Enhancing SWITCHaai with Micropayment Functionality for Swiss Universities

White Paper

Document management

Version/status:	1.0 / final	
Date:	6 September 2006	
Author(s):	Patrik Schnellmann	SWITCH
	Patrick Chénais	University of Berne
	André Redard	at rete ag
File name:	AAA_WhitePaper_v1_0.doc	
Replacing:		
Approved by:		

Table of Contents

Abstract

In the past, many Swiss universities have deployed SWITCHaai, the Shibboleth based infrastructure for authentication and authorization.

This white paper extends the concept of Shibboleth by adding (micro-)payment functionality. It shows how the proposed payment infrastructure could settle the usage of services provided by other organizations and how it could be integrated with existing local payment solutions based on electronic student card authentication.

1 Introduction

In 2001, SWITCH started a project for implementing an authentication and authorization infrastructure (AAI) for higher education in Switzerland, based on [Shibboleth]. Today, many Swiss universities have implemented [SWITCHaai], as it is called, and many resources can be accessed by users of different organization using it as authentication and authorization mechanism.

Figure 1: Authentication and Authorization Infrastructure (SWITCHaai)

The main purpose of SWITCHaai is to authenticate a user by the authentication system of the user's home organization and to authorize the user to access a particular resource. The authorization decision is made by the resource based on authorization attributes received from the authentication system.

SWITCH believes that optional accounting services are important in a shared and distributed environment in order to understand how resources are used and to be able to split costs of shared resources among their users. Nevertheless, accounting applications like billing, usage reporting etc. were initially deliberately excluded from the SWITCHaai for various reasons. Yet, already the SWITCHaai preparatory study [AAI Study] postulated that it would have to interact with these applications.

A billing system for a resource, will need to know e.g. who has used the resource and how it has been used (how many transactions, which information, how long, etc., depending on the tariff model for that resource).

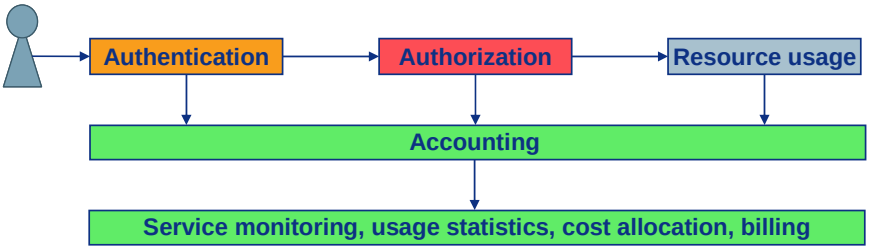

Figure 2: Accounting system interacting with SWITCHaai and the resource

SWITCHaai is able to answer the question of who has accessed the resource because it is able to link the information the resource has about users (e.g. anonymous user IDs) back to real persons only known by the home organization. However, SWITCHaai has no information on the question of how the resource was used. This answer can only be given by the resource itself, which can measure the interactions between a user and the resource.

Therefore, it was a logical step that SWITCH started a project called "Accounting for the Authentication and Authorization Infrastructure (AAI)" in September 2005. In the first project phase, the requirements of home organizations, service providers and SWITCH have been documented and accounting technologies compatible with SWITCHaai have been investigated (cf. [AAAStudy]).

This white paper goes a step further. It shows the potential of the combination of micropayments with the SWITCHaai in a Swiss-wide context. It also addresses implementation issues specific to Shibboleth, which is used in SWITCHaai. A general model is proposed how to use the current possibilities of SWITCHaai for the inter-organizational billing of services.

However, the model presented in this paper is not limited to billing with AAI. It may well be used as a model for the inter-organizational exchange of accounting information, which is in fact a part of the proposed model. Billing is imposed on the model, in terms of mathematicians, “without loss of generality”.

2 Micropayment Use Cases

While accounting has been a necessity for commercial telecom providers or IT outsourcers in order to charge for their services, accounting has been rarely implemented in the academic environment. Since organizations cooperate in developing new applications and sharing resources, cost for implementation and operation have to be allocated to the parties involved. The following sections show three use cases, where micropayment functionality is a necessity in order to offer these services to users of other organizations.

