CHALMERS

A proposal of a method for evaluating third-party authentication services

Master of Science Thesis in the Programme Software Engineering and Technology

ERIK JOSEFSSON
ERIK STENBÄCKA

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, October 2011

A proposal of a method for evaluating third-party authentication services

ERIK JOSEFSSON
ERIK STENBÄCKA

Examiner: MIROSLAW STARON
Supervisor: ROBERT FELDT

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden October 2011

Sammanfattning

Datasäkerhet som akademiskt fält är väl studerat, eftersom konsekvenserna av att misslyckas kan bli katastrofala. Om en extern användare kommer åt information eller funktioner hon inte borde kunna komma åt kan det ha förödande följder. Tredjepartsautentisering är ett växande fenomen som eftersträvar att lösa problemet med att användare måste registrera sig på de flesta hemsidor de vill få tillgång till. Med ett konto hos en tredjepartsautentiseringstjänst kan en användare komma åt alla hemsidor som stödjer den tjänsten utan att behöva registrera sig på hemsidan. Möjligheterna och begränsningarna med tredjepartsautentisering är inte fullständigt kartlagda i dag och det saknas ett gemensamt kommunikationsprotokoll för att autentisera användare.

Målet med det här examensarbetet är att utöka kunskapen om dessa tredjepartstjänster genom att studera den tillgängliga litteraturen inom området och sammanföra denna till en utvärderingsmetod för tredjepartsautentiseringstjänster. Vidare så kommer rapporten att utforska möjligheterna att gå runt problemet med att det inte finns ett gemensamt protokoll för användarautentisering genom att skapa en insticksbaserad autentiseringslösning som enbart använder sig utav tredjepartsautentiseringstjänst.

En utvärderingsmetod, som baseras på de centrala koncepten inom användarautentiseringsområdet, kommer att presenteras i rapporten. Dessutom presenteras en konceptimplementering av den insticksbaserade autentiseringslösningen, som visar att det är möjligt att kringgå problemen med att det inte finns något gemensamt kommunikationsprotokoll.

Abstract

The security field is a highly studied area of knowledge, since the consequences of failing can be catastrophic; if an external user accesses information or function she should not be able to access. Third–party authentication is a growing concept that tries to remedy the problem of users having to register at most websites they want to access. With an account at a third–party authentication service a user can access all websites that support the third–party service without having to register there. While this seems like a good architecture are the capabilities and limitations of third–party services not well understood and there are no common protocols for authenticating users.

This master thesis aims at increasing the knowledge about these services by reviews current literature in the field in order to define a method for evaluating third–party authentication services. Furthermore, in the scope of the thesis is to explore the possibility of circumventing the problem that there is no common protocol for authenticating users by creating a plug–in based authentication solution that utilizes third–party authentication services for user authentication.

An evaluation method that tries to to capture the essential aspects of third–party user authentication is proposed. In addition a proof–of–concept implementation of the previously mentioned plug–in based authentication solution is implemented to show that it is possible to circumvent the described problem.

Acknowledgements

We would like to thank Pagero for providing us with facilities and equipment at their office. We would like to especially thank Thom Birkeland and Fredrik Dyrkell who have together acted as our supervisors at the company.
We would also like to thank our supervisor Robert Feldt for providing us with feedback and directions on how to improve our thesis report.

Contents

1 Introduction **1**
- 1.1 Definitions . . . 1
  - 1.1.1 Identity and Authentication . . . 1
  - 1.1.2 Identity Provider and Service Provider . . . 1
  - 1.1.3 Identity Management Solution . . . 2
  - 1.1.4 Plug–in Authentication Security Service . . . 2
- 1.2 The Company . . . 3
- 1.3 Background . . . 3
- 1.4 Purpose . . . 5
  - 1.4.1 Research questions . . . 5
- 1.5 Limitations . . . 5

2 Method **6**
- 2.1 PASS design and evaluation . . . 6
- 2.2 Case Study . . . 6
  - 2.2.1 Features . . . 7

3 Evaluation of the IMS field **10**
- 3.1 Criteria by Myllyniemi . . . 10
- 3.2 Method by Straub and Baier . . . 10
- 3.3 Kylau, Thomas, Menzel and Meinel . . . 11
- 3.4 Trust Requirements in Identity Management . . . 12
- 3.5 Summary of Presented Papers . . . 13

4 Plug–in Authentication Security Service **13**
- 4.1 User Authentication . . . 13
- 4.2 Security Level . . . 13
- 4.3 Changing Authentication . . . 14

5 Result **15**
- 5.1 Plug–in Authentication Security Service . . . 15
  - 5.1.1 User Authentication . . . 15
  - 5.1.2 Multiple Security Levels . . . 16
  - 5.1.3 New features for supporting multiple security levels . . . 17
  - 5.1.4 PASS implementation . . . 18
  - 5.1.5 Demonstration feedback . . . 21
- 5.2 Authentication Service Evaluation . . . 22
  - 5.2.1 Evaluation Scales . . . 22
  - 5.2.2 The ETA method . . . 26
  - 5.2.3 Authentication Service Categories . . . 27
- 5.3 Case Study . . . 28
  - 5.3.1 Security . . . 28
  - 5.3.2 Ease of Deployment . . . 32
  - 5.3.3 Certainty of Identity . . . 32
  - 5.3.4 Evaluation Conclusion . . . 33

6 Conclusion **34**
6.1 Authentication Service Evaluation 34
6.2 Plug–in Authentication Security Service 34
6.3 Future work 35

Bibliography **37**

A Identity Management Solutions **39**
A.1 OpenID 39
A.2 Kalmar2 39
A.3 Facebook Connect 40
A.4 BankID.se 41

B Identity Management Solutions Evaluation **42**
B.1 Security 42
B.2 Ease of Deployment 42
B.3 Certainty of Identity 43
B.4 User Adoption 43
B.5 Summary 44

1 Introduction

This section will present the background and purpose if the thesis. It will start by presenting some definitions of concepts relevant to the report. After that the background followed by the purpose will be presented.

1.1 Definitions

This section will give definitions to terms used in this report. Some other terms are described in short in the glossary.

1.1.1 Identity and Authentication

Agudo–Ruiz [1] provides a good description of what a digital identity and identity management are. He refers to RFC 2828 [2] when he describes an entity as a person or an automated process that incorporates a specific set of capabilities (in this report entities will be referred to as users.). A user can have several identities, each uniquely identifying her in a given context. An identity is associated with a set of access rights or privileges within the given context.

Authentication is the act of attesting ownership of an identity using some credentials that connects the entity and the identity, Isaac refers to this as an ID–card. A real world example of this is the use of a passport at the passport control. By showing her passport the bearer can prove that the person holding the passport has the name and nationality indicated on the passport. Different IMSs have different identity and ID–card concepts; some services use email addresses for identification while others use unique ids generated using some domain–specific policy.

'Certainty of identity' is a relevant term with the meaning of how sure the system can be that the identity provided contains correct information about the user. In a scenario of low certainty of identity the entity can herself assert any information in the digital identity. On the other hand, high certainty can be achieved, for example, if the information of the identity is provided by a government authority. This aspect can be deemed of high interest to some services (e.g. money transferring) while most are satisfied with a way to uniquely identify users.

1.1.2 Identity Provider and Service Provider

An Identity Provider (IdP), as the name suggests, provides identification for a site or application, known in this context as a Service Provider (SP). The communication protocol between these parties varies between providers. However, the overarching communication flow is often similar; here follows a description of a service access scenario (see Figure 1).

The user starts out on the SP and requests to log in. The user is redirected to the IdP. The user logs into the IdP using her existing credentials, independent of the SP. After the user has been authenticated at the IdP, she is sent back to the SP with the result from the IdP. The response is validated by the SP, and if found valid her login is accepted.

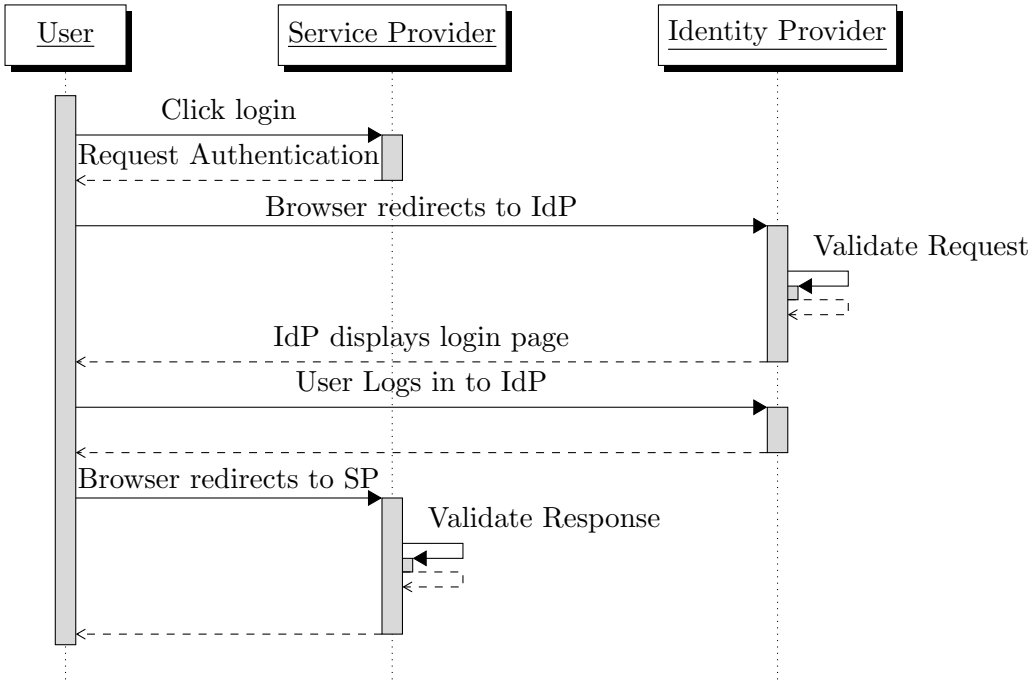

Figure 1: Using a IdP for authentication

1.1.3 Identity Management Solution

Identity Management Solution (IMS) is a term this report will use for any service that provides a method to handle identities. The IMS contains all the information related to authentication, such as identity, password and any other information relevant to that IMS. An IMS consists of one or more IdP, that is the service that actually issue authentication assertions.

Different IMSs have different structures; two common structures are identity federation and single IdP. An identity federation is when several services, with their own internal user management system, come together to share authentications and cooperate as a single IMS. In an identity federation, a user can authenticate with his affiliated provider and then gain access to services offered by the other federation members. The single IdP is a centralized solution where all authentications are done against a single point. While these two structures are common there also exist other structures.

1.1.4 Plug–in Authentication Security Service

Plug–in Authentication Security Service (PASS) is a service for authentication handling, introduced in this thesis, which supports multiple IMSs. A PASS is pre–configured with a set of IMS that can be used to authenticate users. The plug–in part of PASS indicates that it is designed to support any type and any number of IMSs. However the IMSs need to be determined by the owner of the system that PASS is used with. That is, IMSs that are unknown to the PASS cannot be used to authenticate users. In this system, a user may use several of her identities as login options to access the service. The different identities used to log in all give access to the user's account, if they are associated properly. The IMSs

supported are built in a plug–in architecture to support easy addition of new IMSs.

1.2 The Company

This report is conducted for a company that has a system where business clients are able to send documents to other businesses, strictly B2B, but at this point there is only a limited possibility to send such documents to consumers and no platform for consumers to utilize. The company is now looking into creating such a platform.

The company has looked into ways of making the platform attractive and easy to access for consumers. One way to make people more prone to using this service is to minimize the effort required to get started with it. The company has identified the registration step as an obstacle, and wants to minimize the effort needed in this step. On today's market, there is a variety of third–party authentication services that can take the responsibility of authentication users away from the service. Using these services could potentially be a solution to the registration problem. The goal of this thesis is to evaluate the possibility of out-sourcing the user authentication to these services.

1.3 Background

Many websites on the Internet today require some form of knowledge about the people visiting the site. Some websites are only interested in being able to connect a user–name with an identity and to make the same connection on future visits (e.g. blog comments), while other websites require knowledge of a physical person, that can be held accountable for her actions (e.g. e–commerce). A common denominator for all these services is that the user has to register at the first visit to the site, possibly providing some information about herself, and at later visits authenticate the ownership of a registration. Due to the large amount of services available today are the users required to register and manage a large amount of accounts. This causes problems for the user as they have to remember the authentication information used at all the services, and they also needs to keep their information up to date on all services.

