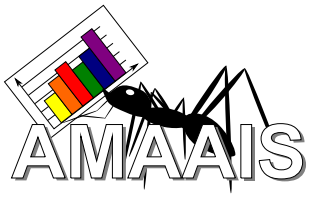

Accounting and Monitoring of AAI Services

Phase 1

Deliverable D1

AMAAIS Phase 1: Scenarios, Requirements, and High-Level Architecture

The AMAAIS Partners

University of Zürich (UZH), Switzerland
SWITCH (SWITCH), Switzerland
ETH Zürich (ETH), Switzerland

For more information on this document or the AMAAIS project, please contact:

Dr. Peter Racz
University of Zürich
Department of Informatics (IFI)
Communication Systems Group (CSG)
Binzmühlestr. 14
CH-8050 Zürich
Switzerland

Phone: +41 44 635 6702
Fax: +41 44 635 6809
E-mail: racz@ifi.uzh.ch

Document Control

Title:	AMAAIS Phase 1: Scenarios, Requirements, and High-Level Architecture
Type:	Public
Editor(s):	Peter Racz, Guilherme Sperb Machado, Martin Waldburger
E-mail:	racz@ifi.uzh.ch, machado@ifi.uzh.ch, waldburger@ifi.uzh.ch
Author(s):	Peter Racz, Guilherme Sperb Machado, Martin Waldburger, Daniel Meier, Patrik Schnellmann, Andy Zbinden, Matteo Corti
Doc ID:	D1
Delivery Date:	26. October 2009

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
0.1	2009-05-15	M. Waldburger	Initial version of template
0.2	2009-06-23	G. Machado	First contents (introduction, scenarios, and requirements)
0.3	2009-07-30	G. Machado	Included corrections/suggestions from AMAAIS members. Some refactoring on the organization of the document (sections). New figures expliciting accounting.
0.4	2009-07-31	P. Racz	Editorial updates and changes.
0.5	2009-08-24	G. Machado	Included corrections/suggestions from AMAAIS members. Major changes: architecture (new fig and text), scenarios (goals, pre-cond, and clarification on the descriptions)
0.6	2009-08-24	G. Machado	Included corrections/suggestions from AMAAIS members . Major changes: new figures, new subsection about scenarios with architecture components, and some modifications due to discussions from the past general meeting
0.7	2009-09-25	P. Racz	Included two new use cases and the overview of Shibboleth logging.
0.8	2009-10-05	P. Racz	Editorial updates and changes.
0.9	2009-10-16	G. Machado	Included corrections/suggestions from AMAAIS members (Patrik and from UZH side). Section Summary and Conclusions added, some modifications in the Functional and Non-Functional Req., figure fonts enhanced, and some editorial updates/changes.
0.10	2009-10-20	P. Racz	Included the executive summary. Updates in scenario 4 and 5, and some editorial updates and changes.
1.0	2009-10-26	P. Racz	Final version. Updates according to comments after internal review.

Legal Notices

The information in this document is subject to change without notice.

The Members of the AMAAIS Project make no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The Members of the AMAAIS Project shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Contents

1 Executive Summary

The goal of the AMAAIS (Accounting and Monitoring of AAI Services) project – a collaboration between the Communications System Group (CSG) at UZH, SWITCH, and ETH – is to extend the current Authentication and Authorization Infrastructure (AAI) with accounting and monitoring support, enabling inter-domain accounting and the management of the AAI. The AMAAIS project is structured into project phases. Phase 1 aims at collecting use case scenarios, requirements, and the high-level architecture, while phase 2 is focused on the fine design of the architecture and the implementation of the accounting and monitoring architecture.

This deliverable is the result of phase 1 of the AMAAIS project. Accordingly, this deliverable contains (1) a collection of use case scenarios for the accounting and monitoring architecture, (2) relevant accounting attributes for all scenarios, (3) functional and non-functional requirements of the architecture, (4) the high-level AMAAIS architecture, and finally (5) an overview of the logging capabilities of Shibboleth.

2 Introduction

A Shibboleth-based Authentication and Authorization Infrastructure (AAI) enables users to access different web resources in a common manner by providing a single-sign-on interface for login. The AAI makes use of the Federated Identity Management concept that allows the use of a single user identity for different services beyond the user's home institution domain. Such characteristic plays an important integration and organizational aspect for institutions: it not only eases the users' access to resources, but also makes the management process more convenient. Pfitzmann et al. [1] provides a comprehensive overview of available federated identity approaches and protocols. Within the scope of the AMAAIS project (Accounting and Monitoring of AAI Services) [2], Shibboleth is used and the considered type of federation includes primarily institutions of higher education in Switzerland. Shibboleth is based on the Security Assertion Markup Language (SAML) [3]. It is open source software and builds on OpenSAML [4], an open source implementation of SAML.

To better illustrate the mentioned aspect, basically, when a user at one institution tries to use a resource at another, the Shibboleth Identity Provider (IdP) authenticates the user and sends attributes about the user (*e.g.*, "employee at UZH", "faculty member", or "student enrolled in course X at UZH") to the Service Provider (SP), rather than making the user directly log in to its own institution to gain access to remote resources. The service provider uses these attributes to decide whether or not to grant access to the user. Only the SP knows the access rules to perform the authorization of the user, protecting user's anonymity in cases where identities are not necessarily to be revealed. Since the IdP can restrict the set of attributes to those attributes that are required to gain access on the SP, Shibboleth also has become an important tool in protecting identity. With respect to management, each resource decides who may have access to it, leaving the responsibility of managing user information entirely to each IdP (institutions). For these aspects, even more Shibboleth-based AAI systems are being deployed around the world, ranging from industry (*e.g.*, Microsoft) to educational institutions [5].

In 2001, the Swiss Educational and Research Network (SWITCH) [6] started a project for implementing an authentication and authorization infrastructure (AAI), enabling many Swiss universities to become part of a single federation. This project, called SWITCHaai [7], gave the possibility both to students and employees to access services at different universities by using a single username and password. However, the current version of Shibboleth, as well as the current deployment of SWITCHaai at Swiss universities, do not support accounting and monitoring functionality. The lack of both features in an integrated manner makes the process of controlling and managing the use of resources difficult.

Therefore, the goal of the AMAAIS project – a collaboration between the Communications System Group (CSG) at UZH, SWITCH, and ETH – is to extend the current AAI with accounting and monitoring support, enabling inter-domain accounting and the management of the AAI. AMAAIS is structured into two content-wise interrelated project phases. Phase 1 is primarily concerned with service-independent accounting and monitoring, as well as with service-dependent scenarios, while the main focus of phase 2 is on the refined design and implementation of the accounting and monitoring architecture. This deliverable is related to the results obtained in the first phase of the AMAAIS project. Accordingly, it

determines and documents (1) use case scenarios for the accounting and monitoring architecture (*cf*. Section 3), (2) suitable attributes for all scenarios description (*cf*. Section 4), (3) requirements for such an architecture (*cf*. Section 5), (4) it sketches the high-level accounting and monitoring architecture itself (*cf*. Section 6), and (5) it provides an overview of the logging capabilities of Shibboleth (*cf*. Section 7).

3 Scenarios for the Accounting and Monitoring Architecture

This section describes service-independent and service-dependent scenarios relevant to the AMAAIS accounting and monitoring architecture. The former is focused on monitoring, collecting, and visualizing usage statistics of the SWITCHaai (see Section 3.1), which is in fact a scenario not dependent of any specific service. While the latter are focused on the printing service (see Section 3.2), the SMS service (see Section 3.3), the SWITCHpoint video conferencing service (see Section 3.4), and the SWITCHcast streaming service (see Section 3.5), which are scenarios that depend on the mentioned services. The descriptions of the scenarios do not aim at providing technical details but they illustrate the accounting and monitoring functionality for the AAI and AAI-based services.

3.1 Service-independent Scenario 1: AAI Usage Statistics

The service-independent scenario considered for the accounting and monitoring architecture consists of the collection, processing, and visualization of authentication and authorization requests in the AAI – in other words, in the accounting of the AAI usage itself.

The benefits of having AAI usage statistics is to improve service quality of AAI, to ease the management of the whole environment, to better understand how SWITCHaai is used, and to possibly detect service misuses. For these purposes, data is collected in a distributed manner and aggregated (possibly in anonymized form) for further processing and visualization. Usage statistics, reports, or graphs are made available via a web front-end which may allow for user-configurable presentation, *e.g.*, with respect to covered time frame or considered provider domain. Access to these resources may be limited to authorized users only.

3.1.1 Description of the Current State

Currently, monitoring and accounting are not formalized or fully-integrated into the Shibboleth-based SWITCHaai. Some efforts were done in order to enable accounting into SWITCHaai scope [8], however the solutions were developed in an ad hoc manner. Besides that, such solutions did not aim to employ a general monitoring and accounting architecture for all institutions in the SWITCHaai, being not possible to measure, for example, an overall usage statistic for the federation.

Some solutions that are currently used:

- WAYF statistics interaction
- local statistics at IdP's based on log files

3.1.2 Goals and Pre-Conditions

Goal (user perspective):

- Any authorized user (administrator), anywhere inside the federation, may be able to consult AAI usage statistics through a Web interface.

Goals (system perspective):

- The system should generate different visualization schemes in order to present AAI usage statistics for users;
- Have a standardized interface to exchange usage statistic data.

Pre-Conditions:

- The system shall be able to collect relevant data related to the AAI environment.

3.1.3 Accounting and Monitoring Extension in the Scenario

Figure 1 describes the scenario with the accounting and monitoring extension which is responsible in this scenario to collect usage statistics related to the Shibboleth-based AAI environment. This figure shows the operator's point of view of the system. Here, the operator actor is a general term to describe an authorized user to access AAI statistics – he/she can be a technical specialist or a person that have an administrative role.

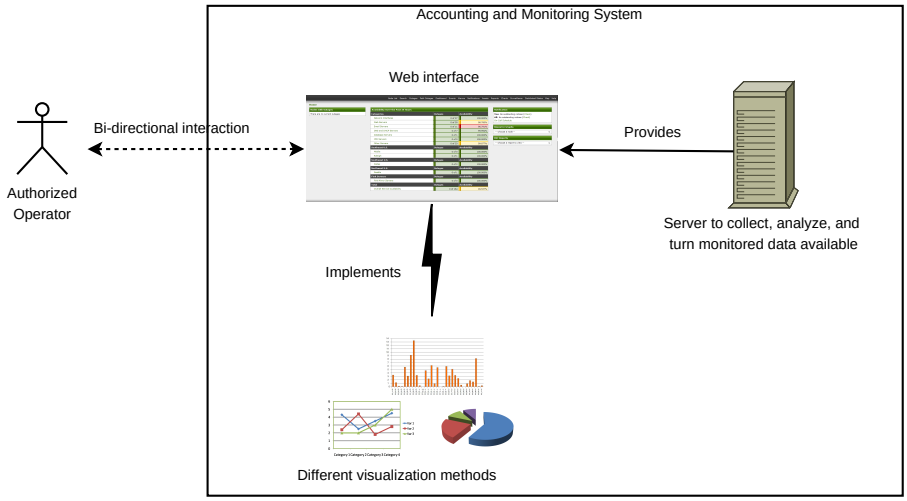

Figure 1: Operator's point of view on AAI usage statistics

The first aspect, and also one of the main points to be considered, is that an operator interacts with a web-based interface which provides statistical information. The access to the interface is made in an "anytime, anywhere" fashion and with certain restrictions (*e.g.*, authentication and authorization).

An implicit step here is the authentication to access the web interface. Access may be granted in different ways, depending on the respective infrastructure needs to set such configuration. For example, a single IP address inside an internal network might be allowed to access the usage statistic interface besides having a login. Alternatively, this resource could also be implemented using the Shibboleth-based infrastructure (as a service provider).