2.1 Using and paying for a printing service at another university

User	Student S from university A, attending a lecture at university B
Service	Printing service offered by university B: The service offers a web-based and AAI-enabled front-end which allows user S to authenticate himself and to start his queued print-jobs.
Financial authorization	Before the printing application provides its service to student S, it requests a financial authorization from university A. Due to the nature of a printing service, it is unknown how many pages the student is going to print during the session. Therefore, the printing service will request the authorization of a credit limit, e.g. CHF 10.-. The financial authorization statement created by university A may depend on the credit rating and the account balance of student S as well as on other parameters.
Accounting and billing	The printing service logs the number of printed pages together with the user's identity and the name of his home organization. Periodically, e.g. once per session, the log is evaluated and a settlement instruction is sent to the user's home organization, i.e. university A. The settlement instruction contains information about the service provider, the federation member providing the service, the identity of the user and detailed information about the service consumed. The printing service controls also that the financial limit is not exceeded. University A charges the cost to its student S. If university A has implemented a student card system with payment functionality, it directly charges the student's account. Otherwise it will send a bill to student S, e.g. at the end of the semester. Periodically, university B bills university A for all services provided to users of university A.

The following figure shows the main interaction between the printing service and proposed micropayment solution.

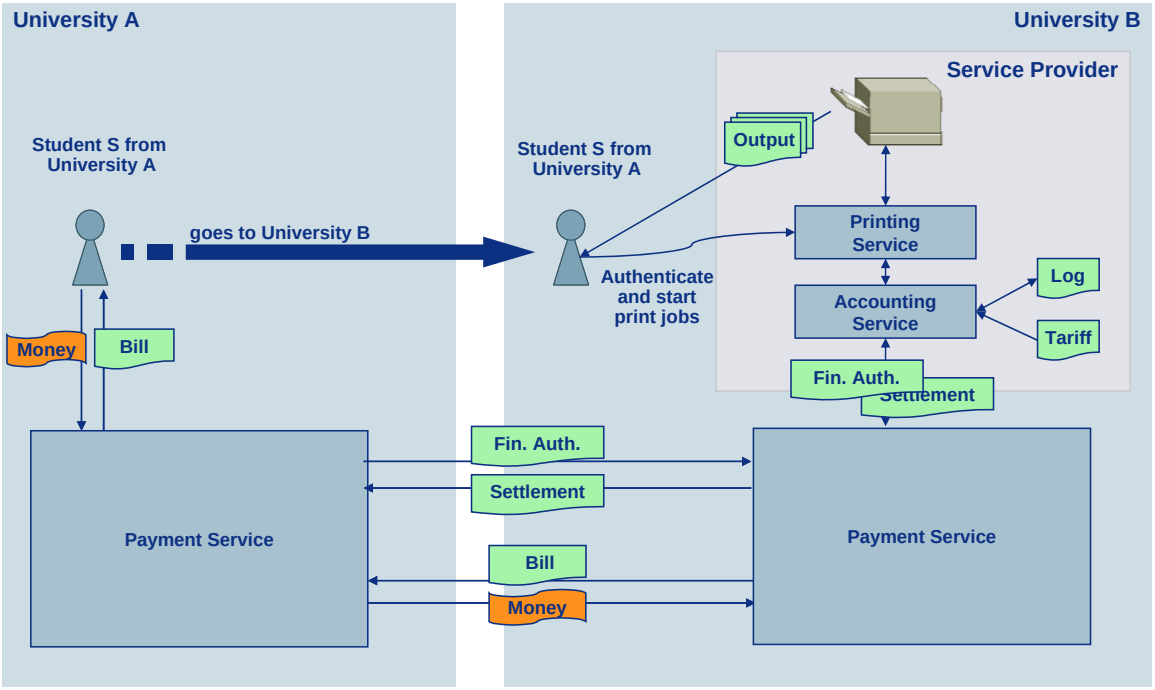

Figure 3: Accounting and billing solution

2.2 Library's scanning service

User	Researcher R working for university A
Service	Scanning service of library B: Library B offers an AAI-enabled web-application which allows researcher R to order a scanned copy of an article in a journal stored at library B. The scanned article is sent as pdf-file to researcher R.
Financial authorization	The scanning service requests a financial authorization from university A. In this use case, the cost for providing the service can be calculated up-front, e.g. based on the number of scanned pages and a fixed service fee; and the service requests an authorization for the calculated amount.
Accounting and billing	Accounting and billing works the same way as defined for the printing service, except that the amount due is not charged to the researcher but to the researcher's cost center.