To deal with these problems a variety of so called Identity Management Services (IMS) (see subsubsection 1.1.3) have been deployed on the Internet, but also for other systems. They enable users to authenticate themselves on one domain and thereafter gain access to resources on other domains. The user stores her information at the IMS, which in turn provide the information to services (called service provider) when requested by the user. Using these IMSs would solve the registration problem. However since there are multiple IMSs in use today a service provider cannot only support one of these and expect all users to be able to access the service.

People have different identities and roles in different contexts (e.g. family and work) and will probably reflect this in their digital life. People in general will want to have more than one account, even if those accounts are IMS account. Also, an IMS have a lot of knowledge and power over the users that utilize the IMS. This can cause privacy concerns for the users, particularly if the IMS have access to all people; it would allow one actor to know what services everyone is accessing. Because of these reasons there will most likely never be one IMS that incorporate all users.

Different IMSs use different communication protocols, which are not understood by other IMSs. Attempts have been made to create meta–languages for enabling the same authentication request to be understood by different IMSs [3] [4]. Looking at this research and other computer related fields, such as computer hardware, it seems possible that a protocol that works with most IMSs will be available in the future. However, such a protocol is not used today so the question of what a company that wants to utilize the power of third-party authentication today can do remains.

This is a question this thesis seeks to answer. In particular, the possibility of circumventing the problem on the service provider's side will be investigated. Today there are software libraries that enable support for specific IMSs in authentication services. However such a solution cannot scale with other IMSs, so there is still a need for a more dynamic and scalable solution. The result of this investigation will primarily be of interest for companies but could also provide some useful insights to the academic research in the field.

As long as there is no common communication protocol amongst the IMSs the service providers will have to develop support for each IMS separately. Implementing support for all IMSs is an unreasonable solution, considering the magnitude of such a project. The stakeholders will instead have to choose which IMSs to support. There has been a number of research papers published relating to the selection of authentication services. Myllyniemi [5] proposes a number of criteria that should be used when evaluating identity systems and Thomas et. al. [6] have developed a mathematical formula for calculating the level of trust for authentication services, just to mention a few.

Third–party identity management is not a mature field and, as highlighted in [7], there are still problems that need to be researched. However as the theme of this thesis is to investigate the capabilities of IMSs available today, the focus will be on what kind of services that can benefit from the IMSs available today. A set of aspects that is relevant for this kind of evaluation is described briefly below and in more detail later in the report.

The identity verification in the authentication services' registration step is highly interesting; some services require a certain knowledge of who the user actually is, for instance when financial transactions are involved. A service that lets its users provide their identity information without any validation of its correctness cannot be relied upon to have reliable information. A service that uses e.g. a governmental database for its verification can safely claim that the user is who she claims to be.

The act of authentication in every login is, of course, also interesting and very relevant. There are a number of ways a user can attest that she is the owner of the account she is trying to access. A common way of asserting the ownership of an identity is a combination of a username and a password. This authentication technique is however less secure than other techniques and high security demanding services often use some other technique.

All authentication techniques can be categories into one of the following categories: 'something you know', 'something you have', and 'something you are'. The first kind contains password authentication; the second implies physical, or software, objects required to login; and the third kind consists of biometrics.

One example of a solution of the second kind is the Swedish BankID, deployed by a collaboration of banks in Sweden, which uses a Public Key Infrastructure solution. While the basic username/password login may be enough for some functions (blog comment, and so

on), a solution with a higher security may be needed for other functions (Internet banking for instance).

There is no IMS today that incorporates all users. It is safe to assume that such an IMS will not be available within a foreseeable future. Therefore, any choice of IMS will exclude a number of users, namely those that do not have an account at that IMS. A service provider obviously wants to minimize the number of people that cannot access the service. As such, an IMS with a large user–base would be preferable over one with a smaller user–base.

1.4 Purpose

The purpose of this paper is to find, based on current literature, a method for evaluating IMSs. The perspective of the evaluation will be that of the SP. Also within the scope of this thesis is the evaluation of a set of IMSs, chosen by the company at which the report is conducted, using said method.

Following the use of the review method there will be an implementation of a PASS that uses some of the authentication techniques deemed good enough. This service is what was desired by the company.

1.4.1 Research questions

Method/Technique

- How do you measure the quality, and certainty of identities, of an IMS from a service provider's perspective? This will be answered in subsubsection 5.2.2
- What characteristics are important for a good IMS? This will be answered in subsubsection 5.2.1

Implementation

- Can the identity concepts of different IMSs be used together in one architecture to identify users? This issue is explored in subsection 4.1 and resolved in subsubsection 5.1.1.
- Does the implementation increase or decrease the quality attributes related to security? This issue is explored in subsection 4.2 and resolved in subsubsection 5.1.2 .
- How do you connect the user to a desired identity? This issue is explored in subsection 4.1 and resolved in subsubsection 5.1.1.

1.5 Limitations

The thesis will focus on IMSs used internationally and specifically within the Nordic countries. The main purpose of this limitation is that the selected areas are the current markets of the company. Also, the chosen bounds were selected because the Nordic countries have come a long way in the implementation of e–invoicing [8]. However there are no indications that the solutions found in this thesis could not be applied to other areas.

It would not be possible to implement all security solutions in the architecture in the time frame of this thesis. Therefore the outcome will be limited to include four implementations.

2 Method

This section will describe how the work was performed. First the work process for the development of the evaluation method and the PASS will be described. Following that an application that will be used for validating the method will be described.

The evaluation method will be based on a literature study of the up–to–date research in the areas of user authentication and identity management. The goal is to create an evaluation method that focuses on the service provider's needs. Special focus will be given to commonalities between different current studies, since such could indicate important aspects of the evaluation.

The intended format of the evaluation method is a number of scales over various aspects relating to the relevant topics. The idea is that a software architect, or other person responsible for software design, should be able to grade the method's scales and in return be given information about what IMSs that satisfy the demands of the software to be implemented.

The implementation of the PASS is straightforward; however the software requirements for such a service are not well established. In the theory section the design problems of PASS will be described and in the result section proposals for solutions for the problems will be presented.

2.1 PASS design and evaluation

The implementation of the PASS will be done in Java EE. Java was chosen because it is a well–established language and platform for web development with a wide collection of libraries. Java is also the preferred language of the company. The idea is that a PASS should be a component that can be attached to different systems and work as that system's security provider.

The utilizability of PASS will be evaluated by group–sessions at the company. A PASS, connected to the case–study app, will be demonstrated for developers and management at the company in two separate sessions. The PASS solution's success will be determined by the feedback received from these people.

2.2 Case Study

To test the evaluation method and PASS, a web application will be developed. The application that the company is planning to implement was chosen for this task. A proof–of–concept version of this web application (hereafter referred to as the application) was developed in order to test the developed evaluation method. The application will utilize a PASS, which will also be developed as part of the case study, for its user authentication.

In this application the documents are invoices that companies want to issue to consumers. Invoices are a good type of documents since they involve the concept of payment, which in general require a higher level of security. The basic idea of the application is to provide a consumer with access to a collection of invoices that has been issued to her. In the application she shall be able to interact with the invoices in a number of ways. Since there is to be no,

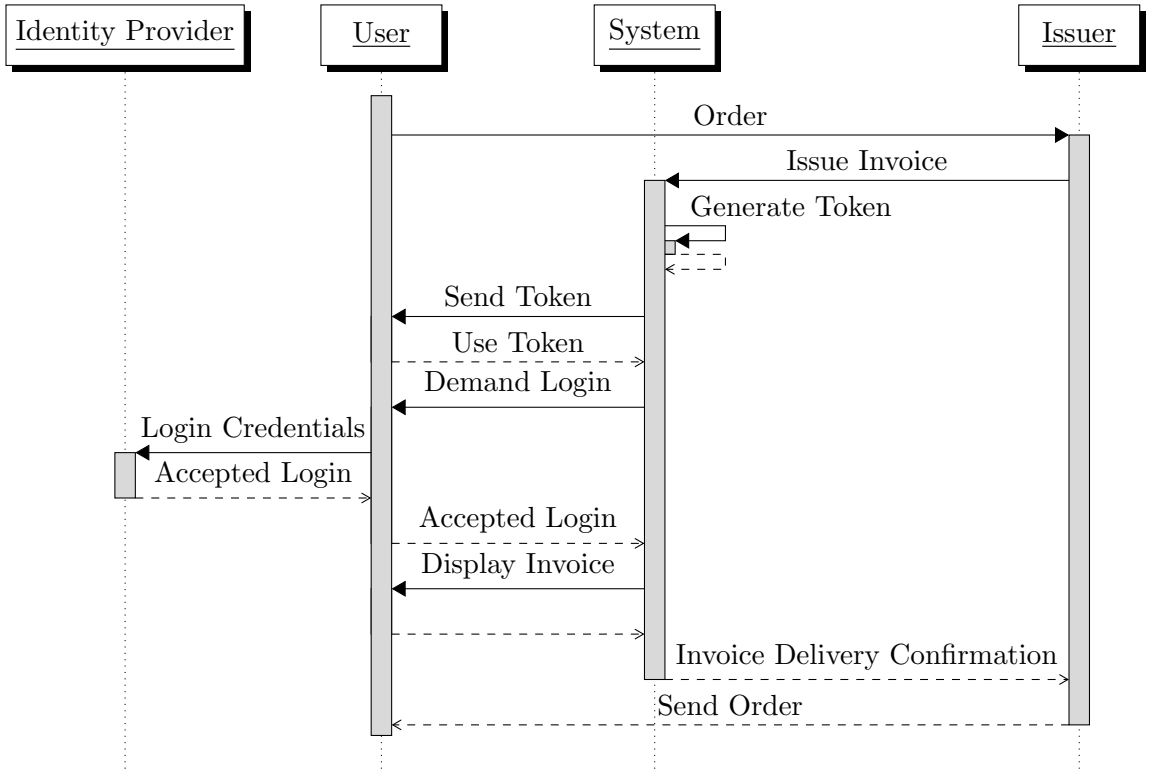

Figure 2: A complete transaction, from ordering to sending

explicit, registration step in the application there has to be some other mechanic that connects a user's account with the invoices.

To connect a user with her invoices (see Figure 2) a unique token will be generated and associated with an invoice. This token will then be sent to the recipient user via some communication channel the invoice issuer knows belong to the user, it can be email, telephone or some other way. The user will then browse to the application and provide the token given to her and log in to an IdP of her choice (assuming the application trusts that IdP's IMS). If the IdP accepts the user's input and validates the user, the invoice is tied to user's application account. If the user does not have an account one will be created with the IMS identity the user authenticated herself to previously. Any further invoice sent from the same issuer to the same recipient will also be tied to the user's account.

When an IMS account that has been logged in to the application before is authenticated by its IMS the user is logged in just as she would have been if the application had used a regular user–management system. When logged in the user can access the invoices previously tied to the application account.

2.2.1 Features

In this section a list of the application's features will be presented. Only the features concerning user login and management, and some basic actions used to prove the concept, will be listed here; the completed application will supply further features not related to this report.

Register a new user using a token

By entering the site using a token augmented link, received in an email or other communication channel, and authenticating herself the user shall be able to become registered.

Register a new user without token

By enter the site without clicking a link containing a token, either through a non–token link or by directly entering the site address, and then pressing the "Get Started" button shall the user be able to become a user of the system.

View an invoice

A user shall be able to access the information in an invoice that belongs to the account the said user has proven ownership of.

Pay an invoice using a third–party service

The user shall be able to pay invoices connected to her account using a third–party service, where the third–party vouches for the security of the monetary transition.

Pay invoice using a monetary system–local account

The user shall be able to pay invoices connected to her account using money she have previously transferred to a monetary account connected to her user account on this service provider.

Connect with another user

A user shall be able to create a connection with another user within the system, see *Add filter rule* and *Forward invoice* for further information. Example of the communication can be seen in Figure 3.

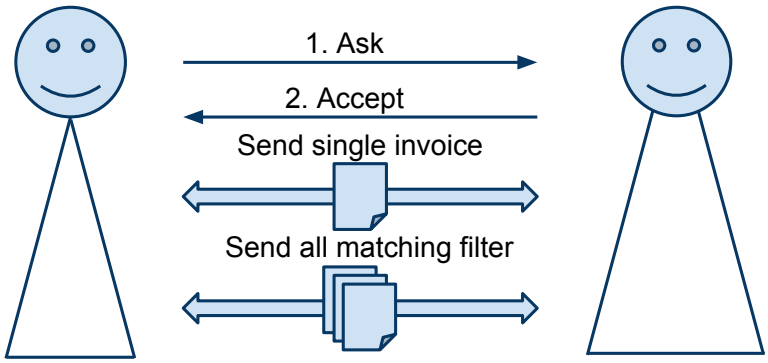

Figure 3: Two users connecting and sharing data

Add filter rule

A user with connections to other users (see *Connect with another user*) shall be able to add filters that automatically forward matching invoices to her connected users. The user that shares the invoice shall be able to define what the receiver will be able to do with the invoice. In particular, if the receiving user shall be allowed to view and pay the invoice.