As shown in Figure 1, a server (located at an IdP, SP, and/or by SWITCH) is used in order to collect, analyze, and make monitored data available for operators. Such data contain relevant information from SPs and IdPs at the AAI environment, depending on the configuration of the accounting data exchange. Based on this data, different means to visualize AAI usage are made available through the aforementioned web interface. Visualization means are discussed in further detail in Section 3.1.4, where the identified set of requirements for usage statistics are presented as well.

Figure 2 presents a high-level view of the scenario. Firstly, a server should have access to the data from the AAI environment in an efficient and secure way. For reasons of security, encryption methods should be considered since sensitive data will be transmitted passing through different networks until it arrives at the final destination. For efficiency reasons, accounting data transmissions should be configurable since each domain has its own bandwidth priorities/policies. For example, IdP and SP data can be collected during the night sent as a batch, or the data can be transmitted on demand as it is generated (real-time).

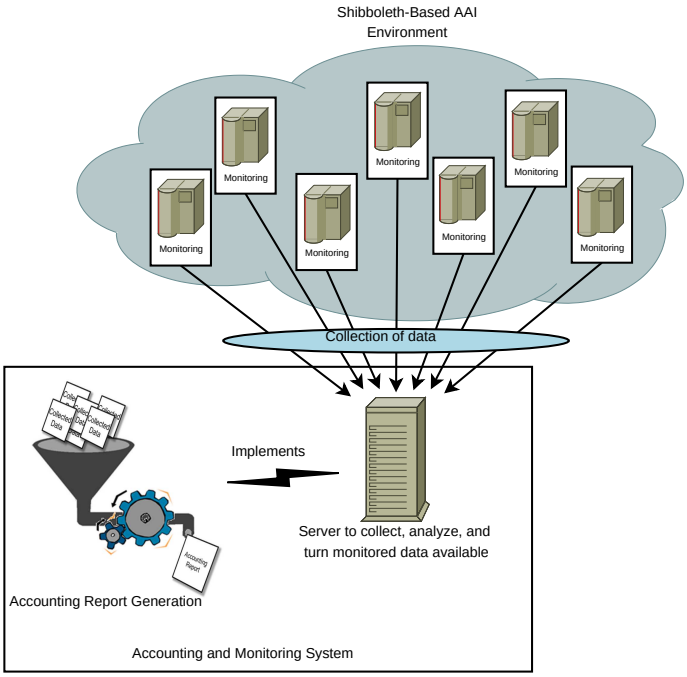

Figure 2: High-level system view related to usage statistics

Secondly, once monitored data is collected and available for further analysis, the accounting report generation is ready to be performed. This process takes AAI collected data as input and produces accounting reports. Such reports must follow a consistent and flexible

format which allows for information presentation in different ways. The accounting process must be decoupled from the visualization part of the system as well as with the data transmission. Having decoupled processes makes it easier to allocate resources, diagnose problems, and also to manage possible failures in a fault tolerance sense. Another issue is the standard to be used: the accounting process should have a documented standard to receive input files and also to produce the output. A common standard allows, for example, the interoperability between systems for future development of additional extensions.

Once available data of the AAI is quantified and analyzed, the server will produce reports which are formatted following those determined requirements as described in Sections 3.1.4.1 and 3.1.4.2, where overall or specific usage statistics for SWITCHaai are discussed, respectively.

3.1.4 Possible Usage Statistics

The collection and visualization of usage statistics address the requirements of the following user groups:

- on a federation level: selected IdP operators, as well as SWITCH
- on a federation member level: administrators of both identity providers and service providers

The main activities of the AAI, which are considered for monitoring, are the following:

1. user authentication at the IdP;
2. attribute release from an IdP for an SP;
3. authentication assertions and attributes received from an IdP on a SP;[1]
4. WAYF (Where Are You From) interactions; [2]
5. SP authorization. [3]

The measurement and evaluation of WAYF interactions has already been implemented by SWITCH and is not part of this requirement specification. In this document we assume that WAYF events are available and ready to be consulted. Moreover, the authorization decision based on AAI attributes (by means of rules) is mainly done by the resource itself

[1] This case will be interesting if a Service Provider provides services to users outside the SWITCHaai federation, where the IdP will not take part in the accounting and monitoring activities.

[2] The accesses on the WAYF are not needed to get the SWITCHaai federation statistics. Provided that the log data of all the IdP's in SWITCHaai are available, all the IdP-SP interactions are covered to generate usage statistics. The WAYF logs can be seen as a source of data complementary to the IdP's logs.

[3] The authorizations on Shibboleth SPs can happen on three different spots: The Shibboleth module in the web server (mod_shib for Apache, isapi_shib for IIS) may evaluate access rules but also the Shibboleth process itself and the Shibboleth protected application may render an authorization decisions. For each of these methods, the authorization decisions are not logged on a production system with a standard configuration.

and is not a core functionality of SWITCHaai. Then, for this reason, it is out of scope of the requirements specified here. It is important to note that authorization events are considered by the project, meaning that just the rules/decisions taken on each resource will not be covered here. All authorization events on each resource are essential to the accounting analysis and should be taken into account for the accounting and monitoring system.

In the following sections, basic usage statistic strategies are presented that can be inferred from the data collected at IdPs. Each strategy follows cases, which are classified in a top-down priority fashion. Cases are means of representing usage statistics.

The events shown in the statistics are accesses to SPs in SWITCHaai. An event can be defined as an authentication assertion sent from an IdP to a SP with or without prior authentication at the IdP. Other events such as specific types of transactions, or events in log files, may be added later for more detailed statistics, such as logs of unsuccessful authentications.

3.1.4.1 Overall SWITCHaai Usage Statistics

Federation member representatives and SWITCH would like to get overall statistics about the general use of SWITCHaai; either within an organization or across organizations. Therefore, two possible options for usage statistics are presented considering the complete SWITCHaai view.

- **Case 1**

  Number of events between organizational domains or within the same domain. The number of events is cumulated per home organization: all the SPs of one organization sum up to a total, *i.e.* all the SPs that belong to University of Zurich are consolidated in the same row (Table 1).

  The creation of these statistics requires additional information on top of the Shibboleth log files. The association of SPs to home organizations is needed for consolidation. This information can be retrieved from the SWITCHaai Resource Registry [9]. The Resource Registry is a database for the entities in the SWITCHaai federation. It contains technical and administrative information to manage and generate the federation metadata. To enable the usage of AMAAIS for other federations it would be beneficial not to require the Resource Registry and to resolve the association of SPs with Home Organizations from the metadata. This presumes that the federation metadata contains the necessary information about the association of an SP to a Home Organization.

  The statistics should be produced for different time frames: on a daily, monthly, quarterly, and yearly basis. Ideally, arbitrary time frames are possible.

- **Case 2**

  The SWITCHaai federation counts over 300 SPs and 40 IdPs by the beginning of May 2009. Usage can vary from very high to rather low load. Therefore, showing a full list with the events per service provider is not adequate. A list of the *top X* most used service providers is more useful.

Table 1: Overall SWITCHaai usage statistics, per home organization

		Identity Providers			
		uzh.ch	eth.ch	switch.ch	...
Service Providers (per home organization)	uzh.ch	32908	2118	97	...
	eth.ch	2024	56702	115	...
	switch.ch	482	1852	7984	...
	...	...	...	...	...

As an other case for the usage statistics mentioned before, the top X service providers in SWITCHaai is shown in Table 2.

Table 2: Overall SWITCHaai usage statistics, top X service providers

	SWITCHaai overall usage	
Service Providers (per Home Organization)	olat.uzh.ch	24093
	moodle.eth.ch	18031
	dokeos.unige.ch	5049
	ezproxy.eth.ch	3252
	exproxy.uzh.ch	2871
	moodle.unifr.ch	2732
	...	...

- **Case 3**

  As a combination of the usage statistics in case 1 and 2 above, the top X service providers could be shown per home organization, as shown in Table 3.

3.1.4.2 Statistics for a Specific Home Organization

Starting with the report shown in Table 1, the administrator of an IdP should be able to drill down and view the number of events for its home organization.

- **Case 1**

  Number of events at one IdP, per SP. As for the statistics mentioned above, different time frames shall be considered: daily, monthly, quarterly, yearly and ideally arbitrary periods (Table 4).

- **Case 2**

  Time series for one SP. For trend analysis, a time series of events is more suitable to visualize (Table 5 and Figure 3).

  The distribution of events over a specific time period. Time periods are the same as for the statistics listed above, the granularity should be defined appropriately (*i.e.* days for a month's period, weeks for a year's period).

Table 3: Top X service providers, per home organization

			Identity Providers			
			uzh.ch	eth.ch	switch.ch	...
Service Providers (per Home Organization)	uzh.ch	olat	25401	3856	32	
		ezproxy	1213	2341		
		www	4352			
	eth.ch	moodle	2024	36159	60	
		ezproxy		3452		
		net		3112		
	switch.ch	www	82	53	7984	
		smap	409			
		econf	302	395	581	
	...					

Table 4: Number of access requests per service for all users of an IdP

Service Providers	**IdP – https://aai-logon.switch.ch/idp/shibboleth**
https://www.switch.ch/shibboleth	70
https://wiki.switch.ch/shibboleth	21
https://bb.switch.ch/shibboleth	14
...	...

3.2 Service-dependent Scenario 2: Printing Service

This scenario includes the accounting for printer usage at universities or other institutions. The collection of accounting data regarding printing will allow administrators to get a better overview about the usage of printing infrastructure and also allows for the charging for printing at a later stage.

3.2.1 Description of the Current State

A printing service is currently implemented and operated by ETH [10] where it is currently used by its employees and students. The service is offered in an electronically prepaid fashion, requiring users to acquire credits (money) before printing their documents. However, the lack of an appropriate accounting and monitoring infrastructure makes, for example, service improvements difficult, since AAI operators do not dispose of detailed information about user behavior. Furthermore, the presence of accounting information may facilitate improvements toward a better general overview on what was consumed when a resource (in this case, printers) is shared.

Table 5: Time series of events, for a specific SP

Date	Number of requests
01.03.2009	34
02.03.2009	12
03.03.2009	2
04.03.2009	1
05.03.2009	57
06.03.2009	31
...	...
31.03.2009	17

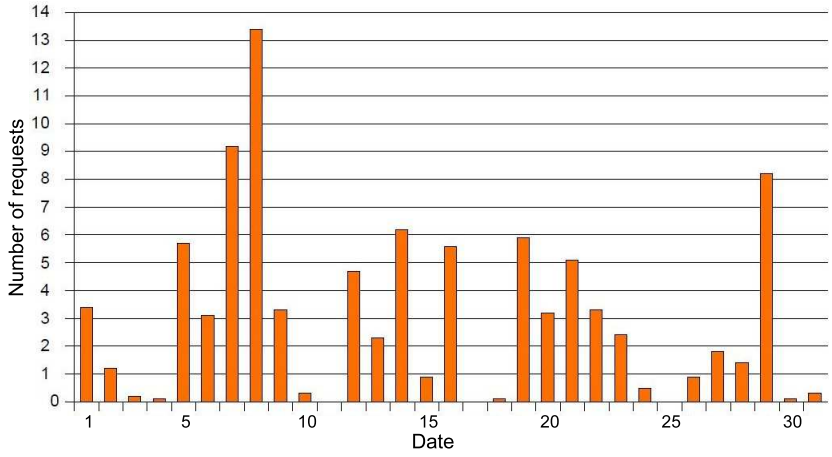

Figure 3: Time series of events, for a specific SP

3.2.2 Goals and Pre-conditions

Goals (user perspective):

- Any user in a federation should be able to print documents at any printer where he/she is authorized.
- Users should be able to see the status of their printing jobs in near real-time. Therefore, accounting should be performed in real-time to allow real-time billing.