2.3 WebSMS service

User	Member of university A
Service	Web-based SMS service provided by SWITCH: SWITCH offers an AAI-enabled web-application which allows all users belonging to a SWITCHaai federation member to send SMS and to maintain a personal phone book.
Financial authorization, accounting and billing	Financial authorization, accounting and billing works the same way as defined for the use case of the printing service. Depending on the affiliation of the user, university A will charge the user (students) or the user's cost center (staff members).

3 Micropayment and SWITCHaai

This section proposes a model to add accounting to a Shibboleth-based AAI. The model was driven by the intention to use as much of the features of the current implementation of Shibboleth version 1.3.

3.1 AAI auditing

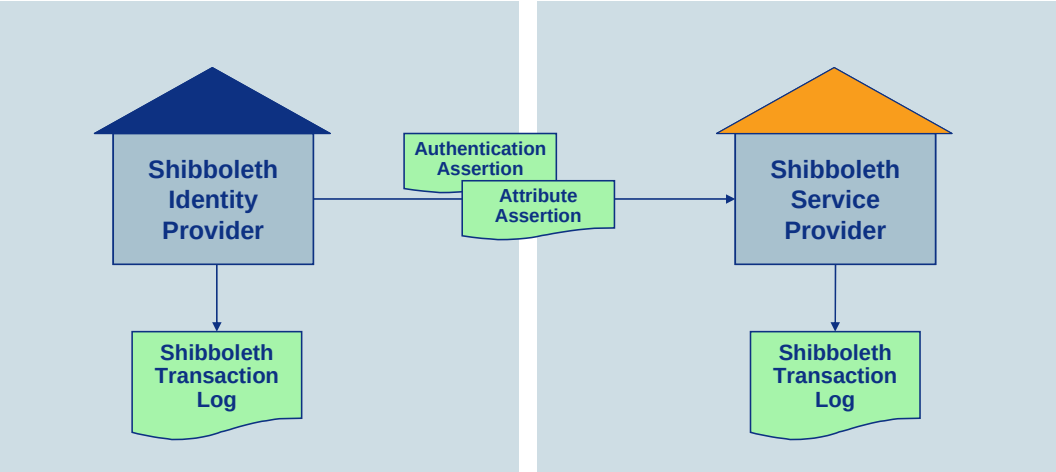

Figure 4: Shibboleth 1.3 – assertions and transaction logs

The participants of the SWITCHaai agree to behave according to common rules which are stated in the [AAIPolicy]. These rules are a fundamental basis for trust in the SWITCHaai community. If a party wants to be sure other parties behave in conformance with the rules, some piece of evidence is needed. For the technical aspects, there are log files which can serve as proof.

The Shibboleth Identity Provider (IdP) and Service Provider (SP) in version 1.3 both generate a transaction log file for each issued assertion (cf. Appendix A). The log files on the IdP and the SP can be matched by means of a unique identifier: The AssertionHandle (referred to as “Name Identifier”) is saved in the transaction log files. This means that the whole process of exchanging authentication and authorization information between the IdP and the SP is auditable.

3.2 Micropayment functionality

This section proposes a model of how payment functionality can be added to SWITCHaai and how security can be implemented based on SWITCHaai federation metadata.