Forward invoice

A user with connections to other users (see *Connect with another user*) shall be able to forward a single invoice to another user. The user that forwards the invoice shall be able to define what the receiver will be able to do with the invoice. In particular, if the receiving user shall be allowed to view and pay the invoice.

Add user information to the account

The user shall be able to add information about herself to the system; such as name, phone number, and address.

Deny invoice

In the event that the user receives an invoice she does not agree on (e.g. scam invoice or erroneous details) the invoice may be dismissed, removing it from the user's account.

3 Evaluation of the IMS field

In this section a number of research papers relating to user authentication and federated identities will be presented and analyzed. The focus will be on finding scales and tradeoffs that can be used to categorize IMSs.

3.1 Criteria by Myllyniemi

In [5] Myllyniemi describes three criteria that an SP could use to evaluate what type of IdP that would suit the needs of the SP. While some of the examples in the study are out-of-date, something that the author acknowledges would happen fast, the criteria is still relevant. The three criteria she presents are:

- **Context**

  Myllyniemi identifies three different contexts: federation of organizations, e–commerce application, and semantic web application. The list is ordered with the context requiring the highest trust and security first. Another consideration for this criterion is that users prefer to have fewer accounts. Using an IdP with many users increases the chance of the user having an account at the IdP.

- **Ease of deployment and use**

  Factors that influence the ease of deployment are: level of openness of the system, the resources the SP has for the deployment, the breadth of the system's feature set, and the technologies user in the environment where the identity management is to be introduced. Myllyniemi also points out that this criterion is of little value to the end-user. The user is interested in the usability of the system, something that the choice of IdP solution does not, according to the paper, affect much.

- **Scope of identity management system**

  In this context, scope refers to the width of features offered by the IdP. In particular it is a choice between only user authentication and also providing attributes about the users.

Myllyniemi emphasizes that these criteria does not form a definitive guide to selecting an identity management system. The criteria should be used as guidelines. She also points out that the evaluation of the scope of an IdP is only relevant for the present time; the scope of an IdP changes over time, something that can be seen today where OpenID, which was described by Myllyniemi as small–scale system, is a large and well-adopted system [1].

3.2 Method by Straub and Baier

In their paper Straub and Baier [9] presented a method, for evaluating PKI–enabled applications. PKI–based IMSs are just one of many IMS types available and this paper does not focus on third–party authentication. However it still concerns user authentication and as

[1] http://openid.net/2009/01/15/momentum/

such it seems reasonable that it can be adapted to third–party authentication. The method consists of fifteen attributes in three categories: *Deployment Issues*, *Ergonomics* and *Security Features*. Each attribute is evaluated by a few concrete questions, used to guide the evaluation process. The categories and attributes are as follows:

- Deployment Issues
  - Obtaining Information and the Software
  - Technical Requirements
  - Installation and Configuration
  - Technical Support
  - Training
- Ergonomics
  - Menus and Dialogues
  - Transparency of the Security Features
  - Warning and Error Messages
  - Help System
  - Reaction in Potentially Threatening Situations
- Security
  - Algorithms and Parameters
  - Secret Key Handling
  - Certificate Management
  - Status Information
  - Advanced Functionality

This evaluation, and these attributes, is not designed to evaluate IMSs, however some of these attributes may still be relevant in such an evaluation whilst others may not be. For example the "Training" property is not interesting in IMSs evaluation, as the end users of the system have a relationship with the IdP prior to using the system. Meanwhile the attributes in the "Security" are of interest as security is a very important aspect in the system.

3.3 Kylau, Thomas, Menzel and Meinel

In their paper Kylau, Thomas, Menzel and Meinel [10] list several aspects of risk and trust between IdPs and SPs in a federated identity context. Here follows a summary of these aspects:

- The service provider has to trust the identity provider to follow the agreed upon rules regarding user-registration, authentication, identity mapping and non-disclosure of usage statistics.

- In a single system with multiple service providers an SP has to trust the identity provider to vouch only for those external SPs that are federated with the IdP and are trusted to follow the agreed upon rules.
- In a system of federated identity providers the service provider has to trust its IdP to authorize only those federated IdPs for authorizing delegated access that follow the agreed upon rules.

The first item in this list is the most important for this analysis; the IMS is required to deliver what the service requests. The other two items relate to systems of differing structures; however neither is relevant in a PASS system.

3.4 Trust Requirements in Identity Management

Jøsang et. al. [11] highlights trust problems in current identity management solutions. They describe trust requirements between all pairs of service provider, user, and IMS. They compile a list of trust requirements from four types of identity management solutions: *Isolated Identity Management*, *Federated Identity Management*, *Centralized Identity Management*, and *Personal Authentication Management Architecture*. The focus of this report makes the first and the forth solution types less interesting. Here follows a list of the trust requirements described by [11] that are relevant for the service provider–authentication service relationship and of interest to the second or third identity management solution.

1. **Service access by assertions between service providers on behalf of users will only take place when legitimately requested by the client**
2. **The service provider has implemented satisfactory user registration procedures and authentication mechanisms (from the client's perspective)**

   This requirement was defined for isolated identity management but is also valid for the federated scenario because all IdPs in the federation needs to know that the other IdPs provide satisfactory user registration and authentication.
3. **The credentials provider has implemented adequate procedures for registering users and for issuing credentials**

Special attention has been given to authentication mechanisms in [6]. In the paper the authors describe a mathematical model for calculating, quantifiable, trust levels for the authentication mechanisms. The metric used in the model is the probability that an attacker can crack the authentication method.

Elahi et al. describe problems with trust and IdPs in [12]. In particular the problem that is highlighted is that the IdP gets full control of the user's account. If a malicious IdP wants to access a user's account it can assert (without the user's consent) that it is the user and the SP will not be able to tell the difference from a normal authentication assertion. Using a goal-oriented model they demonstrate what effects the misuse of the user's trust has.

3.5 Summary of Presented Papers

A common theme in the identity management field is trust establishment and trust management. The same problems exists in systems where the user authentication is not delegated to another service however on those systems the trust is implied to a greater extent; the security component is assumed (trusted) to authenticate users and then provide correct data to the rest of the system. The problem with delegating the user authentication is that the security component is a black box, so the provider of the black box has to be trusted in order to establish trust in the component.

The service provider has to trust the technology used by the IMS; both for authentication and assertion transport. Furthermore, the service provider has to trust that the IMS provide valid information. An extra dimension of the last past is that the information the IMS believes is correct might not be so since the user can have lied about the information she provided to the IMS. So there also needs to be trust between the user and IMS.

4 Plug–in Authentication Security Service

A PASS is supposed to work as an authentication software component and should be independent from the other components in the software. What separates PASS from most other authentication components is that it delegates the actual authentication to third–party off–site services (IMSs) and also that it utilizes several different such services. There are a number of problems associated with designing this kind of system. In this section the problems will be described.

4.1 User Authentication

An application that performs its own user authentication stores information about its users, along with their authentication information, locally. When a user authentication takes place the information provided by the user is validated against the locally stored information. When utilizing an IMS such a validation is not possible since the application does not store any authentication information about its users. Instead the application receives an assertion from the IMS that attests that the user is authenticated. It is common that said assertion contains a persistent identifier that can be used to uniquely identify a user of the application, without necessarily knowing the users true identity.

The mapping of persistent identifiers to application users is nothing new; however in a plug–in based architecture this is not as trivial. It is not reasonable to assume that a persistent identifier is globally unique; a common identifier is an email address (which is globally unique) but some IMSs that focus on anonymity will not give out their users' email addresses. Instead they will use some hash-value that is guaranteed to be unique within the IMS's domain.

4.2 Security Level

When determining what authentication solution to use in an application that handles its own authentication the solution can be tailor-fitted for the application. The authentication

solution can guarantee that the required level of trust and level of security is met since the solution is designed for this particular application. The same does not apply for applications that depend on authentication from IMSs.

Different IMSs provide different level of trust and security. It is not reasonable to expect that you should be able to tailor-fit the authentication solution for your needs. When selecting an IMS to use, this has to be considered. The natural solution would be to select the IMS that provides a level of trust and security as close to the required level as possible while still being above the required level. However for a plug–in architecture that tries to maximize the number of users that can access the application this may not be enough. The selected IMS might have a small user base which means that the number of users that can access the application through the IMS is low. There could be an IMS with a large user base that almost satisfy the application's requirements, but it cannot be used since all requirements are not met.

4.3 Changing Authentication

Authentication credentials on applications that handle their own authentication are tied to the application and will therefore be around as long as the application is around (assuming no data loss due to accidents).

When using IMSs this is not guaranteed; a user's authentication could be tied to an organization (such as an academic institution) and as such be invalidated if the user quits that organization. If this were to happen the IMS would no longer accept the user's authentication attempts and due to this no authentication assertion would be sent to the application. The user's account on the application would still be active with all its data intact but the user would be unable to access it since the account only allow access to users who can prove ownership of the IMS account associated with the application account.

This is a serious problem since there is no intuitive way for the user to regain access to her account. For the user to regain access she would have to somehow prove that she is the rightful owner of the account. Possible solutions for this could be for the user to contact the IMS and request some sort of reactivation or the user could contact the applications owner and try to convince them that she owns the account. Both solutions require that the IMS and the SP respectively would need to offer customer support.

5 Result

In this section the results from the literature study and the demo application will be presented.

5.1 Plug–in Authentication Security Service

In this section we describe the solution to the problems described in section 4.

5.1.1 User Authentication

The problem stated concerned how to uniquely identify a user based on the authentication assertion sent by the users IdP. Since it is not possible to guarantee that the provided identifier is globally unique (a problematic scenario is described in Figure 4) some additional information has to be tied to the user. A good candidate is a unique identifier for the IMS used by the user. Since a user's identifier must be unique within an IMS's domain the identifier becomes globally unique if we can uniquely determine the IMS; which we definitely can since the IMSs supported by the PASS are pre–configured, and as such it is possible to assign a unique ID to each IMS. The unique identifier used in PASS will thus be the IMS's user identifier concatenated with the unique identity of the IMS.

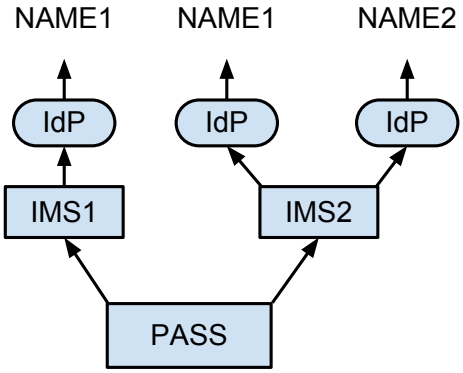

Figure 4: NAME1 in IMS1 and NAME1 in IMS2 would be in conflict if PASS handled the names unaltred.

This problem related to the two registration–features described in subsubsection 2.2.1. Since there is no locally stored user authentication information in a PASS–system the registration consists of receiving an authentication assertion for a user that does not have a PASS–account yet. In both registration cases the user will visit the PASS utilizing site and when trying to login will be offered selection of IMS that the PASS supports. The user will be redirected to the selected IMS. When the user is authenticated an authentication assertion will be sent to the service provider. The PASS will determine if there are any previous references to the provided IMS account id, if there are none a new user will be added to the PASS. A user in this context is an object that refers to the IMS account and, as the user utilizes the service, will also reference various service resources (e.g. invoices will be referenced in the case–study).

In the token augmented case, the invoice will also be connected to the newly created user object.

On consecutive authentication assertions, when the PASS has a previous reference to the IMS account, the above described scenario will work as a login (instead of a registration).

5.1.2 Multiple Security Levels

In subsection 4.2 the problem of delimiting the user–base that has access to the application was described. To mitigate this problem a multi–level security–architecture was adopted. In which the security level was determined by the level of trust the PASS has in the IMS and the security used when authenticating the user. In other words, a user logged in with a trusted IMS gains access to more security demanding features than a user who log in with a, not as highly trusted, IMS.

