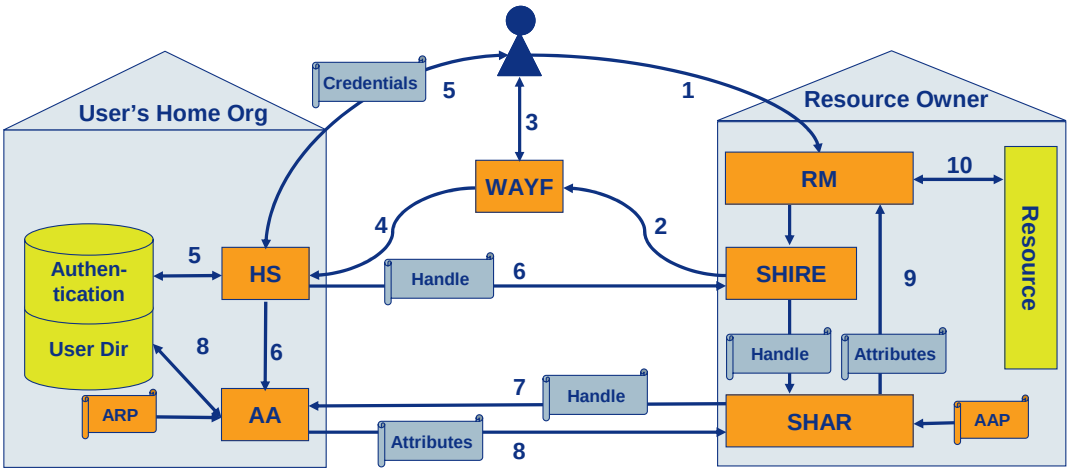

Figure 2: Shibboleth interactions

[3] http://shibboleth.internet2.edu

[4] http://middleware.internet2.edu/MACE/

The major components of Shibboleth are:

Name	Description	Operated by
HS	Handle Server Authenticates a local user according to the methods of the Home Organization and provides an opaque handle identifying the user.	*Home Organization*
AA	Attribute Authority Retrieves the attributes which a user allows to be given to a resource (according to the Home Organization's and user's Attribute Release Policy) and passes them to the SHAR on behalf of the resource.	*Home Organization*
SHIRE	Shibboleth Indexical Reference Establisher Makes sure that the resource gets a 'pointer' (handle) back to the user without requiring more knowledge about a user. In case it is missing it refers the user via the WAYF server back to his/her HS to get one.	*Resource*
SHAR	Shibboleth Attribute Requester Contacts the AA to fetch the available attributes describing the user and verifies them based on Attribute Acceptance Policy (AAP). Passes attributes on to RM.	*Resource*
RM	Resource Manager Decides on access to the resource based on the information received and where necessary the information about earlier sessions of the same user.	*Resource*
WAYF	Where Are You From Server Redirects the user back to the HS at his/her Home Organization. At least one WAYF server is needed, but it may be replicated as desired.	*Service Provider or any Organization*

2.3.2 Example of Shibboleth Usage

Figure 2 shows the basic interactions between the components of Shibboleth. A user U, affiliated to the Home Organization O, wants to access a web-based resource R located at some remote site.

- U connects with his or her web browser to the web site R (1). The server R does not detect the required authorization information and redirects U (2) to the 'Where Are You From' web server WAYF. The URL of R gets passed along.
- On WAYF, U selects his/her Home Organization O from a list of organizations participating in Shibboleth (3). WAYF redirects U (4) to the web server HS (Handle Service) located at the Home Organization O. The URL of R gets passed along again.
- U authenticates him-/herself according the local rules and methods towards HS (5). Once authenticated, HS generates an opaque handle H for the user U. H is the authentication info U needs to present to R. U gets redirected to R (6).
  R sends handle H together with the URL of R (7) to the Attribute Authority (AA) located at the Home Organization O.

AA checks which Attribute Release Policy (ARP) of user U applies to resource R. AA returns the attributes it is allowed to send to R (8).
Within R the attributes retrieved are verified based on the Attribute Acceptance Policy (AAP) and get passed to the Resource Manager RM (9) that decides on providing access.

- U gains access to the resource (10)

2.4 Terminology

Within the AAI project we use an architecture-independent terminology describing the components involved in authentication and authorization. Now that we selected the architecture we will have to get used to Shibboleth architecture-specific terms as well.