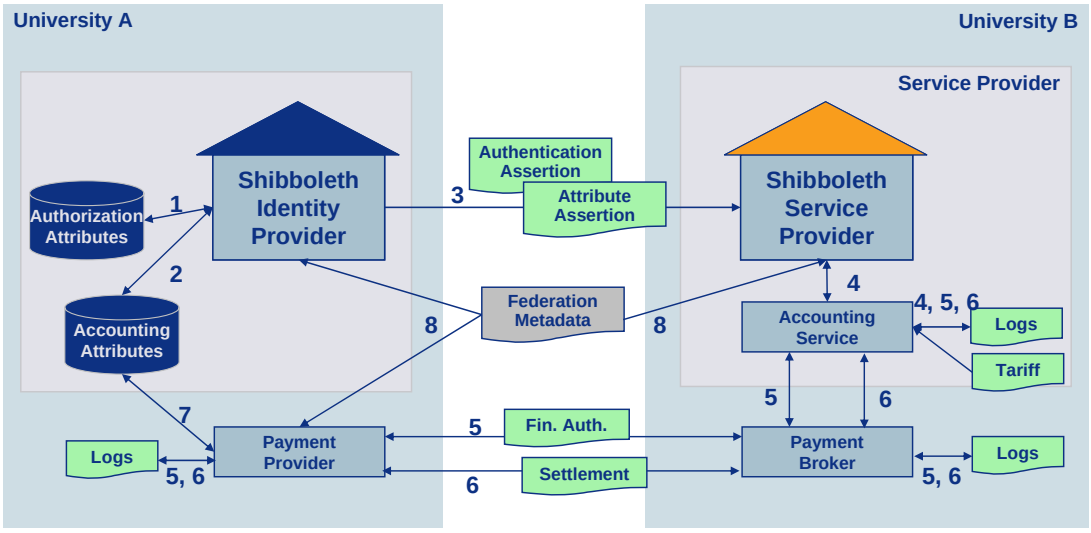

Figure 5: Adding micropayment functionality to Shibboleth

1) The Shibboleth Identity Provider (IdP resolver) fetches authorization attributes from the attribute store over a JDBC / JNDI connection.
   Attributes for accounting purposes can be stored in a database separate from the authorization attributes or in the same database. These attributes are to be managed by the home organization's account and credit manager. Possible attributes are: account number, cost centre identification, etc.
   Due to the design of Shibboleth, attributes are only transferred during the authentication and authorization process (initialization of the session). Therefore, the usefulness for a payment solution is limited.

2) The user's authorization and accounting attributes are packed in a SAML assertion which can be signed using an X.509 certificate. The IdP uses its server certificate to sign the SAML assertion. The different profiles of Shibboleth will have to be taken into account: if the Browser/Artifact profile ("push model") is used, the assertion will be signed as the attributes are sent to the SP via the user's web browser.
   In case of the Browser/POST profile ("pull model"), the Shibboleth SP has to make sure it gets a signed assertion. This requirement is stated in the federation metadata.

3) During a user's session, the service provider application will generate resource usage records. The SAML assertion is logged along with the usage records. A session record is composed of several usage records, the SAML assertion, a financial authorization and a cost record. It is signed by the Shibboleth SP using its X.509 certificate.

4) Financial authorization: Via the local payment broker, the payment provider at the home organization is asked for the financial authorization of a specific amount. The payment provider answers with a granted credit limit and its validity of time.

5) Settlement: The session records are sent to the user's home organization for settlement. The home organization can verify the integrity of the record by checking its signature. The signed assertion in the records proves that the user has been authenticated by the Shibboleth IdP.

6) The federation metadata identifies the participants of the SWITCHaai and micropayment infrastructure and names the trust anchors (X.509 certificates of Shibboleth servers / certificate authorities). The metadata enables mutual trust between the IdP, the SP, the payment broker and the payment provider. The payment broker can get the address of a payment provider (e.g. the url of a web service) based on attributes, like the name of the user's home organization or the identifier of the involved IdP. The payment provider can verify the correctness of data signed by the SP.

Accounting service, payment broker and payment service log the transferred data for further processing (cf. 3.3 and 3.4).

The following table defines the main functionality of each new system component shown in Figure 5:

Accounting service	This system component is meant as a standard module run by the service provider. It interacts with Shibboleth and the Shibboleth protected web-application, requests financial authorization from the payment provider via payment broker, manages credit limits per session, calculates service cost and requests settlement. Service usage, financial authorization and settlement statements are logged.
Payment broker	The payment broker system component is operated by each federation member providing services that are charged. It interacts with all accounting services of this federation member and all payment providers. It accepts financial authorization and settlement requests from the accounting services and sends these requests to the corresponding payment provider (depending on the user's home organization). The payment provider's answers are sent back to the requesting accounting services.

	Financial authorization and settlement statements are logged.
Payment provider	The payment provider is operated by each home organization, providing micropayment functionality to their users. It handles financial authorization requests and settlement requests received from other payment brokers. It either debits directly the user's account, e.g. managed by a student card system, or stores the settlement statement for future billing and updates the credit limit per user.
	Financial authorization and settlement statements are logged.
Account and Credit Manager	Handles the account balance and the granted credits for each user and cost center. Since each user may use more than one service at the same time, it has to be able to manage several granted credits per user in parallel.
	The implementation of the account and credit manager depends on the home organization's infrastructure. It can be implemented as part of a card solution providing payment functionality or as stand-alone application (cf. 3.3 and 3.4). Shibboleth user accounts (i.e. username, password) and financial accounts have to be linked together, e.g. by storing the user's financial account number as part of the user's accounting attributes and vice versa.
Federation Metadata	For each payment broker and payment provider, their identifier, server certificate, etc. are stored in the federation metadata. To remain compatible with Shibboleth, the payment related metadata should be stored in a separate file.