Goals (system perspective):

- Based on the relevant set of specified rules and available user attributes, the system is able to take a decision to either restrict or grant access to a resource;
- Provide usage statistics about the printing service.
- The system keeps a record of all events relevant to this resource (*e.g.*, number of pages printed), in order to make the accounting a viable and consistent process;

- Send all monitored data to a data repository in order to concentrate all information for the accounting server (accounting process).

Pre-Conditions:

- Have an account at one of the IdPs in the federation;
- The user's IdP participates in the accounting activities;
- The user is authorized to print (*e.g.*, have enough credits);
- A valid print job file (PostScript or PDF).

3.2.3 Accounting and Monitoring Extension in the Scenario

The description of the scenario is divided in two parts. In the first part, a user's point of view of the printing scenario is depicted, where the interaction between user and system are expressed, while the second part is focused on presenting the system's point of view.

3.2.3.1 User Perspective

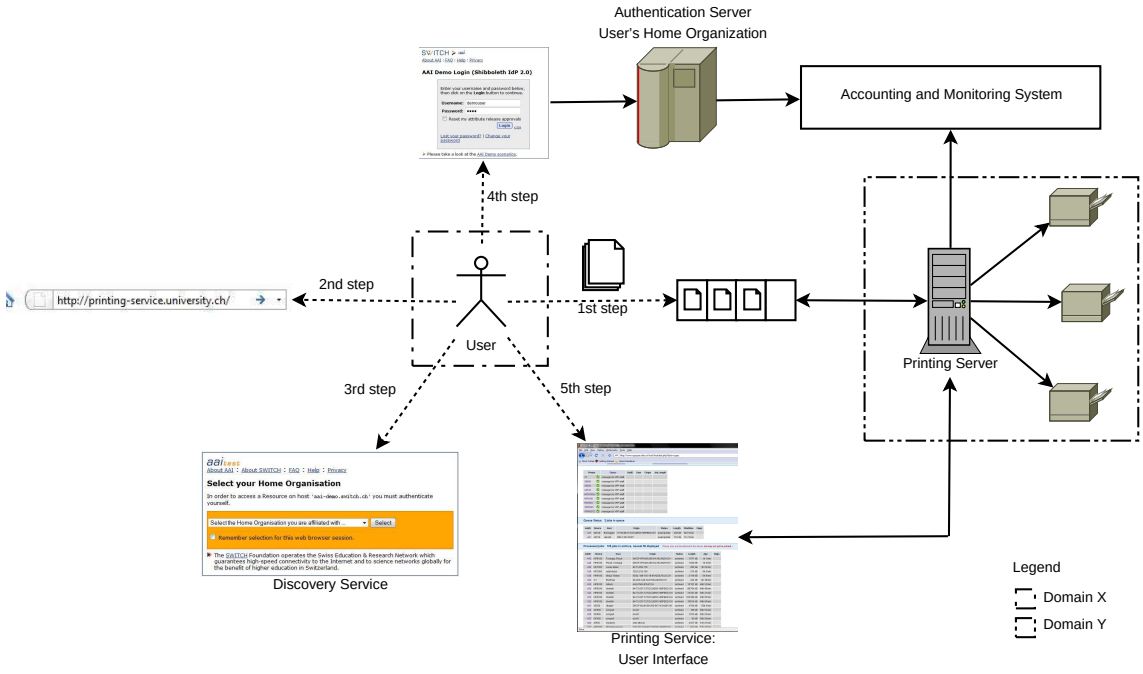

Figure 4: Printing scenario

In Figure 4, the user interacts with the system and performs the following steps:

1. The user send documents (PDF/PS files) to the printer server's queue using printing drivers (*e.g.*, lpr, CUPS, Windows/Mac drivers). The documents remain queued up until the user request. Note that the user is not yet authenticated;

2. In order to print the queued document, the user accesses the printing service via a web interface;
3. However, as the user is not yet authenticated, the discovery service asks ”where are you from“. Then, the user selects which organization he/she belongs to in order to complete the authentication process;
4. The user authenticates hat his IdP and, with that, completes the authentication phase. The Printing Service authorizes the service access based on the user's attributes;
5. Therefore, the resource becomes available and the user is able to access the printing service interface, which has the main function to visualize a list of queued printing jobs and, consequently, to request to print them. The printer server will perform such action depending on the available printing credit/quota for the given request;
6. Finally, the user is able to visualize the confirmation that the submitted job is authorized to be printed.

It is important to mention that step 3 is basically a manner of discovering which entity would be in charge of authenticating the user. In this case, the entity here would be a university or any organization that participates in the SWITCHaai. Another point to be observed is that some steps are transparent to the user, like when the discovery service sends a request to the user's web browser to be redirected to the identity provider for authentication. Such steps are further explained in the Shibboleth documentation [11] and are not explained within this document.

Another aspect to take into consideration is that the user may be at a different domain from where the Printer service is, but not necessarily. Figure 4 shows a user inside the domain X printing some documents at domain Y, *e.g.*, an ETH student printing some articles inside UZH library. The use of such resource should not be only accounted accordingly inside a unique domain, but also across domains.

3.2.3.2 System Perspective

In the system's point of view of the scenario, the main artifacts are the monitored data generated both by the identity provider and service provider. The identity provider should generate accounting data (*e.g.*, in form of logs) about who was authenticated and what are the attributes related to the user (name, affiliation, etc.). While in the service provider side, the accounting data are more directed – but not totally – for the printing job that the user requested, for example, the printing quality, the number of pages, and if it is colored. In other words, the attributes inherent to printing jobs. Applied to the accounting context, an aggregation process between the accounting data from the IdP and SP (*e.g.*, log files) should be done in order to exactly calculate resource usage. This process will make it possible to know, for example, that user *X* requested a printing job *Y* with *Z* printing characteristics/attributes at a *W* date. Therefore, the accounting and monitoring system should be directly integrated with the identity provider (Authentication Server of User's Home Organization) and with the Service Provider (Printing Service), in order to collect, process, and analyze relevant data for the printing scenario.

3.2.4 Attributes

The following potential accounting attributes were determined specific to the printing service:

- Number of pages: total number of pages of a document;
- Content-orientation: portrait, landscape;
- Media size: A4, letter, etc;
- Sides: on which paper side(s) printing should performed. Double-sided, single-sided, etc;
- File size: in Kbytes;
- Resolution: in dots per inch;
- Color: colored or monochrome printing;
- Surface: how many square meters are printed;
- Estimated-time to print: an estimate of the total time that a printer will take to perform a job. Mostly used for 3D plotters;
- Total time to print: the real (*i.e.* measured) total time used to perform a job. Mostly used for 3D plotters;
- Billing type: the purpose of a printing job. It can be private, university-related, etc.

This list is a preliminary set of attributes, meaning that it can be enhanced depending on the necessity of involved parties.

3.3 Service-dependent Scenario 3: SMS Service

This scenario describes the accounting for SMS service. Accounting data will provide details about the usage of the service and can be used to charge or limit the service usage of individual users.

3.3.1 Description of the Current State

As it is the case for the printing scenario, an SMS service is also currently implemented and operated by ETH. It is currently a free service for students and employees. The only restriction is the number of SMS per day. In this case, the implementation of an appropriate accounting and monitoring infrastructure can bring numerous benefits like cost savings and a better understanding of user needs. Moreover, proper accounting could help to refine a Service Level Agreement with telecommunications companies, as for example, making it clear what is the time period in which most SMS are sent.

3.3.2 Goals and Pre-conditions

Goals (user perspective):

- Any authorized user in the federation is able to send SMS to any mobile number.
- Users are able to see their SMS usage and limits.

Goals (system perspective):

- Based on rules which are applied to SMS and user attributes, resource access is either restricted or granted;
- Provide usage statistics about the SMS service.
- Keep a record of all events relevant to this resource (*e.g.*, number of sent SMS), in order to make the accounting a viable and consistent process;
- Send all monitored data to a data repository in order to concentrate all information for the accounting server (accounting process).

Pre-Conditions:

- Have an account at one of the IdPs in the federation;
- Provide a valid mobile number.

3.3.3 Accounting and Monitoring Extension in the Scenario

The description of the scenario is divided in two parts. The first part presents a user's point of view of the SMS scenario, where interaction between a user and the system is depicted, while the second part is focused on presenting the system's point of view.

3.3.3.1 User Perspective

In Figure 5, a user interacts with the system and performs the following steps:

1. A user tries to access a resource by web, in this case, the SMS service;
2. However, as the user is not yet authenticated, the discover service asks "where are you from". Then, the user selects which organization he/she belongs to in order to complete the authentication process;
3. The user enters the login information (username & password) and completes the authentication phase. The SMS service authorizes the service access based on the user's attributes;

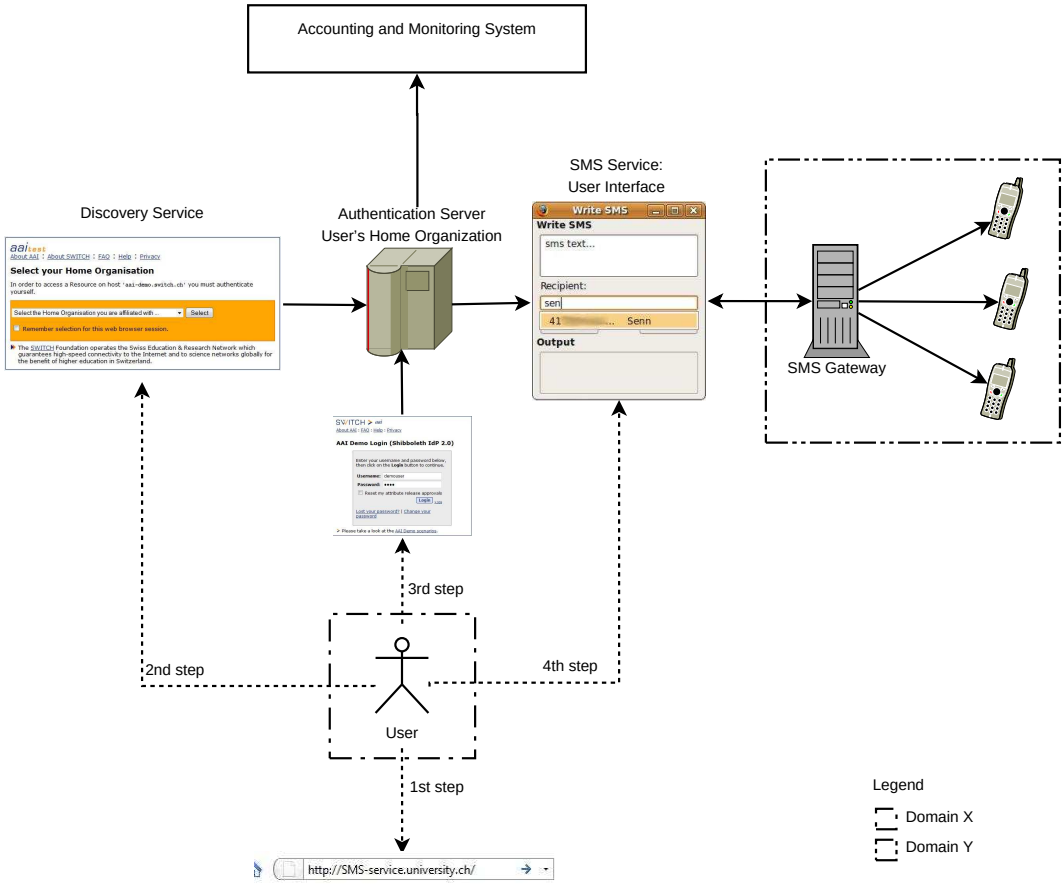

Figure 5: SMS scenario

4. The resource becomes available and the user is able to access the SMS service interface;

5. By accessing the SMS service interface, the user can write an SMS by providing a mobile number and a short message (limited by a certain amount of characters);

6. In the end, the user receives a confirmation that the SMS was successfully sent.

Another aspect to take into consideration is that the user may be at a different domain from where the SMS service is, but not necessarily. Figure 4 shows a user inside the domain X sending some SMS at domain Y, *e.g.*, an ETH student sending an SMS through UZH SMS gateway. The use of such resource should not be only accounted inside a unique domain, but also across domains.