We do want to stick to the architecture-independent terms when discussing concepts, policies and rules; however, when talking about implementation we intend to use the architecture-specific terms from Shibboleth.

AAI term	**Shibboleth term**
Home Organization	Origin side
User Directory	LDAP
Resource Owner	Target side
Access Control Manager	Resource Manager

3. Components and Interfaces

The details regarding the Shibboleth implementation as explained in this chapter are based on Shibboleth version 1.1.

3.1 Components operated by Service Provider(s)

3.1.1 Component Overview

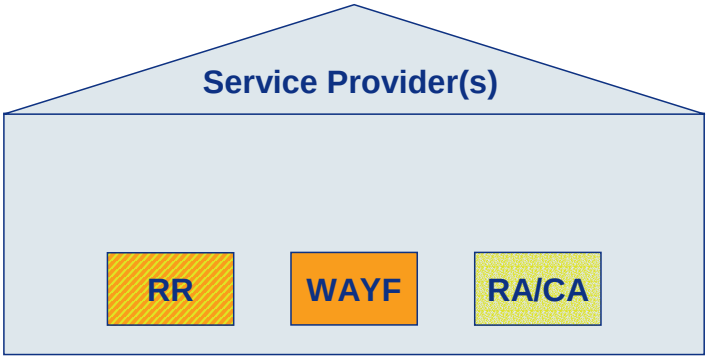

Figure 3: Inter-organizational components

Basic inter-organizational components, operated by Service Provider(s)

- Shibboleth WAYF (Where Are You From Server)
  It provides the user with a list of Home Organizations from which he or she selects the correct one and gets redirected back to the HS at his/her Home Organization.
  At least one WAYF server is needed, but it may be replicated as desired. However, if replicated, the list of supported Home Organizations has to be well synchronized and the user interfaces should be aligned.
  The WAYF is implemented as a Java servlet and requires an Apache web server as well as the Tomcat servlet container.
  Communication with the WAYF *must* be secured by using https.
- Resource Registry (RR)
  SWITCH will implement a Resource Registry (RR), where Resource Owners will be able to register their resources. The purpose of the RR is
  - to ease the administration of the attribute release policy (ARP) by administrators of Home Organizations and end-users
  - to increase the transparency and the trustworthiness of resources
  - to tie the resource into the Federation

  The RR will include at least this information:
  - Resource name
  - Description
  - Host(s)
  - URL(s)
  - Contact person(s)
  - Contact address(es)
  - Organization
  - Organizational unit
  - Mandatory attributes
  - Optional attributes

- Home Organizations allowed to access
- Legal status (resource of SWITCHaai federation member, SWITCHaai federation partner, none), see [AAIPol] for further information about the SWITCHaai federation

Only authorized administrators of federation members will be able to set a legal status other than "none".

The design and implementation of the RR has to be coordinated with the development of future releases of Shibboleth, since Shibboleth plans to implement an ARP Editor and to store further information about resources, so called "target metadata".

- RA/CA
  An AAI task force has worked out the requirements and solutions for a certification infrastructure for the server certificates needed for AAI. The infrastructure will be available by February 2004.

3.1.2 Component Interfaces

- Service Provider ⇔ WAYF
  The Service Provider maintains a list of registered Home Organizations together with the respective HS URL which the WAYF uses to list the participating Home Organizations.
- Resource ⇔ WAYF
  Each resource has to configure the URL of the WAYF to which users should get redirected to in order to authenticate.
- Home Organization ⇔ Service Provider
  Each Home Organization has to register the URL of their Handle Server (HS) with the Service Provider
- ARP Management Tool ⇔ RR
  The ARP Management tools will use the database of the Resource Registry as input to be able to support the admin.

3.1.3 User Interfaces

- WAYF
  The end-user receives a web page from which he/she selects the appropriate Home Organization against which he/she wants to authenticate.
- Resource Registry
  A resource owner registers his/her resource via a web page.