3.3 Post-processing by the payment broker

Periodically, e.g. once per day or once per month, the logged settlement statements of a payment broker are evaluated and aggregated per service provider and per home organization. The resulting amounts are credited to the service providers' accounts and debited to the home organizations' accounts. Then, university B sends a bill to university A and the amount due is transferred to university B. The payment provider of university A can provide aggregated settlement statements per payment broker which allows university A to control the received bill from university B.

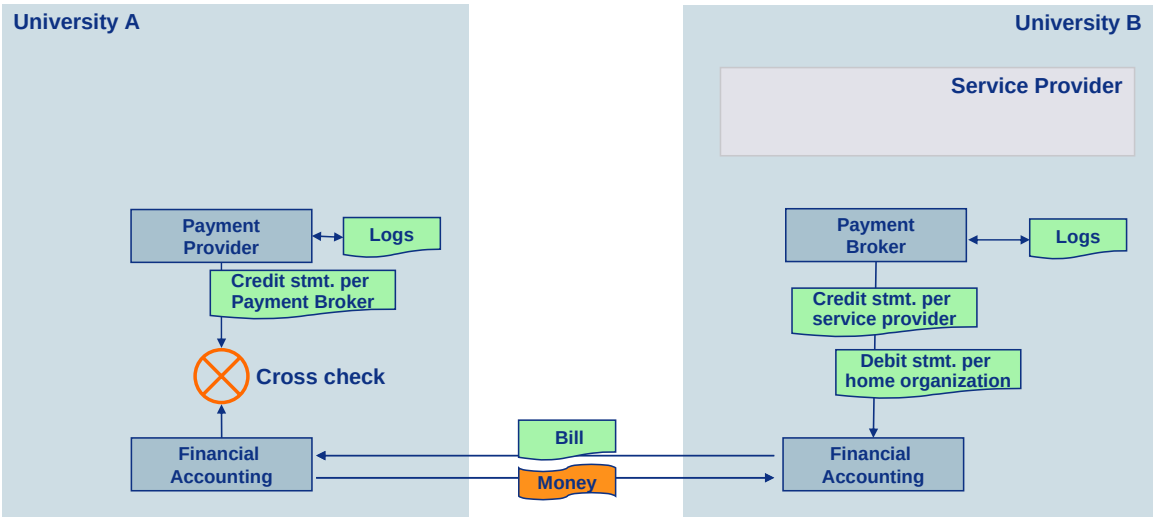

Figure 6: Post-processing by the payment broker

The service provider can evaluate the log of the accounting service and crosscheck the result with the credit statement provided by the payment broker.

3.4 Post-processing by the payment provider

3.4.1 University operating a card solution with payment functionality

Some of the Swiss universities are planning to implement (or have already implemented) new card solutions with payment functionality for their students and employees. In this case, the card solution maintains an account for each user.

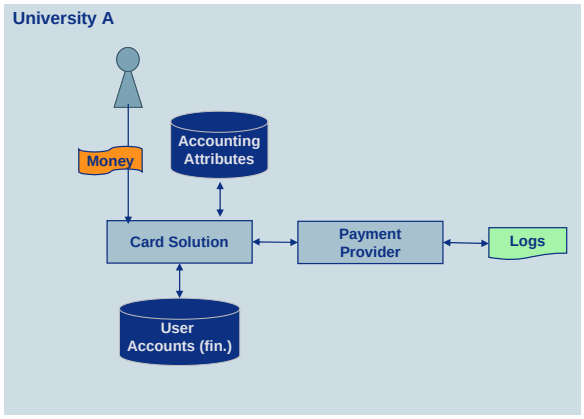

Figure 7 Post-processing at payment provider with a card solution

The users pay into their account in advance. Ideally, the payment provider can assign the task to manage the accounts, to settle the payments and to monitor credit limits to the card solution. The card solution has also to maintain the information made available to Shibboleth via the accounting attributes database, e.g. account number per user.