3.3.3.2 System Perspective

In the system's point of view of the scenario, and applied to an accounting context, collected data from the IdP and SP (*e.g.*, log files) should be properly manipulated (*e.g.*, parsing log files) in order to exactly calculate resource usage. Such a process will make

it possible to know, for example, that user *X* sent an SMS *Y* with *Z* message characteristics/attributes at a certain *W* date period. The accounting and monitoring system should implement this process by collecting, processing, and analyzing relevant data related to the SMS service.

3.3.4 Attributes

For the SMS scenario, the following potential accounting attributes were determined specific to the SMS service:

- Number of characters: The number of characters the message is composed of.
- Characters encoding: The character encoding used to write the message.
- Number of SMS: The number of SMSs sent at the same time. Note that a message submitted by a user might require to send several SMSs due to the limited number of characters allowed in a SMS.

This list is a preliminary set of attributes, meaning that it can be enhanced depending on the necessity of involved parties.

3.4 Service-dependent Scenario 4: SWITCHpoint / Collaboration

This scenario includes the accounting for an audio and video conferencing system. Similarly to the scenarios above, accounting data will be used to get detailed usage statistics and to monitor the system.

3.4.1 Description of the Current State

SWITCHpoint/Collaboration – also known as Adobe Connect (AC) – is a service that provides audio and video conferencing and online collaboration services. It allows authorized users to set up audio and video conference sessions and to share the user's desktop, slides, and webcam image. The service is enabled for subscribed organizations. These organizations are charged per number of students and staff members.

A usage statistics of the service is helpful for analyzing the acceptance, usefulness, and indirectly, the necessity of this service. It is also helpful for deciding whether the service is still worthwhile to have it in the portfolio of both the respective organizations and SWITCH.

3.4.2 Goals and Pre-conditions

Goals (user perspective):

- Any authorized user is able to setup audio and video conferences. SWITCHpoint provides users with real-time communication over distance, either in very limited video-quality but enriched with collaboration facilities or in excellent video-quality with very limited collaboration facilities.
- Users might get an overview on their service usage statistics.

Goals (system perspective):

- Statistics give a measure for necessity and usefulness of the service within a university or on a community perspective. They are also used to get information on growth of demand and expected expansion of service. Statistics on number of meeting-rooms per organization, average frequency of usage per room and average or peak members per meeting are considered to be helpful values for decision makers to estimate the value of this service for their organization or for SWITCH to decide on the usefulness and need for extending the service in whole.
- The statistics are aggregated in reasonable manner. Exact figures are presented in tabular form, for trend observation a graphical representation would be preferred.
- The system keeps a record of all events relevant to this resource, in order to make the accounting a viable and consistent process;
- Send all monitored data to a data repository in order to concentrate all information for the accounting server (accounting process).

Pre-Conditions:

- For users of the service: to have an account at one of the IdPs in the federation and that the IdP is subscriber of the service.
- For organization administrators: to have access to the accounting data/server using an AAI account.

3.4.3 Accounting and Monitoring Extension in the Scenario

Currently, the platform statistics exist only rudimentarily, and there is no interface for organizational administrators to see any metrics on usage. Statistical data is gathered partly within the application itself, partly done via frontend.

In Figure 6, Users A and B interact with the system in the following steps:

1. User A of organization A logs in (with AAI, in the same way as described in other examples) and opens a meeting in Adobe Connect.

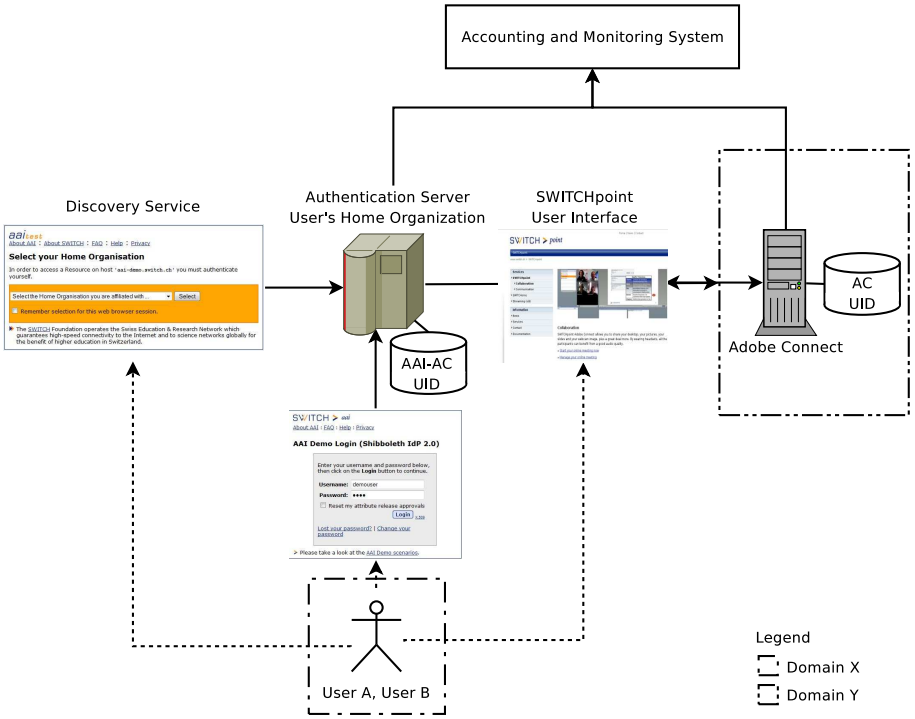

Figure 6: SWITCHpoint scenario

2. User A invites User B.
3. User B of organization B logs in (with or without AAI) and joins the meeting of User A.
4. User A and User B interact within Adobe Connect.
5. After a random period of time, User A and B leave the meeting. Who has to leave first is not specified.

The Adobe Connect utilizes the user's initially used e-mail address as Unique Identifier (AC-UID). Besides that, it stores the following information (list not comprehensive):

- Which meeting room belongs to (*i.e.* has been created by) which user.
- Which content is stored for which user (*i.e.* meeting room).

The AAI-Frontend performs the authentication and authorization and uses the AAI-UID as its Unique Identifier. Besides that, it stores the following information (list not comprehensive):

- Which AAI-UID is mapped to which AC-UID.
- Which user belongs to which organization (given by AAI).
- When a user logged in (was authenticated and authorized).

3.4.4 Attributes

To gather statistics of certain value, data is needed from both system components (AAI-AC and AC database). The system shall provide information about the number of meetings per organization, number of users per organization, number of logins per organization. The data shall be subsumed on a monthly basis. Therefore, the following potential accounting attributes have been determined specific to the SWITCHpoint service:

- Number of participants: The number of participants reserved for a conference.
- Date and time of the conference: The reserved date and time for the conference.
- Length of the conference: The reserved duration of the conference.
- Type of conference: Audio or video conference.

This list is a preliminary set of attributes, meaning that it can be enhanced depending on the necessity of involved parties.

3.5 Service-dependent Scenario 5: SWITCHcast and nanoo.tv

This scenario describes the accounting for a streaming service. The accounting functionality in the scenario is used to get usage statistics of the the service, which might be also used for cost distribution between subscribed institutions.

3.5.1 Description of the Current State

SWITCHcast contains videos generated by organizations within the SWITCH community. Besides these videos, it will also contain recordings of TV broadcasts for educational purposes. These recordings are done in nanoo.tv, which is the name of a software of the Zürcher Hochschule der Künste (ZHdK). nanoo.tv is also the name of a project that aims at making available the recorded broadcasts for the Swiss higher education community through SWITCHcast. This raises copyright and compensation payment issues: for every view of the recorded TV broadcast content, royalty fees will be due. Therefore, precise usages statistics will be needed.

3.5.2 Goals and Pre-conditions

Goals (user perspective):

- Any authorized user is able to watch streaming videos.
- Users can get an overview on their service usage statistics. Users see how many videos they watched and what that means regarding royalty fees.

Goals (system perspective):

- A general overview of the usage is obtained, which indicates popularity of the service.
- A general overview of royalty fees on whole or per organization is obtained.
- Statistics on number of consumptions per movie (how many consumptions per organization and in whole) and of number of watched minutes per movie (again per organization and in whole) are desired.
- The system keeps a record of all events relevant to this resource, in order to make the accounting a viable and consistent process;
- Send all monitored data to a data repository in order to concentrate all information for the accounting server (accounting process).

Pre-Conditions:

- AAI-account and adequate access rights.

3.5.3 Accounting and Monitoring Extension in the Scenario

Consumption of copyrighted and royalty-fee payable content will in a first stage be measured by clicks, later by number of minutes. The usage data will be gathered and summed up for each organization (and sub-organization, if possible) and for the SWITCHcast platform as a whole.

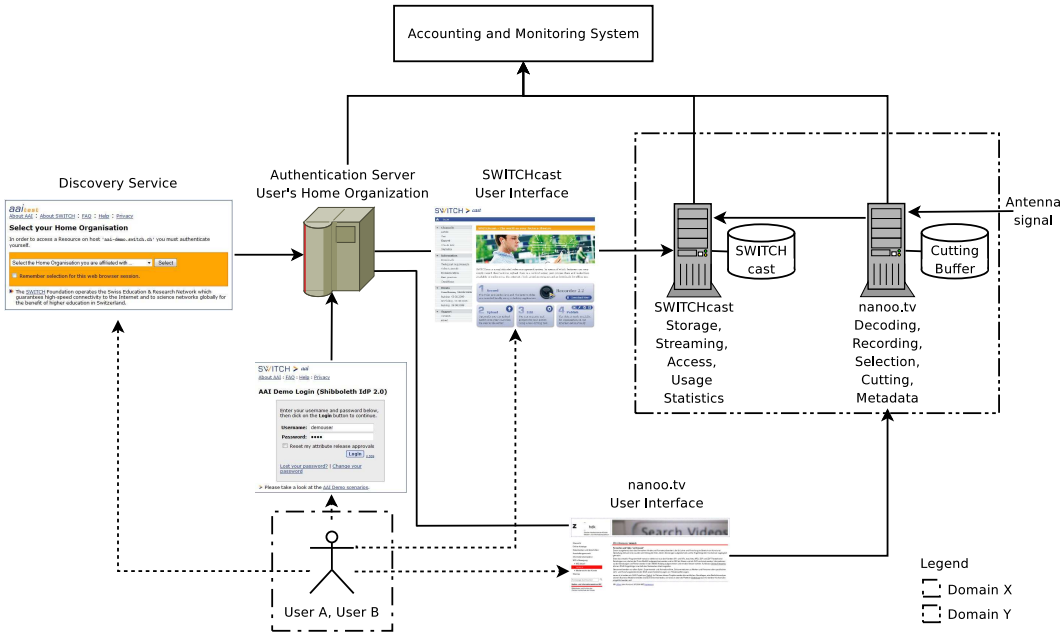

Figure 7: SWITCHcast scenario

A TV channel is recorded to the nanoo.tv cutting buffer according to the selection made by the operator (User A) (*cf*. figure 7). He then cuts away commercials and adds meta data for the content (not part of the figure). The recording is then transferred to the SWITCHcast platform where it is permanently stored and prepared for streaming. When User B views the video stream, the number of minutes consumed will be recorded. Both the recording and the consumption may be subject to royalty fees. The respective information is transferred to the Accounting and Monitoring System.