3.2 Components operated by Home Organizations

3.2.1 Component Overview

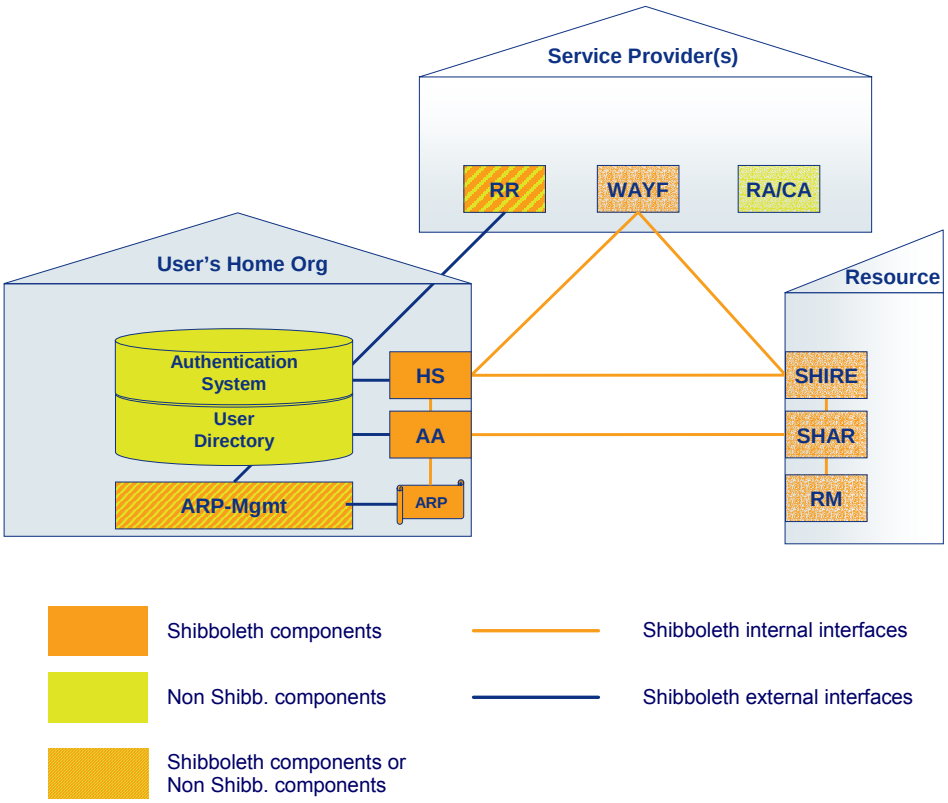

Figure 4: Components operated by the Home Organization

The following Shibboleth components have to be installed and operated by each Home Organization:

- Handle Server (HS)
  Once a user is properly authenticated, the Handle Server generates an opaque Handle with limited life-span which gets passed via the user's web browser on to the Shibboleth SHIRE component installed at the resource.
  The Handle Server keeps the generated handles during their life-spans together with a reference to the user in memory or within a CryptoHandleRepository.
  The Handle Server is implemented as a Java servlet and requires an Apache web server as well as the Tomcat servlet container.
  Communication with the HS *must* be encrypted by using https.
- Attribute Authority (AA)
  The Shibboleth SHAR components running at resources contact the Attribute Authority to receive information about the user who wants to access the resource. This information is required by the resource for authorizing the user.
  The resource provides the handle it got from the Handle Server. The Attribute Authority checks for the validity of the handle provided, identifies the user corresponding to the handle, checks the Attribute Release Policy (ARP) for that user and the resource asking and decides which attributes it may release.
  The Attribute Authority is implemented as a Java servlet and requires an Apache web server as well as the Tomcat servlet container. The AA makes use of the Java Naming and Directory Interface

(JNDI) API to retrieve attributes, typically via the JNDI LDAP context factory.
Communication between SHAR and AA *must* be encrypted using https.

Components which will have to be integrated with Shibboleth:

- Authentication System
  A prerequisite for a Home Organization participating in the AAI is the availability of a centralized authentication system against which users authenticate themselves.
- User Directory
  The Attribute Authority needs a User Directory from which it can retrieve the information about an authenticated user to be released to a resource. Therefore, the User Directory has to be able to access the information about a user based on the identifier known by the authentication system, e.g. by using a common key.

Further components

- NTP Server (Network Time Protocol)
  Since the Shibboleth communication between Home Organizations and resources requires well synchronized time, use of NTP synchronization for all servers involved is required.
- ARP management tool
  The ARP is defined in an XML-formatted file and can be modified with any editor or XML authoring tool. The future major release of Shibboleth will include a tool for Home Organization administrators and user to manage their ARP. SWITCH will integrate this tool with the AAI specific Resource Registry (RR) or provide its own ARP management tool.