3.4.2 Post-processing without card solution

Periodically, the payment provider evaluates its log file and aggregates the settlement statements per payment broker and per user. The resulting amounts are credited to the payment broker and debited to the user. The user receives a bill and pays the amount due. The credit statement per payment broker can be used to check the bill received from other organizations (cf. 3.3).

Due to the absence of a card solution, a system component “account and credit manager” has to be added which maintains account balances and controls credit limits.

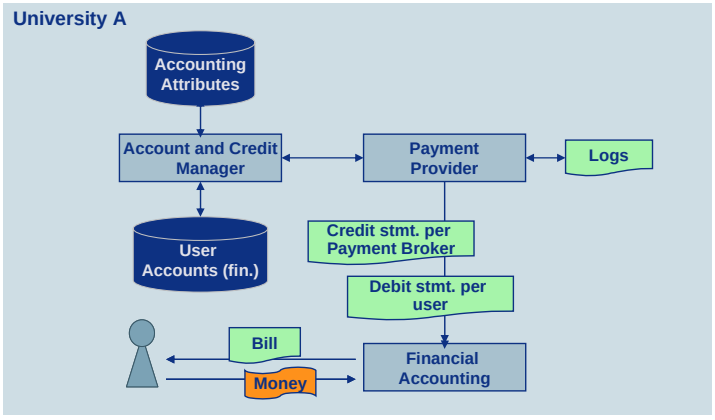

Figure 8 Post-processing at payment provider without a card solution

3.4.3 Charging a cost center

As described in the use case of the scanning service and the SMS service, the micropayment solution should allow charging service costs not only to users but also to users' cost centers. If the employees' cost center identification is stored as an accounting attribute, released to the service provider as a Shibboleth attribute assertion and is part of each financial authentication and settlement request, then the process and interactions above can be implemented in a similar way to manage the charging of cost centers (i.e. credit limits, account balances, debit statements per cost center).

4 Software Components

In addition to the open source Shibboleth components, developed by Internet2 (cf [Shibboleth]), two new components have to be specified and implemented:

- payment broker
- payment provider

It is assumed that each university offering payment functionality operates these two components. The payment provider has to be integrated with the local account and credit manager (e.g. the card solution) by implementing interface A.

Service providers offering services at cost have to integrate their services with the payment broker (accounting service component and interface B). The implementation of the accounting service components depends on the type of service.

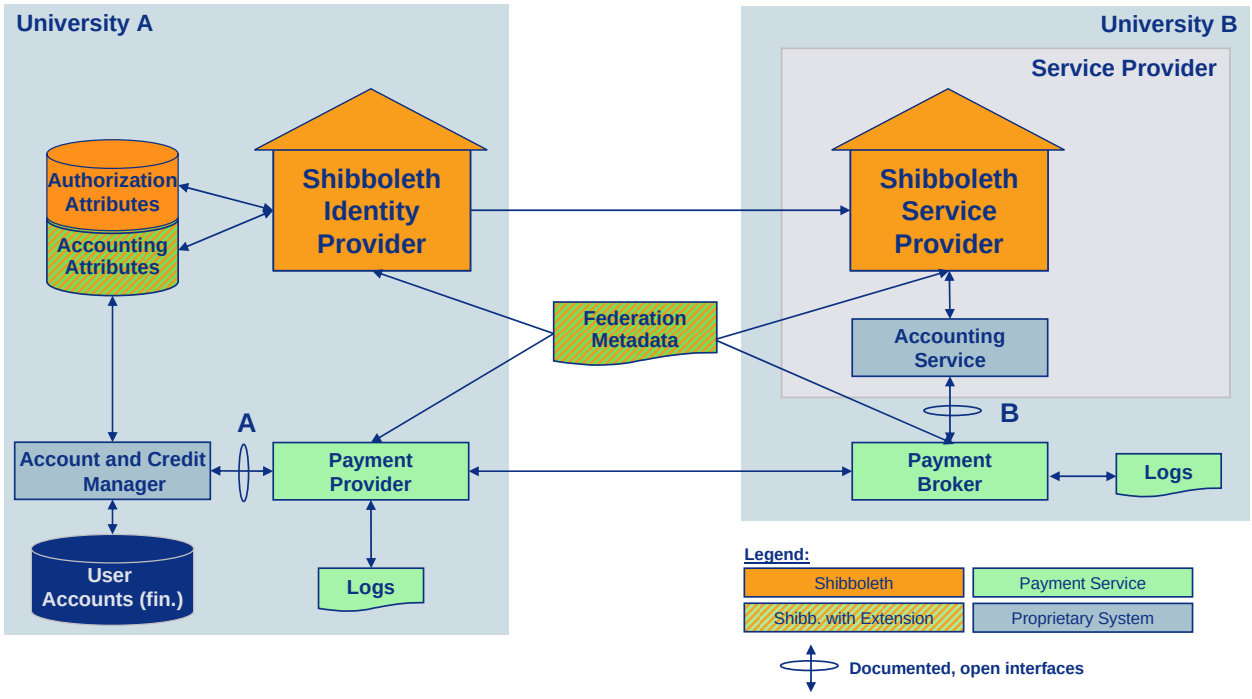

Figure 9 Software components

5 Micropayment and Card Based Authentication

For universities which have deployed compatible student card solutions the payment system could be further developed to provide also payment services for users authenticated by a student card instead of Shibboleth (e.g. student S from university A pays with a student card issued by his university at a vending machine provided by university B).

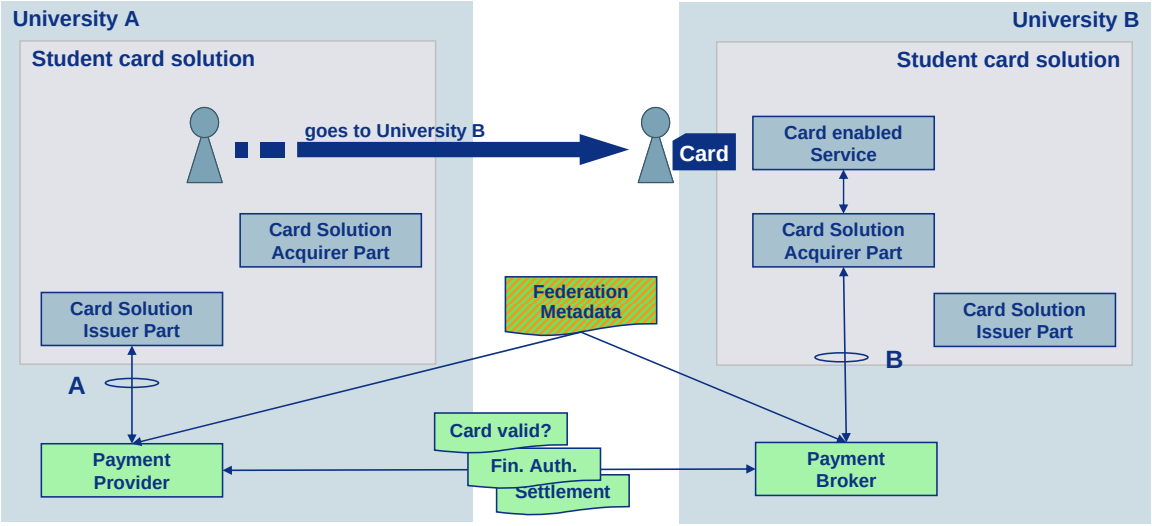

Figure 10 Micropayment and card based authentication

The card solution of university B has to be able to distinguish between cards issued by university B itself and cards issued by another university. Since there is no “WAYF” functionality, the user’s home organization, i.e. the issuer of the card, has to be stored on the card itself (e.g. name/identifier of home organization stored on the card or specific card number range per home organization).

In addition to the transactions defined in chapter 3 (financial authorization, settlement), the card solution of university B has also to be able check the validity of the foreign card.

6 Micropayment Security

The main risks faced by the user are liability for unauthorized payment and not receiving the requested service. The service providers risk not being paid by the users' home organizations, while the home organizations risk not receiving the outstanding money from the users, especially in cases where user and service provider are in dispute. Protocols which cannot clearly identify which party was at fault are likely to incur substantial customer service costs.

Therefore, the payment solution and the protocols involved have to be designed accurately to fulfill the following basic security requirements:

- Privacy
- Authentication
- Integrity
- Nonrepudiation

We believe that an adequate security architecture should be and can be built based on the technologies and concepts already used within Shibboleth:

- Authentication of the systems involved based on federation metadata
- Secure communication over TLS between the systems involved
- Messages signed with X.509 server certificates; signature verification based on federation metadata and the approved Certification Authorities
- User authentication based on Shibboleth
- Session identification based on the Shibboleth handle as described in chapter 3.1

The details of this security architecture would have to be designed in a next step and audited against typical threats for payment systems (man-in-the-middle-attack, replay attacks, faked authorization or settlement instructions by third parties, etc.).