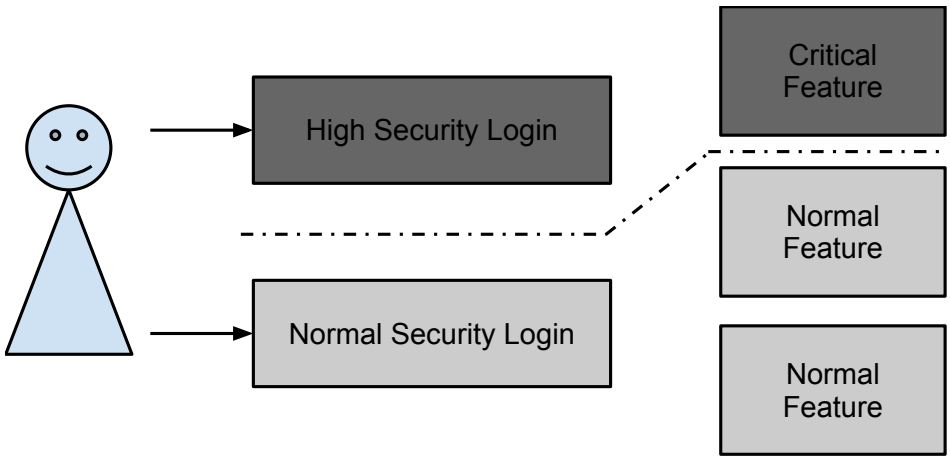

Figure 5: Limiting access depending on security level

Some IMSs support more than one authentication technique. In particular, some IMSs support both low–security and high–security authentication. E.g. VeriSign's Personal Identity Portal [2] offers password–based authentication with an optional possibility of adding a second factor, namely one–time password, to the authentication process. Some IMSs also allow their IdPs to determine their own security level (an example of this is Kalmar2, whose SAML profile does not state what authentication method is required). This means that, even if the capabilities of the IMS are known beforehand, the PASS cannot determine the security level of an identity before its IMS sends an authentication assertion. Upon receiving a login authentication from an IMS the application will try to determine what security constraints were satisfied during the login process (many IMSs provide information about the authentication technique used in their authentication response).

The justification for the multi–level security architecture is that even a high security application often has features that does not require as high security. An example of this is a bank that enables issuing of bank statements and transfer of funds between the user's own accounts

[2] https://pip.verisignlabs.com/learnmore.do

through an app with single–factor authentication while the, full–featured, web–application requires multi–factor authentication. Another example is the case–study, where a user with a lower security IMS account can view invoice, pay using third–party services, etc. Meanwhile a user with a higher security IMS account can do all the things the lower security user can do and she can also pay using a monetary system–local account.

With a multi security level system the user using lower security IMSs can still access some features and as such still take part of the application. This solution raises some new problems; what if a user who is accessing the application at the lower level wants to utilize features only available to higher level security users?

To address the new problems of the multi security level solution, multiple IMS–identities per user were introduced. By connecting a higher security identity to the PASS account the user can gain access to more features. A user that has previously tied an IMS–identity to her PASS account can tie a second IMS–identity to the PASS account. This is done by proving the ownership of the new account, in other words; login to the account, while logged in with the old account. By doing so the user has two access points to her account. Both are valid but retain their individual security level, so the old IMS–identity still has a lower security level.

Retaining the lower security identity is not necessary for enabling the heighten security, however it mitigates the problem discussed in subsection 4.3; by adding a new IMS account to the PASS account the old account that will become unavailable is no longer needed.

5.1.3 New features for supporting multiple security levels

Based on the solutions for the PASS a number of features, needed for the PASS to work, can be derived. In this section these features are described.

Add IMS account

A user must be able to add another authentication option to her account. This allows the user to migrate from one IMS to another without having to worry about not being able to access the application. An IMS account is connected by, while logged in, attesting the ownership of a non–connected IMS account.

A sub–feature of this feature, that mostly provides convenience, is that whenever a user tries to access a feature that requires higher security than the currently authenticated IMS identity a prompt for adding a new IMS identity is displayed. The flow of this feature can be seen in Figure 6. Any action taken will be subject to a security check, and if the current level is found to low, strengthening is required. The user will be prompted to authenticate a new IMS identity. Preferably the new account should provide stronger authentication, however there is no way for the application to enforce this. If the newly authenticated IMS identity provides authentication that satisfy the features security requirements it will be tied to the PASS account and the user will be given access to the feature.

Remove IMS account

The user shall be able to remove an IMS identity from her account. By removing the identity it can no longer be used to access the user account it was previously connected to. A constraint on this feature is that when only one IMS account remains it cannot be removed since then the user would not be able to log in. Furthermore, if the account to be removed is the last account

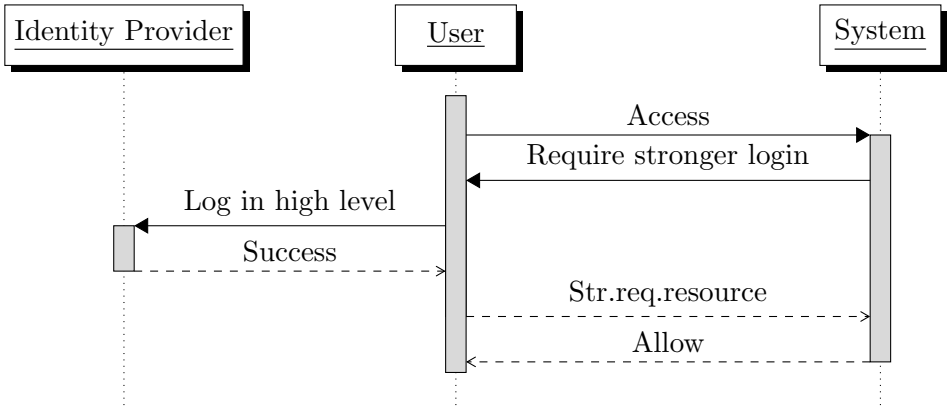

Figure 6: Strengthening the login to make use of advanced features

on its security level the information, attached to the user account, that requires that security level should also be removed. This could cause problems depending on what information it is. What should be done about the removal of the information is an application–specific business decision.

Merge accounts

It shall be possible to join two user accounts into one. This occurs when an authenticated user attests her ownership of an IMS identity connected to a different user account.

5.1.4 PASS implementation

The general structure of the PASS can be seen in Figure 7. Each IMS that is added needs a handler object that extends the abstract class *IMS*. Common for all IMSs is that an authentication will be requested and that some authentication response will be received. These two scenarios are handled by the methods *sendAuthnRequest()* and *getAuthnResponse()*.

The *IMSServlet* sent to IMSs. When a user selects an IMS that she wants to use a request is sent to the *IMSServlet*. The servlet determines which IMS was selected and calls *sendAuthnRequest()* on that IMS. What happens next is IMS specific but at some point the IMS will want to send an authentication assertion. This assertion is received by the *IMSServlet* that delegates the interpretation of the response to the IMS handler. The IMS handler successfully validates the response it creates an *AccountInfo* object, which is an IMS independent representation of an IMS account.

After an *AccountInfo* has been created the *UserService* can be used to retrieve a system–local user based on the *AccountInfo* object, assuming such a user exists. If no user exists for an *AccountInfo* object a new user is created (registration) and connected with the *AccountInfo*.

The sequence diagram in Figure 8 describes the login scenario of the PASS. When the user first enters the website she must login to access the features. The website queries the PASS for available IMSs and shows a list of available IMSs to the used (see Figure 9).

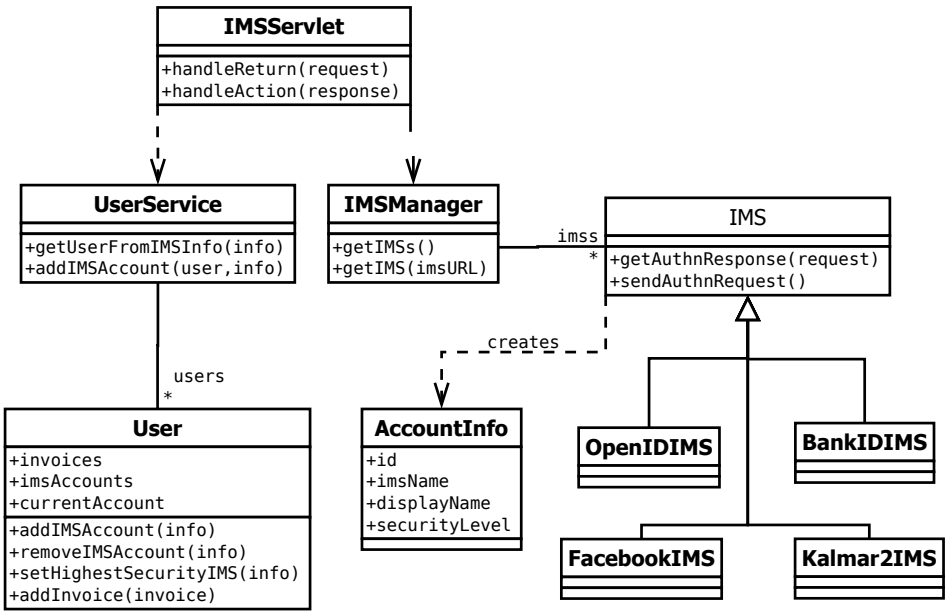

Figure 7: PASS Class Diagram

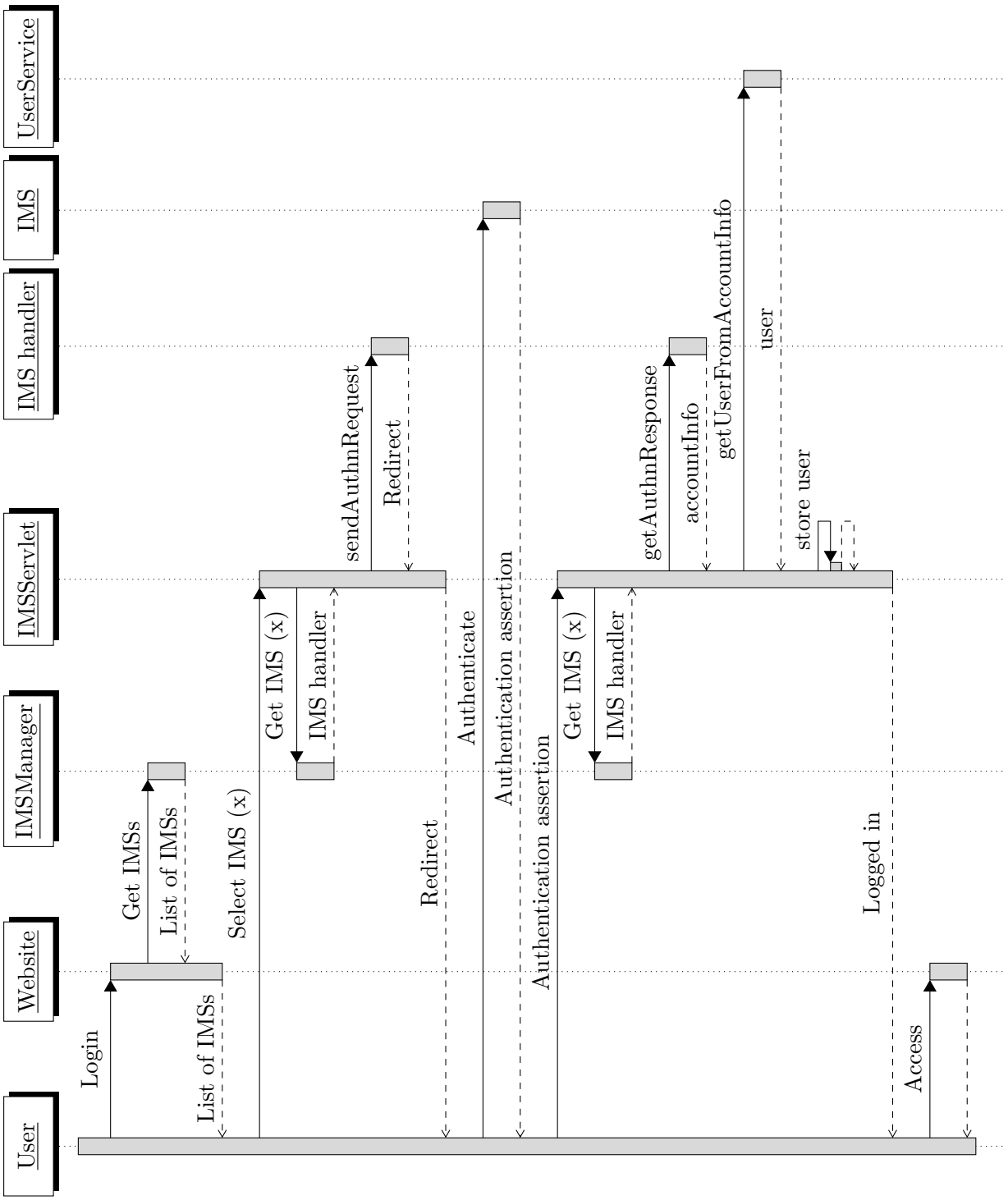

Figure 8: A complete transaction, from ordering to sending

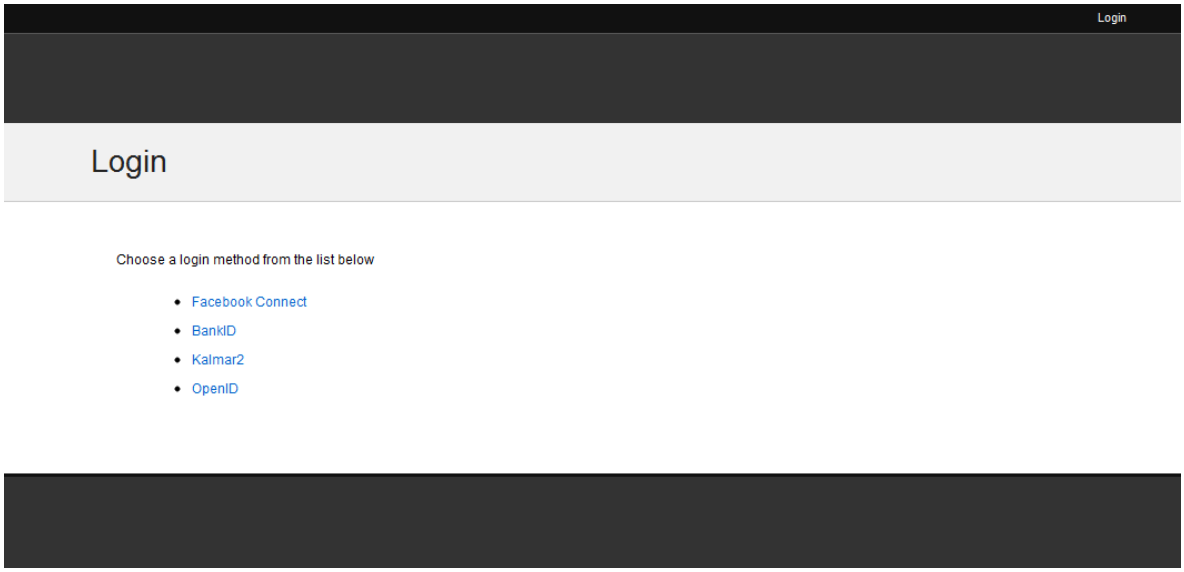

Figure 9: IMS selection in case study app

After the user select an IMS that IMS's handler will be called to initiate an authentication request. What this entails is IMS specific but often it will involve a redirect to the IMS's domain (see Figure 10). After authenticating at the IMS's domain the user is sent back to the website where a call to *getAuthnResponse* is made. The response is interpreted and a user object is created or retrieved from the database. At this point the user is logged in and can access the website's contents (see Figure 11). Notice at the top of Figure 11 that the IMS used by the user is displayed along with the IMS identifier the user is logged in with. The screenshot is from the case study application and displays the invoice listing.

Whenever a user tries to access a feature that requires that a user is logged in a check is made to see if there is a user object saved and that the user object has the appropriate security level. This behavior is much like a conventional user management service. A difference however is that if, as described earlier, a user lack the appropriate security level she will be given the chance to provide a more secure IMS account instead of being denied access (of course if the user chooses to not add an account the action will be denied). This feature behaves like the registration with the difference that when the user is returned the authentication assertion is used to connect an additional IMS account to the current user instead of logging in a new user.

If the newly authenticated IMS account already belongs to a user in the PASS the two accounts should ideally be merged, since the user can prove ownership of both accounts it should be safe to assume that it is the same person.

5.1.5 Demonstration feedback

The feedback received from both developer as well as management was positive. The implementation satisfied all required user stories and the developers were not able to find any flaws in the implementation. Management raised some concern about the limited selection of IMSs that enables access to higher security features; however this is not a problem with PASS.

Figure 10: Authentication page at an OpenID IdP

5.2 Authentication Service Evaluation

In this section the evaluation method will be presented and demonstrated. The result of the literature study will be presented first. A list of evaluation scales, derived from aspects deemed important by the literature, will be presented. After the scales have been presented some reasoning about modeling the identity management domain for use with the method will be discussed, and finally the evaluation method itself will be presented.

5.2.1 Evaluation Scales

At the core of the evaluation method is a number of scales. The user of the method will determine what value on each scale her service requires and from this she will find what IMSs satisfy her needs.

Security

There exists a lot of literature on the importance and evaluation of security in a service (e.g. [9] and [13]). Security considerations in the context of an IMS evaluation are slightly different. Some SP features require higher security authentication, such as multi–factor authentication. However, if an IMS utilizes, for example, multi–factor authentication, it is not the service provider's responsibility to guarantee that the authentication works correctly and well.

The best a service provider can do is look at the authentication assertion sent by the authentication service. Most authentication services provide some mechanism for determining what authentication technique was used in the user authentication. If the assertion says that the

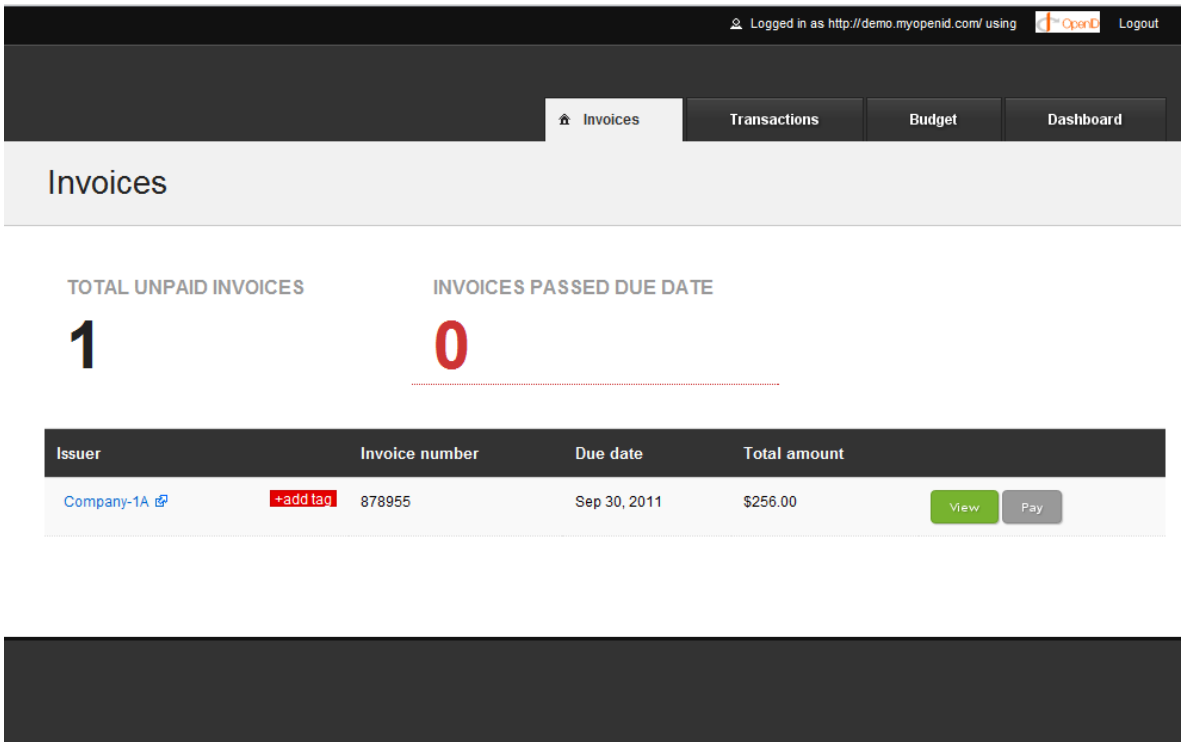

Figure 11: Logged in user in case study app

authentication was performed with higher security the service provider has to trust it (more on this in Table 5.2.1).

What is interesting for a user of the evaluation method is what type of authentication techniques the IMS uses. The internals of the authentication solution is not considered since the possibility to evaluate them in a third–party identity management service may be limited. The question the evaluation method user should ask is: "is the IMS using single–factor or multi–factor authentication?" Single–factor means low security and multi–factor means high in this method. If a more fine–grained evaluation of the authentication trust is needed [6] may be used. However this is not required by the evaluation method.

Ease of deployment

According to Myllyniemi [5] the end user has little to no interest in the time the developers spend on creating the identity management for their service. However this is very important to the developers, and will thus be considered in the evaluation. Straub has one category concerning deployment, from the view of the IdP developer, where she poses questions to the COTS products owner. However some of these questions bear asking as a service provider, this time directed to an IMS: "Are there FAQ or manual pages on the Internet?" and "How can the user contact the support team?" I.e. if we encounter difficulties in development, is there help to be found? Also "How complicated is it to obtain the product?", "Does the installation require additional programs?" i.e. is there some obstacle in the way of developing support for this IMS?

The main aspects to look at in the ease of deployment (EoD) scale are: what is required to

Value	Manual Registration	Available Software
1	yes	no
2	yes	yes
3	no	no
4	no	yes

Table 1: Ease of Deployment

get started with the IMS and the availability of software that provides an interface toward the IMS. Some IMSs allow anyone to request authentication for a user while other IMSs require that the SP register with the IMS so that the IMS can control how and who requests authentication assertions. Registration is used to describe any form of action that establishes a relationship between the IMS and the SP.

Some registrations are more tedious than others and as such decrease the ease of deployment more. There are registrations that are automatic, meaning that no manual work has to be performed by the IMS for the SP to be able to utilize the IMS. An example of an automatic registration is *Facebook Connect*. Others require some action from the IMS's side, which means that the SP loses control of when the service can be deployed. The IMS could be fast in approving the SP but it could also take a long time. It is the element of not knowing when the IMS will approve the request that makes such registrations reduce ease of deployment.

When looking at EoD in this context it is only the second version of registration that has any noticeable effect on the EoD scale and because of this, when registration is mentioned in this text it refers to the second version, the one that require some action from the IMS's side.

Using preexisting software will, in most cases, require less development time than developing a new solution. In other words the time–to–market is decreased if preexisting software is used and it follows from this that ease of deployment is increased. Well–established and well documented software will of course further improve ease of deployment.

The ease of deployment scale is a four level–scale where each level represents a combination of the two aspects discussed above (see Table 1). The IMSs that are the hardest to deploy are the ones that require registration and have no, or a very limited amount, of available software. In the other end of the scale are authentication services that require no registration and have available software that can be used to ease deployment.

In the middle of the scale are the two remaining combinations: requires registration and has available software and vice versa. It is not completely obvious which of the two combinations should be level two and level three respectively.

However developing some additional software is still something the SP can control and as such can be planned. Which will if not decrease the time–to–market at least allow the project manager to plan the release better. Therefore the ones that require registration and have available software will be level two on the scale while the other combination will be level three.

For the user of the evaluation method availability of software means shorter time–to–market and could also mean less technical training for the developer, but at the same time it limits the range of IMSs to choose from. The evaluation user should assess if there is a need for fast deployment and if there is a lack of technical knowledge in the development team. If there

is not, it is a good idea to not require available software (in order to increase the number of possible IMSs).

Registration is in general mostly interesting for the IMS to establish trust in the SP; by knowing who the IMS provides user information to can the IMS reduce the risk of the users' information being misused by malicious SPs. For some SPs establishing trust is more important than being quick to market and for those SPs it might be worth sacrificing ease of deployment. The question that should be considered here is: is it more important to get a quick release or strengthen the trust relationship between the SP and the IMS.

Certainty of Identity

Certainty of identity is defined as whether or not the IMS can connect their digital identity to a physical person. For the SP the certainty lies in the trust it has for the information received from the IMS. This trust is achieved in two steps. First the IMS has to give out the information about the user, acquired in an extensive registration. Secondly the information has to reach the SP untampered. Both of these points are based on trust; trust towards the IMS to send the correct information, and trust towards the protocol used in the exchange.

Trust between parties is an important issue of IMSs [11]. This seems very natural since any authentication assertion, or other information, provided by a source that you do not trust cannot be relied upon to be true. That is, even if everyone has an account on a theoretical IMS with the best security available but with untrustworthy data, the IMS may still be unusable in identity critical services since the data may still be wrong. Likewise, if the IMS sends the correct data to the SP, but without any encryption, the SP cannot be certain it has not been the target of a man–in–the–middle attack (see Figure 12).

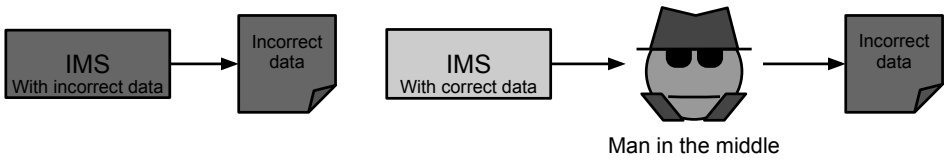

Figure 12: Two possible ways to receive incorrect data

Services that do not require a high certainty of identity however does not require as high trust in the authentication assertion. For these services it is often enough to be able to uniquely identify a user. Who that user is has little importance, as long as she can be uniquely identified in the service's context.

This scale focuses only on the trust between the service provider and the IMS. It is important to observe this in the previous reasoning about using low–trust IMSs; just because it is acceptable for the service provider to trust the IMS enough to use it does not mean that the user necessarily should. The IMS could for example violate the user's trust and log in as the user in services the user uses. The central thing to notice here is that while it is not good for the IMS to exploit its users, this problem is not directly relevant for the service provider. As a side–note, a misbehaving IMS will probably not get many users and since the evaluation method considers the user adoption of the IMS a low user–IMS trust will be reflected in the evaluation method.

As described in the theory section, [11] provides three trust requirements that are of interest

for the service provider–IMS relationship, two of which focuses on the need for adequate user registration. In the evaluation method adequate registration is defined as registration that can guarantee a connection to a physical person.

Jøsang's third, relevant, trust requirement states that an IMS should not misuse its power by issuing assertions when the legitimate user has not requested them, the same is stated by [10] and also highlighted as a problem in [12]. The recommended way to achieve this is by having both parties agree to a common set of policies and possible by establishing a contract that defines the obligations and responsibilities of the parties [11]. On the certainty–scale only IMSs with some sort of mandatory contract or agreement that guarantees certainty of identity will be graded at the higher level.

The certainty–scale is a binary scale where the higher value represents IMSs that fulfills the requirements described in this section. The lower value shall be given to any IMS that does not meet the requirements.

The user of the evaluation method should consider if the service needs to be able to link its users' accounts to the actual physical user using the service. If so the higher level of certainty is needed, if not need the lower level is sufficient.

User Adoption

An IMS that is perfect in every aspect, except that it has no users, would not provide any value to the service provider. It is however hard to define scale–levels for number of users. A scale with two levels could be used: small user–base and large user–base. The problem here is determining where to draw the line between the levels. In the end it would probably be an arbitrary number and the scale would not provide much value. The same reasoning applies if more levels were to be introduced.

User adoption will instead be used in a different way; instead of a scale used together with the other scales, user adoption will be used to sort the IMSs that were found appropriate for the service provider. The list would be sorted with the IMS with the highest user–base at the top. The user of the evaluation method would then have a recommended order for looking at the IMSs that fulfills the service's needs.

Usability

When looking into evaluation of IMSs an often reoccurring aspect is Usability (e.g. [9] and [14]); the metric that measures how easy it is for an end–user to utilize the service. However the usability of an IMS is not directly relevant for the selection of an IMS; if a user has an account on the IMS it is reasonable to assume that she is satisfied with the user experience. As the usability of an IMS only affects the end-user's experience and her willingness to make use of this IMS, this aspect is already accounted for in the section user adoption (5.2.1).

5.2.2 The ETA method

The Evaluating Third-party Authentication (ETA) method is used to determine the quality of IMSs. The conclusion from the literature study is that there are three important aspects involved when measuring the quality of IMSs. That is, the combination of security, ease of deployment, and certainty of identity defines the quality of an IMS. It is worth noting that with this definition services that does not require, for instance, high security will have IMSs

Scale	**High**		**Low**	
Security	The IMS uses multi–factor authentication		The IMS uses single–factor authentication	
Certainty of Identity	IMS can guarantee a connection to a physical person. A contract or agreement that guarantees Certainty of identity must be established		IMS does not need to guarantee a connection to a physical person. No contracts are required.	
Scale	1	2	3	4
Ease of Deployment	Requires manual registration. Software that interfaces toward the IMS is not available	Requires manual registration . Software that interface toward the IMS is available	No registration required. Software that interface toward the IMS is not available	No registration required. Software that interface toward the IMS is available

Table 2: Evaluation rubric

with low quality in the result returned by the method. However, this is not a problem since all services does not need the highest quality to operate.

The evaluation scales are summaries in the rubric seen in Table 2. The rubric is not as straightforward as a conventional rubric since the scales are not all graded in the same way, however it is still a good way to represent the evaluation.

There are three parts to the input to this method: the IMSs and their analyzed attributes, the user adoption of the IMSs, and the requirements placed on the IMSs by the user of the evaluation method. The first part (the service analysis) is constructed by analyzing the services. It should be updated when services get removed, start up or are heavily changed. The second part (the user adoption) needs to be updated continuously as the amount of users of an authentication service may shift rapidly. The final part (the requirements) is defined once by the evaluation method user, based on the needs the evaluator sees for the service application.

The evaluation method user should use the rubric defined in Table 2 to help her determine what evaluation scale values her service requires. If the requirements were to change the user would have to reevaluate the values set on the evaluation method's scales.

Firstly the requirements are used as a filter to eliminate the authentication services that does not fit the required profile. The remaining services meet or exceed the requirements, according to the scales in subsubsection 5.2.1.

The output of this method is a list of authentication services matching the requirements of the user. The list is sorted with the most used service at the top.

5.2.3 Authentication Service Categories

A prerequisite for the user to be able to use the evaluation method is that it is known what scale–values each IMS has. This section will present a list of what values a selection of well–established IMSs have today. The IMSs that will be presented are *OpenID*, *Kalmar2*, the Swedish *BankID*, and *Facebook Connect*. A description of these IMSs is given in Appendix A.

This sort of list will be subject to early aging. It is therefore important to reevaluate the available IMSs from time to time to make sure they are still relevant. This should be considered when applying this evaluation method. If too much time has passed since the writing of this report a reevaluation is recommended.

In Table 3 the values for the well–established IMSs are presented. The evaluation and reasoning that derived these values can be viewed in Appendix B.

IMS	Security	Ease of Deployment	Trust	Users
Facebook Connect	low	4	low	750 million
OpenID	low	4	low	N/A
Kalmar2	low	2	low	N/A
BankID	high	2	high	3 million

Table 3: Evaluation values for some well–established IMSs

5.3 Case Study

In this section the evaluation method will be used on the test application first described in subsection 2.2. To do this all the previously defined features will be analyzed to determine what level of security that is required. After that, a reasoning about the need for ease of deployment and certainty of identity is presented, from an application–wide perspective. When this is done all input for the evaluation method is defined. What is left is to filter the IMSs based on the requirements.

5.3.1 Security

To apply the evaluation method to the test application the security scale–value of the application has to be defined. To determine the value, a security requirement analysis will be performed for each feature. Each feature will be evaluated based on the criteria defined by the evaluation method. The conclusion of the analysis for each feature is highlighted in boldface in the text below. When all features have been evaluated a summary of all features' security level will be presented.

Register a new user without token

Registration has to be available to not–logged–in users, and as part of the registration process users are required to log in. Since the user has no account, **no security constraint can be required**.

Register a new user using a token

Registration have to be available to not–logged–in users, and as part of the registration process users are required to log in. Since the user has no account, **no security constraint can be required**.

View an invoice

Looking at the security of paper invoices we find that people in general are content with receiving invoices in the mail, a communication channel vulnerable to man–in–the–middle attacks and other attacks. It seems reasonable that a not–so–high security level should be

sufficient for this feature. **It is enough that some user, by authenticating herself at an IMS, has claimed ownership of the account the invoice is tied to**.

Pay an invoice using a third–party service

Paying invoices in this manner may seem as a feature that requires a heightened security, however this is simply a convenience feature shortcutting having the invoice open and copying the data to the third–party payment service. A hacker that takes control of a user account cannot spend any of the real user's money since the hacker also needs to have access to the third–party payment service, and if she does have access to that account there will be no need to use the account of this system. Since the required heightened security for dealing with monetary transactions is handled by the third–party payment service **any login with invoice viewing privileges may use this feature**.

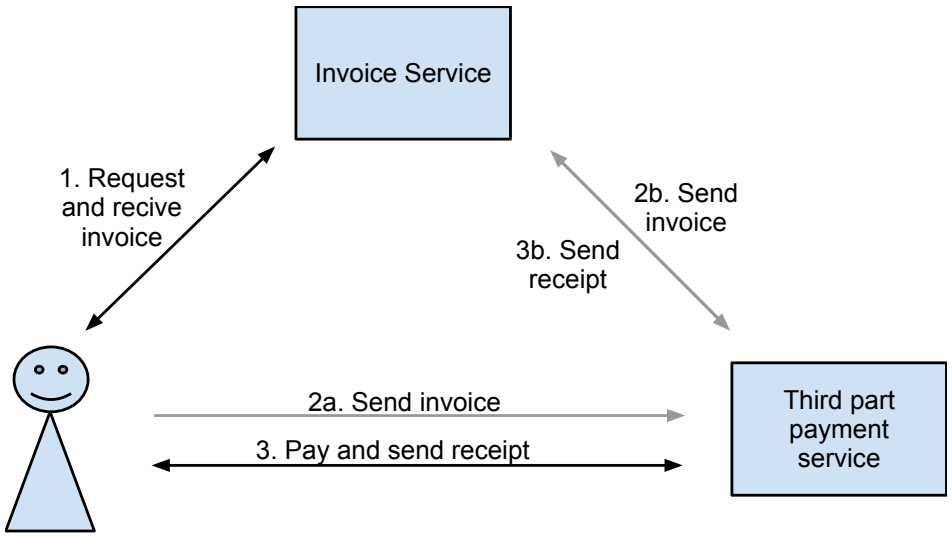

Figure 13: Paying invoice in two ways

The figure Figure 13 describes a payment both by manually handing the invoice to the third party (path A), and using this feature (path B). The important thing to note is that the critical point is the same in both solutions, namely the payment.

- 1. The user receives the invoice from the system.
- 2a. User manually sends the invoice to third party.
- 2b. The service sends the invoice to the third party on request by the user.
- 3. The user pays the invoice with her account there, and receives a receipt
- 3b. The Service receives confirmation that the transaction is completed

Pay invoice using a monetary system–local account

It has been recommended that when dealing with financial transactions a higher security level should be used when authenticating users [15]. Furthermore, since the company controlling

the system has to keep accounts of the financial transactions and who was involved in them the system need to know that the physical person behind the IMS identity used with the user account is not an impostor. **These requirements on the security demand that a higher security level be required**.

Connect with another user

Sharing an invoice with another user in the system is comparable to having the other user next to the first while the invoice is displayed, or to printing the invoice and explicitly giving the other person it. **Access to this feature is granted while the user can view invoices**.

Add filter rule and Forward invoice

Same reasoning as in 5.3.1. Giving another user the pay permission while the sharing user does not have it is not a problem since when the invoice is received it is the receiver's authentication that guarantees security. In other words, a user may have permission from another user to pay her invoice but not permission from the system to spend the money on the users account. **Access to this feature is granted while the user can view invoices**.

Remove an email address from a user's account

No permanent damage can be done by this feature. Email addresses are used for establishing channels for invoice sending between invoice issuers and user accounts. As such removing an email address only means that the next time an invoice issuer issues an invoice the invoice will be sent to the email address only and not to the user account. The user can then reattach the email address to her user account. **It is enough that the user is authenticated in order to use this feature**.

Add user information to the account

Obviously the feature requires that we have access to a user, in other words it requires that the user have authenticated herself. However the required level of trust between the user and the system is not as obvious to determine. Regardless of the security and trust level the information provided by the user is going to be self–asserted; the system will not know if the given information actually belongs to the user. At the same time it is in the best interest of the user to provide information that belongs to her. **Based on the analysis we see that the level of security does not matter as long as the user is logged in**.

Deny invoice

Since a user can chose to ignore any invoice in her invoice list this feature is mostly a convenience function. As such it does not increase the demand for higher security. **As long as the user can view the invoice she should be able to deny it**.

Add IMS account

As this is an action to add something to a PASS account **the user must be logged in to access it**. This feature does not directly affect the user's sensitive data so any logged in user has a sufficient security level.

In the scenario where a user has a PASS account with two IMS identities, with different security levels, attached to it, a hacker could gain access to the user's higher security information by first gaining access to the lower level account and then attaching a new account with the same level as the previous higher security level account (as depicted in Figure 14). This is a security flaw and to remedy it a further constraint needs to be put on this feature:

A PASS account can only attach a higher security level account if the current authenticated IMS identity is the user's highest security level identity.

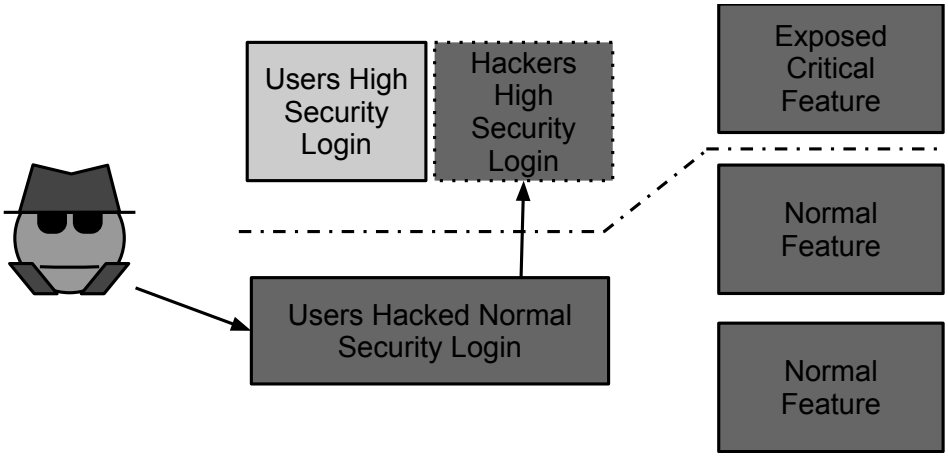

Figure 14: Malicious actor achieving high level authentication

Remove IMS account

There are no privacy concerns to consider here but instead it has to do with account security; a hacked IMS account could potentially result in the hacker denying the owner access to the account. At least if an authenticated user could remove accounts as she pleased. This can be avoided if the user is only allowed to remove accounts that she have authenticated at the time of removal. This way a hacker cannot hijack a multiple IMS identities account completely because the user can still use non–hacked IMS accounts to log in. However, if a user loses access to an authentication (e.g. forgot password or canceled account) it can't be removed. Given the alternative (the user losing access to her account) this is an acceptable disadvantage. **The required security level for this feature is the security level of the IMS account that shall be removed**.

Merge accounts

As long as the user is logged in no special security level is needed since the security will be provided by authenticating the two IMS identities. Imagine the scenario where a minimum security user account have been hacked and the hacked tries to gain access to an account with higher security (with the possibility to do more harm). The hacker will not be able to do so since she has to attest the ownership of the higher security account before gaining access the higher security features.

As discussed above, the security required for this feature is provided by the two accounts to be merged and because of this the feature only requires that a user is logged in.

Summary

In Table 4 a list of all features and their security level are displayed. As can be seen in the list, there is only one feature that require high security. Following the method this means that the application should require high security from the IMSs. This seems a bit to restricting. One solution would be to exclude the high–security demanding feature and then grade the

Feature	Security Level
Register a new user without token	–
Register a new user using a token	–
View an Invoice	low
Pay an invoice using a third–party service	low
Pay invoice using a monetary system–local account	high
Connect with another user	low
Add filter rule	low
Forward invoice	low
Add user information to the account	low
Deny invoice	low
Add IMS account	–
Remove authentication option	–
Merge accounts	–

Table 4: Required security scale–level for the features

application as low security demanding. However the high–security feature is considered central to the application so removing it is not an option. Another solution would be to use a PASS.

Using a PASS the evaluation method could be run twice, once for the high–security case and once for the low–security case. This solution will be used since it is in line with what the company wants to achieve. The first run will require high security and the second run will allow both low– and high–security (since there is no problem in authenticating the low security demanding features with high–security).

5.3.2 Ease of Deployment

For the first usage of the evaluation method the security scale will be set to high, and that means some trust has to be established between the SP and the IMS. To achieve this, some agreement has to be reached between the SP and the IMS that increases trust. However since the case study is required to be developed quickly and without extensive prior knowledge, some preexisting software is requested. Thus level 2 is used for the first run through.

In the second usage no agreements are required for security purposes. While the low development time, and available software, is still required. That will mean that level 4 will be used for the second evaluation.

5.3.3 Certainty of Identity

In the first adhibition of the test application is the importance of certainty of identity very high as it will handle monetary transactions [15].

During the second usage the certainty is not as important, as the only information requested from the IMS is a unique identifier.

5.3.4 Evaluation Conclusion

Table 5 show a summary of the case study analysis. Using these values to filter the list of IMSs will generate the list of acceptable IMSs. The result can be seen in Table 6. The only IMS, from the input, that satisfy the high–security demanding run's requirements is *BankID*. For the low–security run both *OpenID* and *Facebook Connect* satisfy the requirements. When determining which IMS to choose the evaluation method advocates using the IMS with the most users. In this particular instance that is not trivial however; the exact number of users using *OpenID* is not known and at the same time *Facebook*'s user base is immensely large. The motivation for considering user adoption was that enabling as many users as possible to be able to access the service. Both user bases are so large that this argument does not seems so important in this instance. So the decision can be made arbitrarily. Kalmar2 was not deemed adequate for either run since it did not meet the security and trust requirements for the first run and it did not meet the ease of deployment requirements for the second run.

Runs	Security	Ease of Deployment	Trust
High–security	high	2	high
Low–security	low	4	low

Table 5: Case Study evaluation values

IMS	First run	Second run
Facebook Connect	—	✓
BankID	✓	—
OpenID	—	✓
Kalmar2	—	—

Table 6: Evaluation method output

6 Conclusion

In this section the results are criticized and some experiences that were learned during the thesis work are presented.

6.1 Authentication Service Evaluation

In this section problems and other observations about the authentication service evaluation are discussed.

Based on the results from the case study, the evaluation method seems to work quite well. The results were, for the most part, the expected outcome. A bit surprising was that *Kalmar2* did not fit into either of the categories. At first glance it seems like a very reliable service, but due to the relaxed *SAML*–profile used in the federation it scores lower than expected on the related scales. While surprising, it could indicate a need for an evaluation method such as this.

When applying the model the user may find that the scales are not entirety disconnected; for example requiring high Certainty of Identity will make low–security IMSs less attractive. The scales do in fact influence each other rather much. The evaluation method would be easier to use if the scales could be considered more separately. However this was not attainable in this thesis.

Looking at the literature it seems like trust is a very central concept when utilizing third–party identity management, something that can be seen in that two out of three scales depend on trust between the IMS and SP. The possibilities in a low–trust environment are really quite limited, as has been highlighted throughout the report. Trust, or rather the lack thereof, could be the largest obstacle for the progression of third–party identity management.

6.2 Plug–in Authentication Security Service

In this section problems and other observations about PASS is discussed. Based on the information provided in section 5 the test application, including a PASS, was implemented. While the application was never tested in a live environment the results of in–house testing was promising. All solutions to problems relating to PASS that was described in Result was implemented and worked as intended.

However, there are some problems with PASS. If, in a theoretical case, all possible IMSs where to be implemented into a single Plug–in Authentication Security Service, there could still be users not able to log in to the system; there may exist some people with no account on any IMS. To accommodate these users the developers can choose either to implement a local registration feature, effectively counteracting some of the benefits of an IMS solution, or to recommend users without accounts to create an account on an IMS supported by the service. This business decision will have to be made in every implementation of a PASS; neither is recommended over the other by this thesis.

6.3 Future work

The evaluation method provided good results in the case study performed in this thesis, however before anything definite can be said about its quality more case studies needs to be performed. In particular what would be interesting to see are more concrete implementations that are based on results from the evaluation method. This would help strengthen the assertion that the result is good and of actual interest to developers.

Further research into the trust aspect is important if the IMS paradigm is to gain further momentum. Looking at the IMSs presented in this report and also other IMSs makes it clear that there is a lack of trusted IMSs on the Internet today. An example of this is *OpenID* which have seen a lot of companies announcing IdP support to their users but the same companies do not act as SPs [16].

References

[1] I. Agudo-Ruiz. "Digital Identity and Identity Management Technologies". In: *Serbian Publication InfoReview joins UPENET, the Network of CEPIS Societies Journals and Magazines* (2010), pp. 6–12.

[2] R. Shirey. *Internet Security Glossary*. 2000. URL: `http://www.ietf.org/rfc/rfc2828.txt` (visited on 2011-10-22).

[3] M. Ates et al. "Interoperability between Heterogeneous Federation Architectures: Illustration with SAML and WS-Federation". In: *Signal-Image Technologies and Internet-Based System, 2007. SITIS '07. Third International IEEE Conference on*. 2007, pp. 1063–1070.

[4] Martin Wolf et al. "A message meta model for federated authentication in service-oriented architectures". In: *Service-Oriented Computing and Applications (SOCA), 2009 IEEE International Conference on*. IEEE. 2009, pp. 1–8.

[5] A. Myllyniemi. "Identity Management Systems: A Comparison of Current Solutions". In: *Security and Privacy in Pervasive Computing*. (Espoo). Ed. by Jukka Manner. 2006.

[6] I. Thomas, M. Menzel, and C. Meinel. "Using quantified trust levels to describe authentication requirements in federated identity management". In: *Proceedings of the 2008 ACM workshop on Secure web services*. ACM. 2008, pp. 71–80.

[7] Gergely Alpár, Jaap-Henk Hoepman, and Johanneke Siljee. "The Identity Crisis. Security, Privacy and Usability Issues in Identity Management". In: *CoRR* abs/1101.0427 (2011).

[8] Bo Harald. *Final Report of the Expert Group on e-Invoicing*. 2009. URL: `http://ec.europa.eu/internal_market/consultations/docs/2009/e-invoicing/report_en.pdf` (visited on 2011-08-10).

[9] Tobias Straub and Harald Baier. "A framework for evaluating the usability and the utility of pki-enabled applications". In: *Public Key Infrastructure* (2004), pp. 617–617.

[10] U. Kylau et al. "Trust requirements in identity federation topologies". In: *2009 International Conference on Advanced Information Networking and Applications*. IEEE. 2009, pp. 137–145.

[11] A. Jøsang et al. "Trust requirements in identity management". In: *Proceedings of the 2005 Australasian workshop on Grid computing and e-research-Volume 44*. Australian Computer Society, Inc. 2005, pp. 99–108.

[12] G. Elahi, Z. Lieber, and E. Yu. "Trade-off analysis of identity management systems with an untrusted identity provider". In: *Computer Software and Applications, 2008. COMPSAC'08. 32nd Annual IEEE International*. IEEE. 2008, pp. 661–666.

[13] B. van Delft and M. Oostdijk. "A Security Analysis of OpenID". In: *Policies and Research in Identity Management* (2010), pp. 73–84.

[14] Anders Bjerg Pedersen. "Usability of authentication in web applications–a literature review". 2010.

[15] F.F.I.E. Council. *Supplement to Authentication in an Internet Banking Environment*. 2011. URL: `http://www.ffiec.gov/pdf/Auth-ITS-Final%206-22-11%20(FFIEC%20Formated).pdf` (visited on 2011-08-10).

[16] K. Helenius. "OpenID and identity management in consumer services on the Internet". In: *Seminar on Internetworking*. 2009.

[17] Å. Grönlund. "Electronic identity management in Sweden: governance of a market approach". In: *identity in the information society* 3.1 (2010), pp. 195–211.

[18] Andreas Åkre Solberg et al. *Interoperable SAML 2.0 Web Browser SSO Deployment Profile*. 2011. URL: `http://saml2int.org/profile/0.1` (visited on 2011-08-10).

[19] Joshua B. Bolten. *Memorandum to the heads of all departments and agencies*. 2003. URL: `http://www.whitehouse.gov/sites/default/files/omb/memoranda/fy04/m04-04.pdf` (visited on 2011-08-12).

[20] Finansiell ID-teknik. *Remissvar på betänkandet E-legitimationsnämnden och Svensk e-legitimation (SOU 2010:104)*. 2011. URL: `http://www.regeringen.se/content/1/c6/17/17/80/d51a6562.pdf` (visited on 2011-08-10).

Glossary

Authentication — The act of proving ownership of a identity.

Biometrics — Measurements for uniquely identifying a user based on physical traits. Examples are fingerprints, face recognition and voice recognition.

Certainty of Identity — The certainty that the identity can be tied to a physical person.

COTS — Commercial Of The Shelf, product or module bought in from outside the company to be fitted into a product.

Identity — A digital persona used by a person.

Identity Management solution (IMS) — Umbrella term for any type of identity provider service. For example federation or proprietary authentication service.

Identity Provider (IdP) — A service that provides and manages identities.

Invoice — Document detailing products or services bought. The invoice specifies the amount of money that the buying part must pay the selling part.

Man–in–the–middle attack — Malicious activity in digital communication. The attacker intercepts, listens to and/or alters all messages sent between the benevolent parties.

Persistent id — A pseudonym used to allow identification of user between sessions in an IMS environment.

Plug–in Authentication Security Service (PASS) — System introduced in this thesis; A system for handling authentications using multiple IMSs.

Pseudonym — A false name used to hide the users true identity.

Register — Providing user credentials to a site, to use at later occasion as authentication.

Service provider (SP) — An (web)application that provides some service to end–users.

Token — A unique value that can be used to identify some object.

A Identity Management Solutions

A.1 OpenID

Defined in 2005, this open standard has grown fast with many large companies and organizations supporting it (e.g. Google, Yahoo! and Symantec). Any user with an account to one of these sites can use it as an *OpenID*.

OpenID uses an *URL* as the unique identifier for each user. *OpenID* is a decentralized standard and the identifier *URL* helps the service provider locate the identity provider [3]; the target of the identifier *URL* provides information about where the service provider should redirect the user for authentication. The service provider supplies information to the identity provider in the form of HTTP-parameters, using a client–side redirect.

When the user arrives at the identity provider she authenticates herself using whatever mechanism the identity provider offers. On a successful authentication the user is returned to the service provider (more precisely to a *URL* specified in the redirect to the identity provider). With the user comes an authentication response that asserts that the user is authenticated.

Since *OpenID* is a decentralized and open IMS anyone can be an *OpenID* IdP. It is therefore hard to determine the exact number of users of this IMS, but only Yahoo's email services have more than 200 million users and Google has approximately 300 million users with Google accounts.

A.2 Kalmar2

The *Kalmar2* Union is a federation between the academic institutions in the Nordic countries. Every country in the union has its own federation of academic institutions:

- Finland (*Haka* federation)
- Norway (*FEIDE* federation)
- Denmark (*WAYF* federation, also includes Iceland)
- Sweden (*SWAMID* federation)

The amount of users are hard to determine, since every country has different policies regarding publication of such numbers, however most students should, by being registered at an academic institution, be a user of the *Kalmar2* federation. So the number of users should be approximately the same as the number of students in the Nordic academic institutions.

Kalmar2 uses *SAML* for communication between the IdPs and the SPs. *SAML* is an XML-based framework for describing and exchanging security information between security domains. *SAML* is very comprehensive and it can satisfy many use cases (a good source of information about *SAML* is the official technical overview [4]). This text will only provide a brief description of the relevant use of *SAML*.

[3] the description of the *OpenID* protocol in this text is a simplification of the complete specification, for a more detailed explanation see http://openid.net/developers/specs/

[4] http://saml.xml.org/wiki/saml-introduction

When a SP wants to authenticate a *Kalmar2* user it must first learn which federation member the user wants to authenticate at. The SP retrieves a list of IdPs from *Kalmar2* and offers the list to the user. The user then chooses the IdP she wants to authenticate at. The SP can now create a SAML–authentication request. The request may contain various types of information, such as who is issuing the request and the identifier type to use. The user is then sent to the selected IdP along with the authentication request.

The IdP examines the authentication request and if found valid offers the user a possibility to authenticate. The IdP authenticates the user, and if successful returns the user to the SP along with an authentication response. The content of the response is, in part, dependent on what was in the authentication request. *SAML* allows for issuing of user information to the SP and if such was required by the SP, in the authentication request, they will be sent as assertions in the authentication response. *Kalmar2* requires all participants to support the eduPerson [5] scheme for user attributes.

The only attribute PASS is interested in is some persistent id that allows for the user to be connected with information at the SP. *Kalmar2* requires that the IdP support either eduPersonTargetedID (persistentID) or eduPersonPrincipleName [6] which both works as a persistent id. *Kalmar2* encourages the use of eduPersonTargetedID since it is a pseudonym, which enhances privacy. When the SP receives the authentication response it validates it and if found valid allows the user access.

A.3 Facebook Connect

The possibility to use the social media network, *Facebook*, as an IMS was released in late 2008. All Facebook users (over 750 million active users [7]) can use their account to log in to sites supporting Facebook as an IMS.

Facebook uses the *OAuth 2.0 protocol* [8] when providing authentication services. In particular they offer support for the *Authentication code flow* and the *Implicit flow* for authentication. The two flows are optimized for different authentication scenarios. This text will only provide a brief description of the *Authentication code flow* since that is the flow that is interesting for the plug-in authentication security service (PASS). A more in-depth description of both flows is given in [8].

OAuth 2.0 requires that the SP registers at the IMS, in this case Facebook, and receives an application id. When the SP later wants to get a user authenticated it sends the user together with the application id to *Facebook*'s *OAuth*-authentication handling page. At *Facebook* the user is asked to login to her *Facebook* account and is then asked if she accepts allowing the SP to access her account. *OAuth* is designed to allow services to access the user's resources at the IMS. In the case of Facebook the resources refers to personal information, pictures, and other information the user has uploaded to *Facebook*. It is because of this the user is required to accept the SP's access request.

If the user agrees to allow the SP access to the user's information she is redirected back to the SP's domain with an authorization code. The SP then request an access code from *Facebook*

[5] http://middleware.internet2.edu/eduperson/
[6] http://www.kalmar2.org/kalmar2web/members_attchmt/appendix_A_2010_10_25.pdf
[7] http://www.facebook.com/press/info.php?statistics
[8] http://tools.ietf.org/pdf/draft-ietf-oauth-v2-12.pdf

using its application id and the authorization code received when the user was redirect to the SP's domain. *Facebook* validates the provided data and if valid issues the access token to the SP. Using this token the SP can query *Facebook* for information about the user.

When the SP receives the access token it knows that the user has been successfully authenticated. However the only reference the SP has to the user is a time-limited access token to her account. This is not sufficient if the SP needs to connect local information to the user. Information that could be accessed the next time the user log in at the SP. To remedy this *Facebook* offers access to a pseudonym that can be used to connect the user with local data. The final step the SP needs to perform is send an *OAuth*–request to *Facebook* for the *third_party_id*–attribute.

A.4 BankID.se

In Sweden most of the major banks are cooperating since 2003 to support the IMS *BankID*. Today there are around 2.5 million active *BankID* users in Sweden [9], and more than 600 web services supporting *BankID*. Many of these services are run by government authorities, and over 90

BankID uses a PKI–solution for authentication and it uses the Swedish citizens personal security number for identifying users [17]. Since *BankID* is a proprietary service the technical details are not publicly available. However the basic concept is that the banks provide a software client that runs on the user's machine. This software handles the user's certificate. It requires the user to enter a password to access the certificate and then validates the certificate. If the authentication succeeds the software client informs the browser that the SP is running in and the user is authenticated.

[9] http://sv.wikipedia.org/wiki/Bankid

B Identity Management Solutions Evaluation

Below follows a brief analysis of what scale-values each of the four IMSs presented in Appendix A have. In Table 7 the result of the analysis can be seen.

B.1 Security

Both *OpenID* and *Facebook Connect* offer an optional multi-factor authentication [10] [11] .Neither of the IMSs enforces the use of this feature, which means that the evaluation method cannot assume that the user uses the higher security level.

Kalmar2 uses *SAML* and applies the *Interoperable SAML 2.0 Web Browser SSO Deployment Profile* which does not require multi-factor authentication [18]. While the specification does not forbid multi-factor authentication a service provider cannot be certain that all authentications are done with higher security so *Kalmar2* has the lower security level.

BankID uses a PKI-solution with password protected certificates. That is, a two-factor authentication (something you know and something you have), so *BankID* is classified as having the higher level of security.

B.2 Ease of Deployment

OpenID has a multitude of software that provides APIs toward the service [12] and *OpenID* does not require any agreement to be signed before a service provider can start using it, so *OpenID* has the highest value of ease of deployment (level 4).

Facebook Connect uses *OAuth2.0* for communication between the IMS and the service provider. They also offer a JavaScript library, which is the recommended way of using *Facebook Connect* [11] . *Facebook Connect* requires that the SP registers their application at *Facebook* in order to use *Facebook Connect*. However the registration process is automatic; the registration is used to generate an *OAuth*–application id for the SP and also determine which permissions the SP is allowed to request from *Facebook*. So even though *Facebook Connect* requires SP registration it is level 4, since the registration is automatic.

Kalmar2 uses *SAML* for communication between the IMS and the service provider, so software that provides APIs toward *SAML* can be used with *Kalmar2*. *Shibboleth* provides such an API[13] and there are also some other software that provide APIs toward *SAML* [14] [15] [16]. To become a member of *Kalmar2* you have to join one of the member federations. Each member federation has its own set of license agreements, each a bit different from the others, however the details of the agreements are not interesting when reasoning about the ease of deployment. What is of interest is that they require agreements, which reduces ease of deployment. So *Kalmar2*'s ease of deployment level is 2.

[10] http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html
[11] http://www.facebook.com/note.php?note_id=10150172618258920
[12] http://openid.net/developers/libraries/
[13] https://shibboleth.net/
[14] http://www.componentspace.com/Products/SAMLv20.aspx
[15] http://simplesamlphp.org/
[16] http://code.google.com/p/python-saml2/

BankID is issued by a collection of banks in Sweden, and in order to use it an SP needs to sign an agreement with one of the participating banks [17]. Another possibility offered is to sign an agreement with a third-party that offers both software and licence agreement. There are also third-parties that offer only software solutions which can be used if the service provider chooses to sign an agreement directly with one of the banks. Whatever solution the service provider chooses she needs an agreement and there are available software APIs so *BankID*'s level is 2.

B.3 Certainty of Identity

Since anyone can be an *OpenID* IdP, there is no way for the service provider to know, and maybe trust, every IdP, which means that it would not be possible to verify the user registration used by the IdP. Nor is it possible for the service provider to sign agreements with all of them. In other words, *OpenID* has the low certainty of identity level. This conclusion is also reached in [13], where they conclude that *OpenID* should only be used with applications that only require a level of assurance of 1 [19]. The same reasoning and conclusion is reached in [16].

In its user agreement [18] *Facebook* says that the user has to provide her proper name, however there is no mandatory verification that this rule is followed. Nor are there any agreements that create certainty of identity between *Facebook Connect* and the service provider. In other words *Facebook Connect* has low trust.

As described in Appendix A *Kalmar2*'s users are students at universities in the Nordic countries. As such the certainty of identity is good for this IMS. *Kalmar2* uses the emphInteroperable SAML 2.0 Web Browser SSO Deployment Profile which does not require validation of signed authentication requests [18] which means that the service provider cannot rely on the assertions to be valid. So while there is an agreement between the service provider and the IMS it does not require reliable identity assertions, so while the IdPs of *Kalmar2* have users that can be connected to physical persons the IMS does not fulfill the trust requirements, so the certainty of identity level of *Kalmar2* is low.

BankID is a government approved e-identity within Sweden [19] and as such has a good user registration; when applying for a *BankID* the user has to identify herself with some identity credentials. To be able to utilize *BankID* you need to sign an agreement with one of the participating banks, or one of their partners. From this we get that *BankID* provides high certainty of identity.

B.4 User Adoption

Many sites provide their users with *OpenID* credentials, acting as IdPs themselves. With the multitude of IdPs follow a problem with determining the amount of users, since few sites provide information about the amount of users they have. Also it is possible to have an *OpenID* provider completely hidden from everyone but its users. To provide a very rough

[17] http://bankid.com/sv/BankID-i-dina-tjanster/Sa-kommer-du-igang/

[18] http://www.facebook.com/terms.php

[19] http://avropa.nu/Hitta-ramavtal/Ramavtalsomraden/IT-och-telekom/Elektronisk-identifiering-eID-2008/

estimate one can examine the larger providers that do provide their numbers; for example, Yahoo mail has more than 275 million users [20].

Facebook Connect can be used by anyone with a *Facebook* account, and according to *Facebook* themselves they have over 750 million users [21].

All students in the Nordic countries have the possibility to use *Kalmar2* as their IMS, using the login information provided by their academic institution. Since this IMS is a cooperation of several federations with several IdPs and the policies of publication differs between country and institution it is hard to determine how many people can access this IMS. However, since the IMS is managed by the academic institutions in the Nordic countries and the majority of IdPs are universities in the same countries a rough estimate of the user count would be the number of registered students at the universities.

Many major banks in Sweden support *BankID* and offers an identity to their customers, there exists a few competing systems, however *BankID* is the largest with about three million users [20].

B.5 Summary

The following table contains the summary of the reasoning in this section Table 7.

IMS	Security	Ease of Deployment	Certainty of Identity	Users
Facebook Connect	low	4	low	750 million
OpenID	low	4	low	N/A
Kalmar2	low	2	low	N/A
BankID	high	2	high	3 million

Table 7: Evaluation values for some well-established IMSs

[20] http://blog.unica.com/full-metal-email-confessions-of-an-anti-spam-zealot/
[21] http://www.facebook.com/press/info.php?statistics