3.2.2 Component Interfaces

- WAYF ⇔ HS (Shibboleth internal)
  The WAYF redirects the user to the HS and passes the URLs of the resource and the SHIRE to the HS as part of the HS URL.
- Authentication System ⇔ HS
  Any of the authentication methods known by the Apache web server[5] can be used for user authentication before access to the Handle Server is permitted. Use of third-party authentication modules[6] is possible (e.g. LDAP, PAM, RADIUS, TACACS, Oracle, Notes etc.).
  In addition, end-user certificate authentication is supported by Shibboleth.
- HS ⇔ SHIRE (Shibboleth internal)
  The Handle Server HS send a SAML authentication assertion containing a handle back to the SHIRE.
- User Directory ⇔ AA
  The Attribute Authority is able to use an LDAP Directory for user attribute retrieval.
- HS ⇔ AA
  Shibboleth offers two methods for passing the handle information from the HS to AA: the MemoryHandleRepository or the CryptoHandleRepository, which uses symmetric encryption.

[5] http://httpd.apache.org/docs/howto/auth.html
[6] http://modules.apache.org

- AA ⇔ SHAR (Shibboleth internal)
  The AA sends the authorization attributes to the SHAR using the SAML protocol.
- Resource Registry (RR) ⇔ ARP management tool
  The user of the ARP management tool will be able to retrieve information about resources from the Resource Registry, especially the names of the attributes required by the resource.

3.2.3 User Interfaces

- Authentication System
  By using the Basic or Digest Authentication method supported by web browsers, the user interface for authentication is defined by the browser of the user.
  By using other authentication methods (e.g. Web SSO, Certificates, Smart Cards), the user interface is implementation specific.
- ARP Management
  The ARP Management tool will have a browser-based interface.

3.3 Components operated by Resource Owners

3.3.1 Component Overview

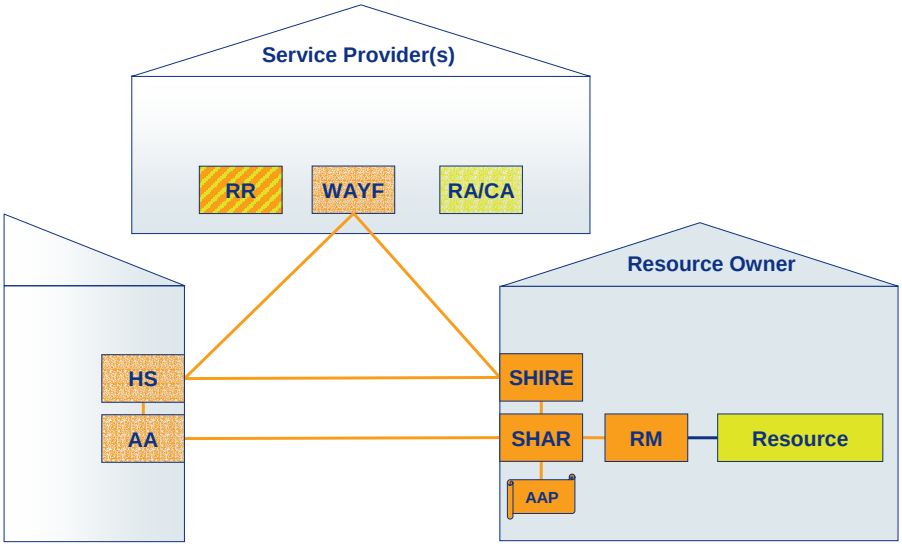

Figure 5: Components operated by the Resource Owner

The following Shibboleth components have to be operated by the Resource Owner

- Shibboleth Indexical Reference Establisher (SHIRE)
  SHIRE is the component detecting whether a request comes from an already authenticated user or whether the user has to be redirected first via WAYF to the Handle Server of the Home Organization. Once the user is authenticated, the handle received gets passed to SHAR for the attribute retrieval. SHIRE is available as an Apache module or a Microsoft IIS plug-in.
- Shibboleth Attribute Requester (SHAR)
  SHAR receives from SHIRE a handle, together with the URL for the Attribute Authority related to it. Out of band with the communication to the user, the SHAR now requests the user attributes to be released concerning the resource the user wants to access.