Appendix A Shibboleth transaction log file examples

The examples below show the log entries generated for the following process:
SP “https://kohala.switch.ch/shibboleth” receives the Authentication and Attribute assertions issued by IdP “urn:mace:switch.ch:aaitest:dukono” for user “demouser”. The SP does only log the names of the attributes received and the number of values, but not the attribute values itself.

The log entries on the IdP and the SP can be linked by means of the “Name Identifier”. On the SP, an application can get access to the “Name Identifier” by extracting them from the SAML Attribute Assertion. In the example below, the Shibboleth session on the SP is identified with the identifier `ID: _44c918c82258ed0ac46a760883c7f5eb`.

Transaction Log of the Shibboleth Identity Provider

```
2006-06-28 08:07:10,396 Authentication assertion issued to provider
(https://kohala.switch.ch/shibboleth) on behalf of principal (demouser). Name Identifier:
(_146f3738a362afe5e03f9daa914f8738). Name Identifier Format:
(urn:mace:shibboleth:1.0:nameIdentifier).

2006-06-28 08:07:11,403 Attribute assertion issued to provider
(https://kohala.switch.ch/shibboleth) on behalf of principal (demouser).
```

Transaction Log of the Shibboleth Service Provider

```
2006-06-28 08:07:10 New session (ID: _44c918c82258ed0ac46a760883c7f5eb) with (applica-
tionId: default) for principal from (IdP: urn:mace:switch.ch:aaitest:dukono.switch.ch) at
(ClientAddress: 2001:620:0:x:x:x:x:x) with (NameIdentifier:
_146f3738a362afe5e03f9daa914f8738)

2006-06-28 08:07:10 Making attribute query for session (ID:
_44c918c82258ed0ac46a760883c7f5eb) on (applicationId: default) for principal from (IdP:
urn:mace:switch.ch:aaitest:dukono.switch.ch)

2006-06-28 08:07:11 Caching the following attributes after AAP applied for session (ID:
_44c918c82258ed0ac46a760883c7f5eb) on (applicationId: default) for principal from (IdP:
urn:mace:switch.ch:aaitest:dukono.switch.ch) {

2006-06-28 08:07:11  urn:mace:dir:attribute-def:eduPersonEntitlement (2 values)

2006-06-28 08:07:11  urn:mace:switch.ch:attribute-def:swissEduPersonUniqueID (1 values)

2006-06-28 08:07:11  urn:mace:switch.ch:attribute-def:swissEduPersonHomeOrganizationType (1
values)

2006-06-28 08:07:11  urn:mace:dir:attribute-def:sn (1 values)

2006-06-28 08:07:11  urn:mace:dir:attribute-def:eduPersonAffiliation (1 values)

2006-06-28 08:07:11  urn:mace:dir:attribute-def:givenName (1 values)

2006-06-28 08:07:11  urn:mace:switch.ch:attribute-def:swissEduPersonHomeOrganization (1
values)

2006-06-28 08:07:11 }

2006-06-28 08:07:11 Successful attribute query for session (ID:
_44c918c82258ed0ac46a760883c7f5eb)
```

Appendix B References

[AAAStudy]	Accounting for the Authentication and Authorization Infrastructure, Pilot Study, 1.0, 5-Jan-2006
[AAI Study]	AAI Preparatory Study, 1.0, 15-Jul-02 http://www.switch.ch/aai/docs/AAI_Study_v10a.pdf
[AAIAttr]	AAI Authorization Attributes, Version 1.1, 15-JAN-2004 http://www.switch.ch/aai/docs/AAI_Attr_Specs.pdf
[AAIPolicy]	AAI Policy, Version 1.12, 7 July 2004 http://www.switch.ch/aai/docs/AAI_Policy.pdf
[SAML]	Security Assertion Markup Language (SAML) http://www.oasis-open.org
[Shibboleth]	Shibboleth Project http://shibboleth.internet2.edu
[SWITCHaai]	SWITCH – Authentication and Authorization Infrastructure http://www.switch.ch/aai/