3.5.4 Attributes

To gather statistics of certain value, data is needed from both system components (nanoo.tv and SWITCHcast servers). The system shall provide information about usage statistics both for recorded and consumed videos, like number of users per organization, number of videos recorded, number of requests for each movie per user per organization, number of watched minutes for each movie per user per organization, and aggregations thereof per month, per movie and per organization. Therefore, the following potential accounting attributes have been determined specific to the SWITCHcast service:

- Video ID: The identifier of the video.
- Streaming time: The time defining how long the user watched the video.
- Video resolution: The resolution of the video, *e.g.*, high-definition (HD) and 1024x768.
- Video length: The length of the video in minutes.

This list is a preliminary set of attributes, meaning that it can be enhanced depending on the necessity of involved parties.

4 Accounting Attributes

Accounting records contain accounting attributes that define the characteristics of an event or a service. In the following, accounting attributes are summarized that have been identified for the accounting and monitoring of the AAI and for the services specified in the scenarios above. Some of the attributes define general characteristics used for the AAI and used for any service (*e.g.*, timestamp), while some of the attributes define specific characteristics of a certain service (*e.g.*, number of pages printed). They are listed separately.

The accounting attributes listed below determine a basic set of accounting attributes. The list of attributes might be extended based on future needs. Additionally, not all attributes listed need to be present in all accounting records. Depending on the configuration of the accounting system and the policies of the operator, the accounting for the AAI and certain service might not include all attributes below.

Accounting attributes for the AAI and generic accounting attributes for services:

- Event-Timestamp: The date and time of an event.
- Accounting-Session-ID: The unique identifier of the accounting session. The accounting session binds together accounting records belonging to a certain service session.
- Accounting-Record-Number: The sequence number of an accounting record. The sequence number together with the accounting session ID is unique.
- Accounting-Record-Type: The type of the accounting record. An accounting record can be an event, a start, an interim, or a stop record. An event record contains information about a one-time event (meaning that the start and end of the event are simultaneous). The start, interim, and stop records are used for services with measurable length. The start record is sent at the start of a session and contains accounting information related to the start of the service. The interim record contains cumulative accounting information for an existing session. Several interim records can be sent during a session. The stop record is sent at the end of a session. Start, interim, and stop sessions might be sent for example in case of a video streaming service.
- Accounting-Client-ID: The identifier of the accounting client, the node that has sent the accounting record.
- IdP-Entity-ID: The unique identifier of the Identity Provider (IdP).
- SP-Entity-ID: The unique identifier of the Service Provider (SP).
- SAML-Profile: The SAML profile used for a transaction.
- SAML-Binding: The SAML binding used for a transaction.
- SAML-Attributes: The names of the attributes released from an IdP.

- User-Unique-ID: A persistent identifier of the user.[4]
- User-Name: The human-readable name of the user.
- User-IP-Address: The IP address from which the user request has come.
- User-Home-Organization: The name of the home organization of the user.
- Authentication-Method: The authentication method used to authenticate the user.
- Authentication-Result: The result of the authentication (success, failed).
- Authorization-Result: The result of the authorization (granted, denied).
- Cause: The cause of the authentication or authorization failure.
- Shib-Application-ID: The Shibboleth application identifier at the SP side.
- Shib-Session-Id: The Shibboleth session identifier at the IdP or SP side.
- SAML-Assertion-ID: The SAML assertion identifier(s), a list of identifiers if there are more than one.
- Service-Name: The name of the service, the accounting record contains information about (*e.g.*, authentication, authorization, printing, SMS).
- Service-Duration: The duration of the service consumption.
- Termination-Cause: The cause of the service termination.

Accounting attributes specific for the printing service:

- Printing-Pages: The total number of printed pages.
- Printing-Page-Orientation: The orientation of the page (portrait, landscape).
- Printing-Media-Size: The size of the media (A4, letter).
- Printing-Double-Sided: Single-sided or double-sided printing.
- Printing-File-Size: The file size of the printed document in Kbytes.
- Printing-Resolution: The printing resolution in dots per inch.
- Printing-Color: Black-and-white or color printing.
- Printing-Surface: The surface of the printed media in square meters.
- Printing-Estimated-Time: The estimated total time that the printer will take to perform the job. Mostly used for 3D plotters.
- Printing-Total-Time: The total time used to perform the job. Mostly used for 3D plotters.

[4] A user may have different identifiers.

- Printing-Billing-Type: The billing type of printing, *e.g.*, private, university-related.

Accounting attributes specific for the SMS service:

- SMS-Characters: The number of characters the SMS contains.
- SMS-Encoding: The character encoding used to write the SMS.
- SMS-Amount: The total number of SMS.

Accounting attributes specific for the SWITCHpoint service (or in general for video conferencing service):

- Conference-Number-of-Participants: The number of participants in the conference.
- Conference-Date: The date and time of the conference.
- Conference-Length: The duration of the conference.
- Conference-Type: Type of the conference, *i.e.* audio or video conference.

Accounting attributes specific for the SWITCHcast service (or in general for streaming service):

- Streaming-Video-ID: The identifier of the video.
- Streaming-Duration: The time defining how long the user watched the video.
- Streaming-Video-Resolution: The resolution of the video, *e.g.*, high-definition (HD) and 1024x768.
- Streaming-Video-Length: The length of the video in minutes.

5 Requirements for the Accounting and Monitoring Architecture

This section defines the functional and non-functional requirements of the accounting and monitoring architecture of the AAI.

5.1 Functional Requirements

The following functional requirements have been identified for the accounting and monitoring architecture of the AAI:

F-R.1 For Shibboleth-specific accounting, the system shall be able to gather information about IdP, SP, and DS (WAYF) usage (accounting data) from Shibboleth. This includes accounting for authentication events at the IdP and accounting for authorization events at the SP. Different sources may deliver data about the same event, therefore unintentional multiple counting of an event should be prevented.

F-R.2 For service specific accounting, the system shall support accounting for service usage at the SP.

F-R.3 The accounting granularity shall be configurable, meaning that the administrator shall be able to configure which accounting attributes to collect for each service and send to the server at which time interval.

F-R.4 The system shall enable to collect and store accounting data both on the federation member level (*i.e.* each member stores his own accounting data) and on the federation level (*i.e.* SWITCH can receive accounting data from federation members). This shall enable IdPs and SPs to monitor the AAI infrastructure and to generate usage statistics about the AAI and services on top of it.

F-R.5 The system shall support accounting record exchange between federation members.

F-R.6 The system shall support the aggregation and correlation of accounting data from an IdP and SP for a single session.

F-R.7 The system shall support accounting policies to define which domain can receive which data in which level of details in case when accounting data is exchanged between domains. For example, domains should have agreements (with some policies) saying that some attributes should be hidden when accounting data from that specific domain's users is exchanged.

F-R.8 The system shall support data anonymization (*e.g.*, remove user ID or substitute it with its hash value if required) for data storage and transfer.

F-R.9 The system shall support different time frames for accounting data transfer both within a domain and between domains (*e.g.*, real-time and batch transfer of accounting data).

F-R.10 The system shall support access control to accounted data. Access control in the sense that just authorized operators should have access to accounting records persisted in a database, for example.

F-R.11 The system shall support the long-term storage and archiving of accounting data (technical and regulation aspects). In long term storage, data should be aggregated to eliminate data protection issues.

F-R.12 The system shall support interim accounting records to periodically update the status of a running session.

F-R.13 The system shall support mandatory and optional accounting attributes.

F-R.14 The systems shall support the definition of new accounting attributes. An initial set of accounting attributes is specified in Section 4.

F-R.15 The required accounting metadata shall be integrated into the Shibboleth metadata.

F-R.16 The system shall make use of the SWITCHaai federation metadata if necessary.

5.2 Non-Functional Requirements

The following non-functional requirements have been identified for the accounting and monitoring architecture of the AAI:

NF-R.1 Reliable transfer of accounting data: the communication and transfer of accounting data should be done without disruption, always reflecting what was originally accounted in the source.

NF-R.2 Secure transfer of accounting data: the communication should be safe, possibly using encryption methods.

NF-R.3 Availability of the accounting system: the availability of servers (and other components as well) should be satisfactory to avoid service and accounting disruption.

NF-R.4 Fault tolerance: in case of service and/or accounting disruption, back-out plans should be invoked to handle such unexpected events. For example, employing a fail-over schema on servers.

NF-R.5 Performance (events/s, records/s): produced data should be processed accordingly in a satisfactory speed rate. To measure what is a satisfactory speed rate, it should be taken into consideration what kind of service the environment supports. For example, if the environment supports a printing service, the processing data performance might be considerably high since it require real-time results.

NF-R.6 Flexible and extensible for new services: aggregation of additional or totally new components/services, and support for defining new accounting attributes which depends on new services.

NF-R.7 Support Shibboleth from v2.0.

6 High-level Architecture

This section describes the high-level accounting and monitoring architecture. The architecture design includes the components and their interfaces. As an example instantiation of the architecture, the architecture is separately shown for the printing and SMS scenarios at the end of this section.

6.1 Components

Shibboleth works in a distributed environment, where multiple services are provided in a single domain as well as in different domains. Shibboleth itself has multiple components as well, like the IdP, SP, and DS. Therefore, the accounting architecture has to collect accounting data from different components (physical machines) at different locations. Additionally, different services might be written in different programming languages and provide different interfaces to access information relevant for accounting. Standard Shibboleth components also expose different interfaces to get accounting-relevant information. Therefore, the high-level architecture of the accounting and monitoring system, as shown in Figure 8, foresees component-specific interfaces (i-idp, i-sp, i-serv, i-ds) in order to interact with Shibboleth components and services. The Accounting Server is a central component of the architecture and is responsible for collecting accounting records and store them in the Accounting Database. In order to enable the integration of all Shibboleth components and AAI-based services (existing and future ones), the architecture defines a common interface towards the Accounting Server (i-acct). Thus, each AAI component and service uses the same accounting interface to the server. The Accounting Client, which is colocated with the Shibboleth components and services, uses this interface to communicate with the Accounting Server. Additionally, this interface is also used between Accounting Servers, where the accounting record exchange may happen within a single domain as well as between federation members.

The components of the high-level architecture are explained based on Figure 8. In the figure, the IdP (web appl.) represents the Shibboleth Identity Provider web application, the SP (shibd) represents the shibd daemon on the Service Provider side, the DS (web appl.) represents the Discovery Service web application, and finally the Service represents any service that is accessible via the AAI infrastructure, *e.g.*, the printing service, the SMS gateway.

The accounting and monitoring architecture defines the following components:

- The Meters (IdP, SP, DS, Service) are responsible for gathering information about events and service consumption for the IdP, SP, DS, and the Service. In the case of the IdP and SP the Meter can be for example the built-in logger of Shibboleth. But the Meter does not necessary have to be a logger. The Meter is integrated into the IdP, SP, DS, and Service. For a description of Shibboleth logging capabilities, see Section 7.
- The Collectors (IdP, SP, DS, Service) receive data from the Meters and make them available for the Accounting Client. They are specific to the IdP, SP, DS, and a certain

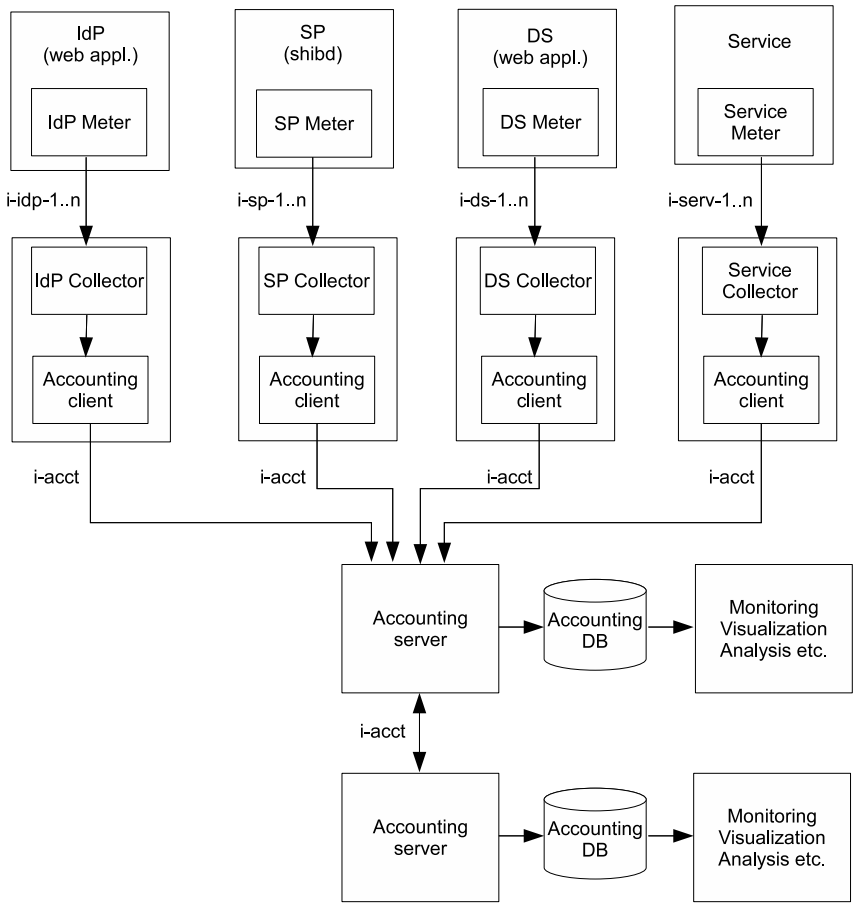

Figure 8: Architecture

Service. There can be different implementations of the Collectors depending on the Meters. A Collector can be for example a log file parser if the Meter is a logger. The Collector will be separated from the IdP, SP, DS, and Service, but it could be also integrated with them (depending on the solution).

- The Accounting Client is responsible to create accounting records and send them to the Accounting Server. The Accounting Client is the same for the IdP, SP, DS, and Service. It provides a common interface towards the Accounting Server.
- The Accounting Server receives accounting records and stores them in a local database. An Accounting Server can send records to another Accounting Server.
- Monitoring/Visualization/Analysis applications retrieve accounting data from the database and make them available for the operator.

6.2 Interfaces

The components of the accounting and monitoring architecture uses the following interfaces, as shown in Figure 8:

- i-idp-1..n: The interfaces between IdP Meter and IdP Collector. There can be different interfaces between Meter and Collector, *e.g.*, Shibboleth log files.
- i-sp-1..n: The interfaces between SP Meter and SP Collector. There can be different interfaces between Meter and Collector, *e.g.*, Shibboleth log files.
- i-ds-1..n: The interfaces between DS Meter and DS Collector. There can be different interfaces between Meter and Collector.
- i-serv-1..n: The interfaces between Service Meter and Service Collector. There can be different interfaces between Meter and Collector depending on the service, *e.g.*, printer log files, RPC.
- i-acct: The interface between Accounting Client and Server and between Accounting Servers. This interface is common for the IdP, SP, DS, and any Service.

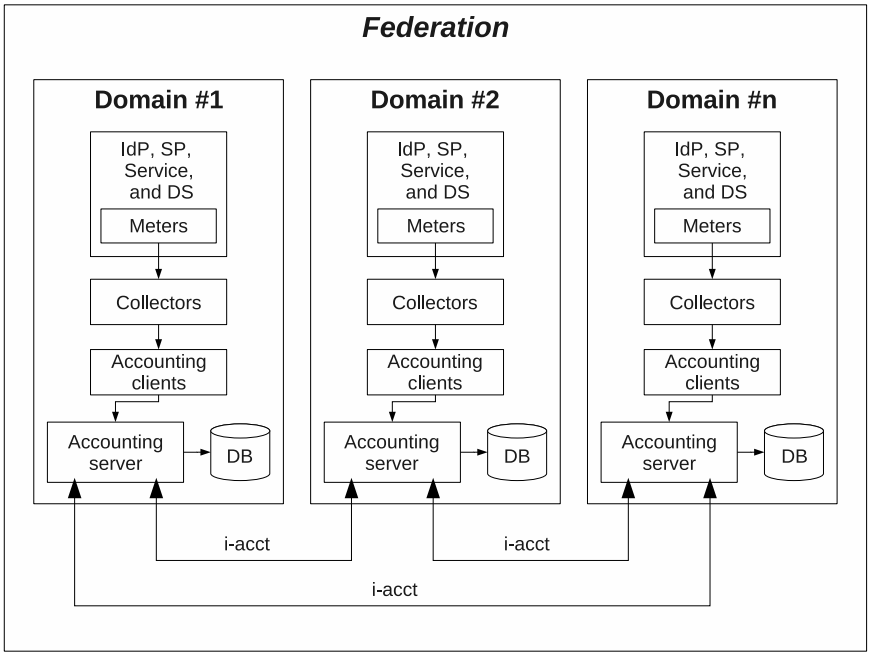

Figure 9: Architecture in a top view

Figure 9 shows the architecture of the accounting system in a top-level view, considering accounting interfaces across federation domains. Regarding organizational aspects the followings apply:

- Each domain within the federation has its own Accounting Server (one or more servers) as well as its own database to store accounting records. Additionally, there might be a central Accounting Server (with also a central database) which accounts for the entire AAI federation.
- The communication between Accounting Servers across domains should be standardized (i-acct interface) and properly secured.

- The exchange of accounting records across domains will be made on demand. It means that a given Accounting Server identifies the necessity to transmit accounting records when any information may influence the accounting process at another domain. A typical example: user *X* from *ETH* used the printing service at *UZH* domain. Then, it is necessary that UZH informs ETH about its use, since the usage must accounted accordingly and billed properly.
- Databases on different domains should be originated from the same model in order to be self-compatible when information is exchanged and/or persisted.
- The architecture supports the idea that accounting information is sensitive and private for a domain. As it was mentioned before, just accounting records that are essential to be transmitted should be exchanged across domains, or even, across federations.
- The architecture should be designed for the case where accounting data leaves the federation and aggregation of multiple federation's data would be done.

6.3 Architecture Components Applied within Scenarios Context

Figure 10 and 11 shows how the architecture components (presented in Section 6.1) are applied in the printing service scenario (scenario 2) and the SMS service scenario (scenario 3).

It is important to note that, on both figures, it is exemplified how a certain user interacts with the system based on steps presented in Section 3. Also, users can be theoretically located anywhere to access a given resource, as in the case of the SMS service. However, some services (as the printing service) may require that users must be located inside the domain/network where the printer server is. As shown in Figure 10, the user location is implicit.

Both figures intend to primarily focus on two things: (a) how architecture components and their interfaces are applied in both scenarios, and (b) where architecture components are located if a given user from Domain X is using a resource at Domain Y. Summarizing what is presented on both figures, Accounting Clients communicate with Accounting Servers inside a domain. Accounting Servers exchange information regarding resource usage only when necessary. Accounting Servers can also exchange accounting records with other Accounting Servers, however both parts should beforehand agree on it. A use case would be that data from different federations are aggregated on a meta-federation level. In that case, the owner and holder of the information has the responsibility to respect privacy and data protection laws when he passes on data.

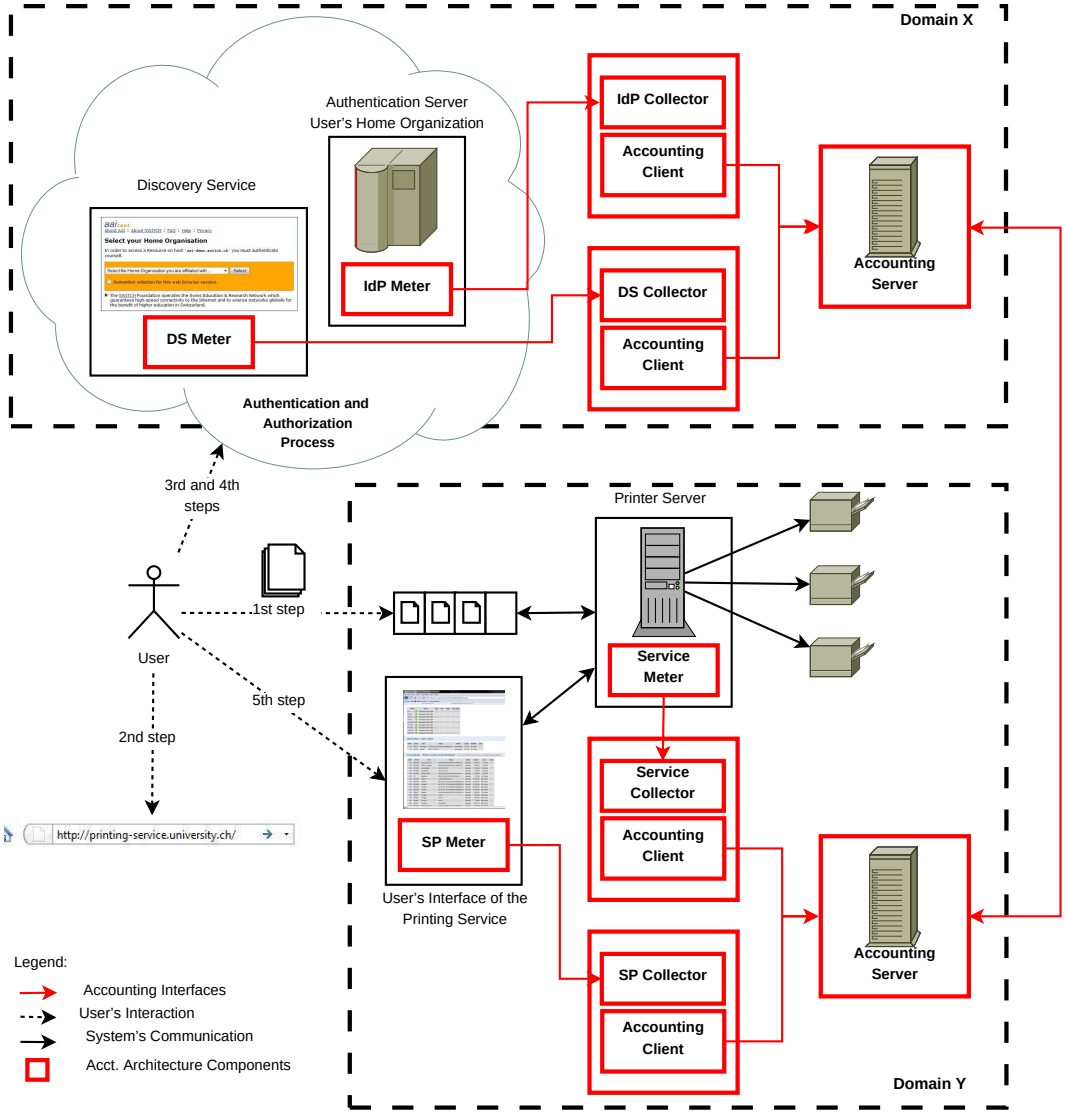

Figure 10: Printing scenario and the relation with architecture's components

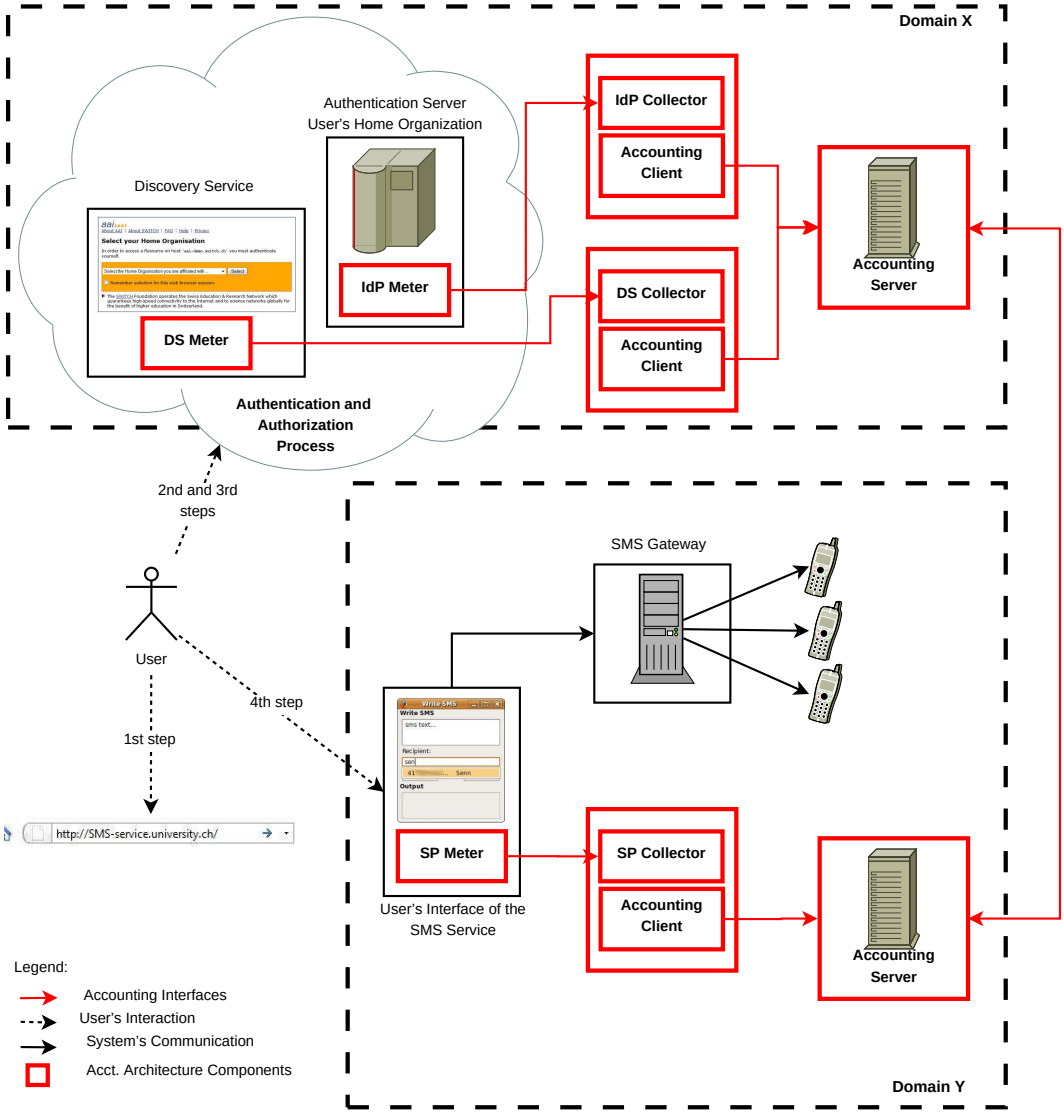

Figure 11: SMS scenario and the relation with architecture's components

7 Shibboleth Logging Capabilities

Part of the data for the monitoring and accounting tasks can be gathered from the Shibboleth log files. Those log files determine the data set used for all further monitoring and accounting tasks. Therefore, this section covers the logging mechanisms, log files and preferences to get an overview of the default Shibboleth logging capabilities.

7.1 Shibboleth IdP Logging Capabilities

This section covers the basic and default logging capabilities of the Shibboleth 2 IdP. The IdP uses the so called Logback logging system which is the successor of the well known log4j logging framework. The Logback system defines five logging levels (TRACE, DEBUG, INFO, WARN, ERROR) from the highest to the lowest log level. Each of those log levels log also all messages of the log level below it. Besides the information and comments in the logging configuration file, the information in this section is based on the Shibboleth 2 Documentation about the IdP Logging [12].

7.1.1 IdP Logging Configuration

The IdP logging configuration file is located in $IDP_HOME/conf/logging.xml. Inside the configuration file, different logging facilities and their log levels are defined. Listing 1 shows two such example entries.

```
<logger name="edu.internet2.middleware.shibboleth">
    <level value="INFO" />
</logger>

<!-- Logs OpenSAML, but not IdP, messages -->
<logger name="org.opensaml">
    <level value="WARN" />
</logger>
```

Listing 1: Example entries of logging.xml

There are also some other useful loggers defined which can provide valuable information for specific logging requirements. The most common loggers are listed in Table 6.

7.1.2 IdP Default Log Files

By using the default configuration, the Shibboleth 2 IdP writes three log files into its logging directory $IDP_HOME/logs/. The following sub-sections take a closer look on those files with reference to their logging format and their contents according to the Shibboleth 2 Documentation Wiki [11].

Table 6: Useful IdP 2.1 loggers

Category	Description
Shibboleth-Access	The logger to which Shibboleth access messages are written (similar to an Apache access.log)
Shibboleth-Audit	The logger to which Shibboleth audit messages are written
PROTOCOL_MESSAGE	The logger to which incoming and outgoing XML protocol messages are logged
org.opensaml	Messages related only to receiving, parsing, evaluating security of, producing and encoding SAML messages
edu.internet2.middleware.shibboleth	Messages related to all the non-SAML message parsing/encoding work; profile handling, authentication, attribute resolution and filtering
edu.internet2.middleware.shibboleth.idp.authn	IdP messages related only to authentication
edu.internet2.middleware.shibboleth.common.relyingparty	IdP messages related to relying party configurations in use
edu.internet2.middleware.shibboleth.common.attribute	IdP messages related only to attribute resolution and filtering
edu.vt.middleware.ldap	Messages related to LDAP actions

7.1.2.1 idp-access.log

The idp-access.log file creates an entry each time the IdP is accessed, whether an authentication was successful or not. This file is in its basics similar to an Apache access.log log file. The logging entries include the request time in UTC, the remote host making the request, the IdP FQDN and port and the profile handler request path. The log file is in a machine parsable format defined in the structure showed in Listing 2 while Listing 3 shows some example entries.

```
requestTime|remoteHost|serverHost|serverPort|requestPath|
```

Listing 2: idp-access.log structure

```
20090820T183446Z|10.0.0.64|idp.example.org:443|/profile/SAML2/Redirect/SSO|
20090820T183450Z|10.0.0.64|idp.example.org:443|/profile/Status|
```

Listing 3: idp-access.log extract

7.1.2.2 idp-audit.log

The idp-audit.log file creates an entry for each successful authentication on the IdP for each time the IdP sends data to a relying party. In this context the relying party is usually

a SP and perhaps in some advanced rare cases also an IdP. These message entries contain the audit event time, the IdP and relying party ID, request and response bindings, communication profile ID, request and response ID, principal name, authentication method and released attributes of the current user. This log is also in a machine parsable format and has the structure shown in Listing 4.

```
auditEventTime|requestBinding|requestId|relyingPartyId|messageProfileId|
    assertingPartyId|responseBinding|responseId|principalName|authNMethod|
    releasedAttributeId1,releasedAttributeId2,|nameIdentifier|assertion1ID,
    assertion2ID,|
```

Listing 4: idp-audit.log structure

Listing 5 shows an extract of such a idp-audit.log file in an example AAI-infrastructure. It is notable that the nameIdentifier and assertionIDs were added since Shibboleth 2.1. The log time here is also in UTC.

```
20090820T193737Z|urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect|
    _e82c59de2348efd0bf65f37f24d78111|https://sp.example.org/shibboleth|
    urn:mace:shibboleth:2.0:profiles:saml2:sso|https://idp.example.org/idp/
    shibboleth|urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST|
    _7fc077c39e357af46232f0432f637942|alumni1|urn:oasis:names:tc:SAML:2.0
    :ac:classes:PasswordProtectedTransport|uid,eduPersonAffiliation,surname,
    givenName,swissEduPersonHomeOrganization,swissEduPersonUniqueID,
    swissEduPersonHomeOrganizationType,transientId,eduPersonTargetedID,email
    ,|||
20090820T193750Z|urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect|
    _e4db4cfebf21e03a8a87f09f95faebf4|https://altsp.example.org/shibboleth|
    urn:mace:shibboleth:2.0:profiles:saml2:sso|https://idp.example.org/idp/
    shibboleth|urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST|
    _529b234aede9cc4e2eea58f13bc78963|alumni1|urn:oasis:names:tc:SAML:2.0
    :ac:classes:PreviousSession|uid,eduPersonAffiliation,surname,givenName,
    swissEduPersonHomeOrganization,swissEduPersonUniqueID,
    swissEduPersonHomeOrganizationType,transientId,eduPersonTargetedID,email
    ,|||
```

Listing 5: idp-audit.log extract

A single log entry contains the following informations:

AuditEventTime: *20090820T193737Z*

This entry represents the UTC date and time (*YYYYMMDD*T*HHMMSS*Z) of the event with T and Z as delimiters.

requestBinding: *urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect*

This entry represents the used binding of the Shibboleth profile handler. This entry is a SAML binding, which is a description of how a SAML message is attached to an underlying transport protocol such as HTTP. That means if the message is sent over HTTP, which HTTP headers need to be set, which URL or form parameter names, etc. Those entries are different and depending on the used SAML version and Shibboleth version. They are defined in the $IDP_HOME/conf/handler.xml configuration file. Listing 6 shows an example XML definition of such a Profile Handler entry.

requestID: *_e82c59de2348efd0bf65f37f24d78111*
This entry represents a SAML requestID which is request specific. Besides debugging on SAML message level, there is probably no real use in it.

relyingPartyId: *https://sp.example.org/shibboleth*
This entry represents relying party which is asking for authentication.

messageProfileId: *urn:mace:shibboleth:2.0:profiles:saml2:sso*
This entry represents SAML message profileId used.

assertingPartyId: *https://idp.example.org/idp/shibboleth*
This entry represents the authentication asserting entity, which is usually the IdP itself and in some advanced cases it could be another relying IdP.

responseBinding: *urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST*
This entry represents the used response binding of the Shibboleth profile handler. It is the same like the requestBinding, just the other way around.

responseId: *_7fc077c39e357af46232f0432f637942*
This entry represents a SAML responseId which is request specific. Besides debugging on SAML message level, there is probably no real use in it.

principalName: *alumni1*
This entry represents the principalName, which is the login name a user supplied. It is the userField value supplied by the utilized LDAP directory defined in $IDP_HOME /conf/login.config. This value can vary, depending on the LDAP scheme and which LDAP attribute is assigned to the userField.

authNMethod: *urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport*
This entry represents the used login handler defined in $IDP_HOME/conf/handler.xml which is in most cases the Username/password login handler and if there is already a valid session it is the PreviousSession handler.

releasedAttributeId*x*: *uid,eduPersonAffiliation...*
Those entries represent the attributes which are released after attribute filtering to a specific SP.

nameIdentifier: This entry represents the defined name identifiers which are released to a specific SP after attribute filtering.

assertionId: Those entries represent just some specific assertion Id's which are sent to the SP after attribute filtering in the context of the Shibboleth SAML authentication.

```
<ProfileHandler xsi:type="SAML2SSO"
 inboundBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
 outboundBindingEnumeration="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
     SimpleSign
 urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
 urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact">
         <RequestPath>/SAML2/POST/SSO</RequestPath>
</ProfileHandler>
```

Listing 6: Example entry of a profile handler

7.1.2.3 idp-process.log

The idp-process.log file is the in depth log file of the IdP. All the data from the other log files is written into it too, but the entries are covered more in-depth. If the log levels are set to more verbose levels like DEBUG, all the data gets written into this file including the specific logger. So basically this log is meant to be human readable and contains messages that indicate what the IdP is currently doing, encountered errors, warning messages that may indicate potential problems, etc. If more verbose log levels are enabled, this file can quickly become huge, so it is recommended to run it in the default log levels if the system is in production and runs without problems.

7.2 Shibboleth SP Logging Capabilities

This section covers the basic and default logging capabilities of the Shibboleth 2 SP. The SP uses the log4shib framework, which is based on the log4cpp framework. That logging framework is very similar to the standard log4j framework and defines also five logging levels (TRACE, DEBUG, INFO, WARN, ERROR).

7.2.1 SP Logging Configuration

The configuration files for the SP logging are located usually in /etc/shibboleth/. There are three logger configuration files, namely native.logger, shibd.logger and syslog.logger which are for different processes and logging entities.

native.logger defines the InProcess logger of the Shibboleth daemon which saves the log messages concerning the Shibboleth Apache module. In the default logging mode there is just some session specific information about the SessionCache and the RequestMapper, but only status information. There are also WARN messages about the Service Provider denying access.

shibd.logger defines the OutOfProcess logger of the Shibboleth daemon which saves the log messages concerning the Shibboleth daemon itself. The information in this file could be used for debugging purposes and shows in the standard log levels mostly status information. Besides the basic log appender, another log appender is defined which writes a transaction log file containing status information about the new sessions.

syslog.logger is basically the same logging facility as defined in shibd.logger, but without the transaction log. This logger can write the log into the syslog of a Unix system for example using remote syslog servers.

Since the syslog appender is basically already contained in the shibd log appender, the focus lies on the log files written by native and shibd loggers.

7.2.2 SP Default Log Files

Using the default SP logging configuration, the Shibboleth 2 SP daemon writes the log files into its logging directory /var/log/shibboleth/. The following sub-sections take a closer look on those files with reference to their logging format and their contents.

7.2.2.1 native.log

As mentioned, in the default log levels of the InProcess logger, there is only a bit of status information available. By increasing the log level to more verbose levels, information about different mappings and session can be shown, also information about successful authorization for a specific resource or denied access. So to get useful information, the log level should be increased to get entries like the extract shown in Listing 7.

```
2009-08-20 15:13:29 DEBUG Shibboleth.SessionCache [3076] shib_auth_checker:
    searching local cache for session (_155a970fa3e7f856019ba2638fb0638fb03c05)
2009-08-20 15:13:29 DEBUG Shibboleth.SessionCache [3076] shib_auth_checker:
    session found locally, validating it for use
2009-08-20 15:13:29 DEBUG Shibboleth.SessionCache [3076] shib_auth_checker:
    comparing client address 10.0.0.192 against 10.0.0.192
2009-08-20 15:13:29 DEBUG Shibboleth.Apache [3076] shib_auth_checker:
    htaccess: a rule was successful, granting access
2009-08-20 15:13:29 DEBUG Shibboleth.ServiceProvider [3076] shib_auth_checker:
     access control provider granted access
2009-08-20 15:13:52 DEBUG Shibboleth.Apache [3077] shib_handler: mapped https:
    //sp.example.org/Shibboleth.sso/Session to default
```

Listing 7: native.log extract

Usually the entries are in the system's local time.

7.2.2.2 shibd.log

The OutOfProcess logger shows a lot of status information about the Shibboleth SP daemon. By increasing the log level more inside information about the SAML messages and other loggers defined in the configuration file can be pulled. But mainly this file contains debugging information and is also a bit of a collection of all the data the SP received from its relying party.

7.2.2.3 transaction.log

This log file lists all session created on the SP side, meaning every session that was initiated by a successful authentication on the relying party. The sessionID, the applicationId, the principal and the used attributes are the most important entries in this file. Listing 8 shows an entry of a new session on the SP.

```
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]: New session (ID:
    _2e22be29c5c49850387c2005c5e697b5) with (applicationId: default) for
    principal from (IdP: https://idp.example.org/idp/shibboleth) at (
    ClientAddress: 10.0.0.192) with (NameIdentifier:
    _e54e6db29f01b582a8c9e1fadcf575d1) using (Protocol:
    urn:oasis:names:tc:SAML:2.0:protocol) from (AssertionID:
    _bcc0f1ff6e3e24d2c7cd2cde76080db0)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]: Cached the following
    attributes with session (ID: _2e22be29c5c49850387c2005c5e697b5) for (
    applicationId: default) {
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-Person-uid (1
    values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-EP-Affiliation (1
     values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-Person-surname (1
     values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-InetOrgPerson-
    givenName (1 values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-SwissEP-
    HomeOrganization (1 values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-EP-Entitlement (2
     values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-SwissEP-
    HomeOrganizationType (1 values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     persistent-id (1
    values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]:     Shib-InetOrgPerson-
    mail (1 values)
2009-08-20 22:59:03 INFO Shibboleth-TRANSACTION [5]: }
```

Listing 8: transaction.log extract

There is no information about the attribute values, but the important values are the ID of the new session and the specific applicationId. The application is defined in the shibboleth2.xml configuration file. The time stamp marks are usually in the systems local time.

7.3 Shibboleth Logging Analysis Conclusions

The information which can be gathered through the log files in default mode can be assigned to the appropriate entities. The log files on the IdP side give information about the authentication, *i.e.* successful logins. By increasing log levels, information about failed authentications could be retrieved. The problem is that increasing the log levels can quickly grow into huge logging files. The information about authentication is the main content of the IdP log files.

On the other side, the SP log files can give a lot of information about newly created sessions, their attributes and the validating relying party. Basic information for usage statistics can be gathered even in the default log levels. If more information is needed about the session like the attribute values, there is no way to get around increasing the log level to DEBUG. The consequences are huge log files even if just specific loggers are set to a more verbose log level. Following the Shibboleth principle, the main information about

authorization is located in the SP log files, but also it contains some basic information about successful authentications, meaning that only authenticated users will create a new session entry on the SP logs.

8 Summary and Conclusions

The architecture developed is coherent with the identified requirements (Section 5), which are aligned to the necessities presented in the production scenarios (Section 3). The current status of this work shows a full-fledged specification of this approach to be in place, which is going to be implemented in the second phase of the project.

Summarizing, the main requirements for the architecture are: flexibility and extensibility, wide exchange of data, security, and reliable monitoring. Flexibility and extensibility can be reached in the sense that the architecture permits to instantiate or aggregate multiple components due to well-defined interfaces, *e.g.*, if the SP produces a huge amount of data, several Collectors and Accounting Clients can be instantiated in order to handle the demand. This is possible based on interfaces between Meters and Collector, and Accounting Clients and Accounting Servers. The architecture can be extended with new components, since the component's hierarchy (from the Meter to Accounting Server) is well-defined. Accounting records can be exchanged between domains inside a federation, and also domains from different federations. Agreements are needed to enable inter-federation exchange, however, the architecture presents an integrated approach to accounting and monitoring of Shibboleth. Regarding the security and data privacy, accounting records that are inherent to other domains have to be exchanged. Otherwise, all user's information remains inside its own domain. If data needs to be exchanged, the Accounting Server component is required to encrypt the transmitted information. Finally, regarding reliable monitoring, it is defined that Meter and Collectors have to respect performance requirements, not losing any information generated by IdP, SP, Service, or DS.

Moreover, the chosen scenarios brought valuable accounting attributes (Section 4) that will be included in the implemented solution. Regarding the Shibboleth analysis, mainly its logging capabilities were investigated which are the key for a solid monitoring base.

Since this deliverable reflects the first phase of the AMAAIS project, the second phase is dedicated to the technical implementation. Therefore, well-known protocols will be adapted to fit to the architecture's interface needs, and the developed Shibboleth analysis may help to clarify implementation decisions specifically regarding Meter and Collector components. These activities are the first steps towards the construction of an end-to-end prototype.

Terminology

IdP: Identity Provider.

SP: Service Provider.

Institution: A large important organization such as a university that, in the scope of this document, provide identity credentials.

Resource: An available supply that can be used/consumed, and, in the scope of this document, provided by Service Providers.

AAI environment: The entire set of conditions under which the Authentication and Authorization Infrastructure is operated, as it relates to the hardware (*e.g.*, servers, network segments), operating platform (*e.g.*, Shibboleth), or operating system.

WAYF: WAYF (”Where Are You From”) service have the goal to send a user to the Identity Provider of his Home Organization.

DS: Discovery Service.

Acknowledgement

This deliverable was made possible due to the large and open help of the members of the AMAAIS project. Many thanks to all of them.

References

[1] B. Pfitzmann and M. Waidner. Federated identity-management protocols. *LECTURE NOTES IN COMPUTER SCIENCE*, 3364:153, 2005.

[2] AMAAIS Project. *Accounting and Monitoring of AAI Services – Project's Website*, June 2009. Available at: http://www.csg.uzh.ch/research/amaais. Visited on: Jun. 2009.

[3] OASIS. *Security Assertion Markup Language (SAML)*, August 2009. Available at: http://www.oasis-open.org/committees/security. Visited on: Aug. 2009.

[4] Internet2. *OpenSAML*, August 2009. Available at: http://www.opensaml.org. Visited on: Aug. 2009.

[5] Shibboleth: an Internet 2 Project. *Shibboleth in Use*, June 2009. Available at: http://shibboleth.internet2.edu/shib-in-use.html. Visited on: Jun. 2009.

[6] SWITCH Website. *SWITCH – Swiss Academic and Research Network*, June 2009. Available at: http://www.switch.ch/. Visited on: Jun. 2009.

[7] SWITCH AAI. *SWITCH – Authentication and Authorization Infrastructure*, June 2009. Available at: http://www.switch.ch/aai/. Visited on: Jun. 2009.

[8] SWITCH. *Accounting for the Authentication and Authorization Infrastructure (AAI) – Pilot Study*, January 2006. Available at: http://www.switch.ch/aai/docs/AAI_Accounting_Pilot_Study.pdf. Visited on: Jun. 2009.

[9] SWITCH. *SWITCH AAI Resource Registry*, August 2009. Available at: http://www.switch.ch/aai/tools#resourceregistry. Visited on: Aug. 2009.

[10] ETH Zürich Website. *ETH – Swiss Federal Institute of Technology Zürich*, June 2009. Available at: http://www.ethz.ch/. Visited on: Jun. 2009.

[11] Shibboleth: an Internet 2 Project. *Shibboleth Documentation*, June 2009. Available at: https://spaces.internet2.edu/display/SHIB2/Home. Visited on: Jun. 2009.

[12] Internet2. *IdP Logging - Shibboleth 2 Documentation - Internet2 Wiki*, August 2009. Available at: https://spaces.internet2.edu/display/SHIB2/IdPLogging. Visited on: Aug. 2009.