Once the attributes are received, they are checked against the Attribute Acceptance Policy for completeness and passed on to the Resource Manager.
SHAR is implemented as a standalone server process which has to be started before Apache starts.

- Resource Manager (RM)
  The RM protects the resource (web pages) and receives the attributes from SHAR. The RM is implemented as an Apache and an IIS module.
  The Apache RM makes the access control decision based on the received attributes using Apache mod_auth-style require-directives, which may either be in the context of .htaccess, server config, virtual host, directory, location or files.
  The IIS RM module supports the mapping of attributes via AAP files, but it does not support rule-based policies as the Apache module does and therefore cannot protect static content at this time.

Components which will have to be integrated with Shibboleth:

- Resource Type A: Unpersonalized web resource
- Resource Type B: Personalized web resource
- Resource Type C: Unpersonalized “black box” web resource
- Resource Type D: Personalized “black box” web resource

Apache or IIS based web resources of Type A or Type B can be integrated with Shibboleth, using the functionality of the RM to implement access control rules and to pass attributes to applications.

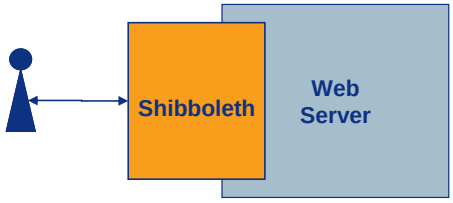

Figure 6: Shibboleth integrated web Apache or IIS based web resources

Resources of Type C and D can be integrated using one of the technologies described in chapter 4.

Further components:

- NTP Server (Network Time Protocol)
  Since the Shibboleth communication between Home Organizations and resources requires well synchronized time, use of NTP synchronization for all servers involved is required.

3.3.2 Component Interfaces

- SHIRE ⇔ WAYF (Shibboleth internal)
  The SHIRE redirects the user to the WAYF. Therefore, SHIRE needs to know which WAYF server to refer users to. The URL of the WAYF server has to be configured into SHIRE.
- HS ⇔ SHIRE (Shibboleth internal)
  The Handle Server HS replies with a SAML authentication assertion containing a handle.
  SHIRE uses a list of accepted Handle Servers: sites.xml.
  That list can either be stored locally to SHIRE or be retrieved automatically in regular intervals from a trusted source. Handles from Handle Servers not matching the sites.xml list will be rejected by SHIRE.

- SHIRE ⇔ SHAR (Shibboleth internal)
  SHIRE hands off the handle and the URL of the AA, received from the HS, to the SHAR. SHIRE and SHAR communicate over RPC or can use TCP sockets for specialized deployment behind firewalls.
- SHAR ⇔ AA (Shibboleth internal)
  The SHAR requests the authorization attributes from the AA using the SAML protocol.
- SHAR ⇔ RM (Shibboleth internal)
  Attributes get passed from SHAR to RM as environment variables.
- RM ⇔ Resource
  Authorization attributes can be passed along to the resource in the HTTP header. The mapping between authorization attribute and header name is configured in the AAP file.
  Optionally, the whole signed SAML answer, as received from AA, can be passed along to the resource. That way, a resource is able to analyze and evaluate the attributes according to its needs, provided it is able to parse the XML structure.

4. Resource Integration Tools

4.1 The AAI Proxy

During the pilot phase of the project it proved to be difficult to “shibbolize” some resources in particular resources that are not running on a supported web server. Therefore, SWITCH implemented an AAI Proxy which allows Shibboleth-protected access to legacy resources.

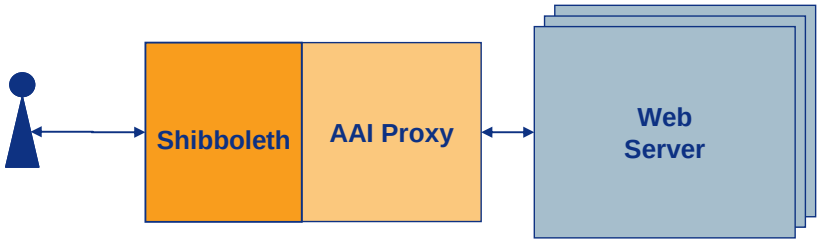

Figure 7: AAI Proxy architecture

The functionality of the AAI Proxy can be described as follows:

- User authentication and retrieval of authorization attributes via the Shibboleth system
- Mapping of authorization attributes to (group-related) credentials of the protected web server e.g. source IP addresses, username/password
- Passing through and modification of the HTTP(S) traffic between the user’s browser and the web server, e.g. transformation of IP addresses, host names, cookies, form fields (method POST)
- Session management:
  It must be made sure that subsequent requests to the resource still originate from the same user:
  - either the AAI Proxy performs this task by means of the built-in session management mechanism of Shibboleth
  - or the resource does it by means of a built-in session management mechanism in the resource.

A prototype version of the AAI Proxy is implemented in Perl and is available from SWITCH. It is based on open source code from the PAPI project. The tool will be improved in 2004.

4.2 The AAI Portal

The AAI Portal is designed to be a mediator between the mechanisms of the core AAI with the connected Home Organizations on one side and between protected resources on the other side. In the future, various community management features (e.g. messaging, discussion forums, chat) could be integrated. This chapter gives a short overview of the concept and the functionality of the AAI Portal.

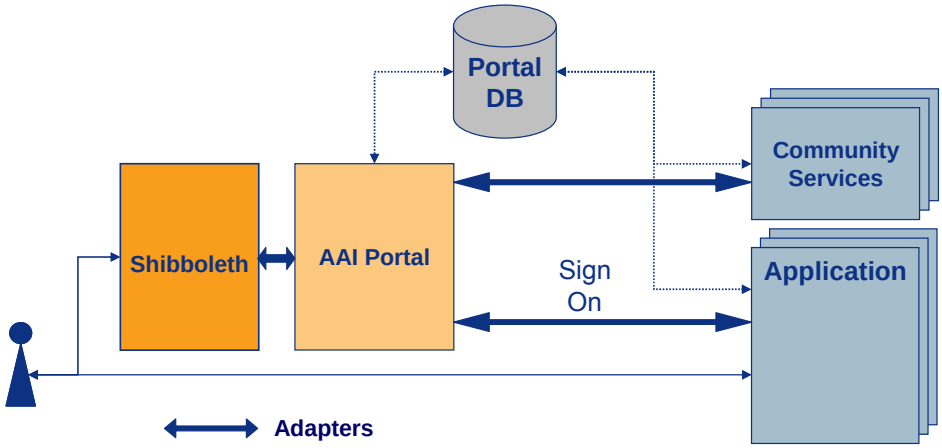

Figure 8: AAI Portal architecture

Users of the AAI Portal may have different roles. They act as portal administrator, resource administrator or resource user. Table 1 shows the most important functions provided to these users:

Portal Administrator	Resource administrators	Resource users
• Add or remove resources • Add or remove resource administrators • Specify SMS gateway • Specify e-mail gateway	• Define how the resource is integrated into the AAI Portal. • Define which way users are redirected to the resource. • Define which attributes a user must provide for getting access to the resource. • Specify additional (not yet existing) user attributes. • Define the resource's AAP. • Open or close resources for subscription • Accept or reject subscription requests. • Add or delete users to the AAI Portal.	• List all resources visible to them • List the set of already subscribed resources • Subscribe to and unsubscribe from resources • View their attributes the AAI has released to the AAI Portal • Enter additionally required information (attributes) for specific resources • Define which attributes are to be released to which resource

Table 1: Functions of the AAI Portal

The goal of the AAI developers was to make it easy to integrate resources, community management features and AAI components. Therefore, the AAI Portal offers the concept of adaptors. Today, the following adaptors are available:

Adaptor Type	Built-in Interfaces
AAI Adaptors	• Shibboleth
Resource Adaptors	• Simple HTTP Redirection • HMAC-based Authentication[7] • WebCT[8] Adaptor
Community management feature Adaptors	none

Table 2: AAI Portal – built-in adaptors

The development of the AAI Portal has been funded by the Swiss Virtual Campus (SVC). The software is under GNU General Public License (GPL). Further information and the software itself are available from http://aai-portal.sourceforge.net.

[7] see RFC 2104: HMAC - Keyed-Hashing for Message Authentication

[8] e-learning solution (http://www.webct.com)

5. References

[AAIPol]	SWITCH – AAI Service Agreement, Exhibit 3: AAI Policy
[ShibArch]	Shibboleth-Architecture DRAFT v05, Marlena Erdos and Scott Cantor, May 2, 2002 http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf