Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Degree Programme of Computer Science and Engineering

Jussi Malinen

Identity Information Transfer and Federation on Ubilogin Authentication Server

TEKNILLINEN KORKEAKOULU
TIETOTEKNIIKAN TALON KIRJASTO
KONEMIEHENTIE 2
02150 ESPOO

Master's Thesis
Espoo, 5th April, 2010
Printing date: 18th May, 2010

Supervisor:	Professor Tuomas Aura, Aalto University
Instructors:	Professor Sasu Tarkoma, Aalto University
	Yrjö Kari-Koskinen M.Sc. (Tech.), Ubisecure Solutions Oy

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Jussi Malinen
Title of thesis:
Identity Information Transfer and Federation on Ubilogin Authentication Server

Date:	5th April, 2010	**Pages:**	72
Professorship:	Data Communications Software	**Code:**	T-110

Supervisor: Professor Tuomas Aura
Instructors: Professor Sasu Tarkoma
Yrjö Kari-Koskinen M.Sc. (Tech.)

Increasing number of user accounts and segmentation of user identity information into separate identity silos is becoming problematic both for users and service providers. Identity federation is a way to mitigate this problem, by enabling single sign-on between services and identity information sharing between identity silos.

In this thesis we examine four specific identity federation scenarios and present a number of use cases for each and we lay out an evaluation criteria for the use cases. Then Ubilogin, a federated single sign-on system by Ubisecure Solutions, is evaluated against the requirements of each use case and a number of possible models for improving the system are analyzed. Especially pseudonym support and federation partner discovery are discussed and changes recommended. Also two different models for handling the external federation links, direct federation and central IDP proxy, are analyzed and central proxy is found to be a useful model in many situations.

The changes were implemented by a group including the author and the new version of Ubilogin is evaluated again against the use case criteria. Also a new tool called Federation Manager is introduced and is found to be useful in simplifying handling of the certain use cases.

Keywords: SAML, Single Sign-On, Identity, Federation
Language: English

Aalto-yliopisto
Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan koulutusohjelma

A?
DIPLOMITYÖN
TIIVISTELMÄ

Tekijä: Jussi Malinen
Työn nimi:
Tunnistustiedon siirtäminen ja federointi Ubilogin Authentication Serverissä

Päiväys:	5. Huhtikuuta 2010	**Sivumäärä**	72
Professuuri:	Tietoliikenneohjelmistot	**Koodi:**	T-110
Työn valvoja:	Professori Tuomas Aura		
Työn ohjaajat:	Professori Sasu Tarkoma Diplomi-insinööri Yrjö Kari-Koskinen		

Yhä lisääntyvät käyttäjätilit ja käyttäjätietojen segmentoituminen eri identiteettisiiloihin on kasvava ongelma sekä käyttäjille että palveluntarjoajille. Identiteettien federointi lievittää näitä ongelmia mahdollistamalla kertakirjautumisen palvelusta toiseen ja käyttäjätietojen jakamisen identiteettisiilojen välillä.

Tässä työssä tutkitaan neljää identiteettien federointiskenaariota ja esitetään käyttötapauksia kuhunkin liittyen. Lisäksi esitetään evaluaatiokriteerit käyttötapauksia varten. Sen jälkeen evaluoidaan Ubilogin-kertakirjautumispalvelin näitä kriteereitä vasten ja mahdollisia malleja järjestelmän parantamiseksi analysoidaan. Erityisesti pseudonyymitukea ja federaatiopartnerien valintaa tutkitaan ja muutoksia ehdotetaan. Myös kahta ulkoisten federaatiolinkkien hallintamallia, suorafederointi ja keskitetty tunnistusvälipalvelin, analysoidaan ja keskitetty tunnistusvälipalvelin todetaan hyödylliseksi malliksi monissa tilanteissa.

Suositellut muutokset toteutettiin ryhmässä johon myös kirjoittaja kuului ja uusi Ubilogin-versio evaluoidaan uudelleen käyttötapausten kriteerejä vasten. Myös uusi työkalu nimeltään Federation Manager esitellään ja todetaan hyödylliseksi tiettyjen käyttötapausten yksinkertaistamisessa.

Avainsanat: SAML, kertakirjautuminen, identiteetti, federointi
Kieli: englanti

Contents

Abbreviations and Notations

B2B	Business to Business
B2C	Business to Consumer
B2E	Business to Employee
CDC	Common Domain Cookie, a method for IDP discovery specified by OASIS
ETSI	European Telecommunications Standards Institute
FINUID	Unique Electronic Client Identifier, the Finnish national electronic identity number (sähköinen asiointitunnus)
G2C	Government to Citizen
G2G	Government to Government
HTTP	Hypertext Transfer Protocol
ID-WSF	Identity Web Services Framework
IDP	Identity Provider
LDAP	Lightweight Directory Access Protocol
MSS	Mobile Signature Service, a standard defined by ETSI for creating digital signatures using a mobile phone
OASIS	Organization for the Advancement of Structured Information Standards
OTP	One-Time Password
PIN	Personal Identity Number, the finnish national identity number (henkilötunnus)
RBAC	Role Based Access Control

SAML	Security Assertion Markup Language, an XML-standard for exchanging authentication and authorisation data between security domains
SOAP	Originally: Simple Object Access Protocol, as of SOAP v1.2 not an acronym anymore
SP	Service Provider
SSO	Single Sign-On
UAS	Ubilogin Authentication Server
UML	Unified Modelling Language
URL	Uniform Resource Locator
WSIDP	Web Service Identity Provider
XML	Extensible Markup Language

Chapter 1

Introduction

According to CIA World Fact Book the Internet had 1,6 billion users in 2008 [CIA2009]. Internet has become an integral part of our everyday lives, the way we keep in touch with other people, shop and the way we conduct business. This all despite that the internet has no built in security, privacy or concept of user identity. This limitation of Internet is showing in the multitude of different user accounts and new passwords that we are forced to create in order to use all these services online.

The increasing number of user accounts and segmentation of user identity information into separate identity silos is becoming problematic for the users who don't want to create and remember new accounts and credentials. Thus it also becomes a problem for the service providers as well, who need to identify the users, but are faced with users who are inconvenienced and might even choose not use the service for the hassle of registering.

Identity federation is a way to mitigate this problem by allowing users to move between services without the need to reauthenticate and enabling service providers to share identity information between identity silos. A number of protocols have been designed for this purpose, one of the most popular at this time being Security Assertion Markup Language (SAML) [SAML-tech-overview]. These protocol standards do not however describe in detail how the federation should be set up in different use scenarios. For example the privacy and security requirements of businesses offering services to consumers are rather different from for example organizations handling identities of their own employees.

In this thesis we analyze how a federation enabling identity provider server – Ubilogin Authentication Server (UAS) – can be used to implement use cases in four specific use scenarios: Business to Employee (B2E), Business to Business (B2B), Business to Consumer (B2C), and Government to Citizen (G2C). After the analysis, we recommend changes to UAS which were implemented by the development team which includes the author.

We also present a reanalysis of the scenarios based on the new version of the server.

Problem statement The three goals of this thesis are:

1. Analyze how UAS can handle the use cases presented in each of the given scenarios.

2. Recommend how UAS could handle the use cases better.

3. Analyze how the implemented changes work in next version of UAS.

The analysis for the first two research questions was done on UAS version 4.1 and the results are documented in chapter 4. Many of the recommendations were implemented in next UAS version 5.0 and the results are analyzed in chapter 5, documenting how the new version of UAS handles the scenarios. Also the evaluation of the implementation of each scenario is given based on evaluation criteria laid out in chapter 3.

Chapter 2

Background

This chapter introduces the necessary background for this thesis. First we introduce the concepts of identity, authentication and authorization. Then we take a look at federation which means moving identities, authentication, and authorization information between different domains. Finally we look at the different technical means of achieving identify federation.

2.1 Identity, Authentication, and Authorization

2.1.1 Identity and Identifiers

Camp defines *identity* in an identity management system as a set of permanent or long lived attributes associated with an entity. Here, the entity could be a human being, as well as a computer, a software process, or an organization. For humans, typical attributes associated with the identity would be name, date of birth, email-address, and so on. Some of these attributes uniquely identify an identity, like for example email address would do when dealing with humans. Camp defines these identifying attributes as *identifiers* [Cam04a].

Different systems need different attributes with the identities and it is often in the interests of the privacy and security of the entity to limit the number of attributes made available to the system to only the bare minimum required. Thus an entity might have many identities in different systems. Linden defines the set of all these *partial identities* as the one universal identity of the entity. [Lin09, Chapter 2.1]. For the purposes of this thesis, when we speak of an identity we mean the partial identity associated with the identity management system in question.

Personal Identity Number and Unique Electronic Client Identifier

In Finland two commonly used identifiers are the Personal Identity Number (PIN) and the Unique Electronic Client Identifier (FINUID). PIN, or henkilötunnus in Finnish, is an identifier that is given to each citizen of Finland by the Population Register Center, and stays the same throughout their life unless the person in question changes his or her sex [886/1993] or if their birth date is corrected. This identifier is widely used especially in Finnish government and banking services. Besides the problem of it at times changing, it has the problem of revealing the person's sex and date of birth, which can be a privacy issue.

Another identifier also given by Population Register Center is FINUID, or sähköinen asiointitunnus in Finnish. FINUID also uniquely identifies a person, but unlike the PIN, it is opaque, meaning that it does not reveal any other information about the person, and therefore does not need to change when a person's sex is changed. The purpose of FINUID is as an identifier in e-services. The Finnish national identity card contains a FINUID as the unique identifier [507/1993].

Many other countries have similar identifiers for all citizens. For example in Denmark, the CPR (Danish civil registry number) or the OCES (Danish electronic identity number) are used in electronic services [DK-SAML]. In Sweden, personnummer, an identity number similar to the Finnish PIN , is given to all Swedish citizens [SWE 481/1991].

However unique national identifiers are not universally used in all countries. For example, in the United States of America there is no mandatory identifier for citizens, although there are claims that the social security number has become a de facto mandatory identifier [Kou05]. Also in the United Kingdom there is no unique identifier for citizens, but instead two different identifier systems: the national Insurance number and the national health service number, which both might change many times during a persons lifetime and are not used as widely as the United States social security number [Wal03].

In this thesis, when analyzing electronic services offered to citizens by the government, we are assuming a national environment where a unique national identifier is available, therefore some of the results might not generalize to nations such as the United States or the United Kingdom.

2.1.2 Authentication

According to Camp *authentication* means proof of an attribute, further noting that identity as it is constructed in identity management systems is an attribute [Cam04a]. Proof is often (see for example [Ren05]) categorized to the following three authentication types:

1. Something you know, for example a password
2. Something you have, like an authentication token
3. Something you are, such as fingerprints

When more than one of these authentication types are combined, it is called two-factor or three-factor authentication, or simply *strong authentication* . Examples of strong authentication on the internet are mobile phone based European Telecommunications Standards Institute (ETSI) Mobile Signature Service (MSS) [ETSI-MSS] and smart cards. In these something the person has (the smart card or phone subscriber identity module) is combined with something the person knows (the pin code).

2.1.3 Authorization

Camp defines *authorization* as the decision to allow an action based on identifier or attribute [Cam04a]. Three common access control models are Mandatory Access Control, Discretionary Access Control and Role Based Access Control (RBAC).

Mandatory Access Control is defined by Department of Defense as security level based access control in which each subject and object are assigned sensitivity labels that are combinations of hierarchical classification levels and non-hierarchical categories [Dep85]. This model is criticized by Anderson for being complicated and hard to integrate to applications [And01].

Discretionary Access Control is lighter alternative to mandatory access control. Department of Defense defines it as an access control in which each object has an associated access control list of subjects [Dep85]. This model is used for example in Unix based operating systems to control file access.

In RBAC model users are given roles which in turn give users permission to access specific operations. Sandhu et al note that this provides RBAC a level of abstraction to access control that allows administration of security policy to focus on higher level roles rather than to individuals [SFK00]. Therefore this model is well suited for web applications and our study is mostly using this model in access control analysis.

2.2 Federation

Whether we are talking about consumers in the internet or for example corporate users accessing partners extranet sites, users tend to have many different user accounts in different services, each service and its associated identity information forming a so called identity silo [PG07]. When users have different logins and passwords to accounts for each identity silo, it can become a hassle to remember all the different passwords, not to mention to keep the information in all identities up to date. Also the administration of user accounts and roles becomes a burden to service providers. For example in case of business to business services, the service provider might prefer that all user account administration is done at the user's home organization locally, instead of the service provider trying to keep up with all the changes in partner organizations workforce. One solution to these problems is federation between the identity domains.

Ihalainen defines identity domain as "a self contained system that maintains a repository of identity information about its users". So according to this definition, each identity domain has its own identity silo. Federation is defined by Ihalainen as "a transfer of user identity between two different domains" [Iha07]. Federation allows linking and mapping users accounts between domains, thus allowing users to only use one login and password pair to access services in two or more different domains. Federation can also allow transferring identity information between domains in order to keep users information up to date and to diminish administration work for service providers.

One of the goals of federation can also be Single Sign-On (SSO) between identity domains, meaning that once user logs in to a service, he can continue to other services without the need to login again [SAML-tech-overview].

2.2.1 Pseudonyms

The actual identity information that is transferred between the domains differs depending from needs of the use scenario. Often simply transferring the user's login name as an unique identifier can be enough or in some cases it might be acceptable to actually send all user's identity information to another domain. However in real world, the user's identity information is often private and only minimum amount on confidential identity information should be transferred between identity domains [Liberty-overview]. The Security Assertion Markup Language (SAML) set of standards, developed by Organization for the Advancement of Structured Information Standards (OASIS) Security Services Technical Committee, proposes pseudonyms [SAML-core] as the solution.

Pseudonym is defined by Liberty Alliance as an arbitrary identifier assigned by the identity or service provider to identify a principal to a given relying party so that the name has meaning only in the context of the relationship between the parties [Liberty-glossary]. Using pseudonyms has the advantage that no confidential information is sent between the federation partners and the pseudonym can not be used to track the user in other services as one pseudonym is used only in one federation context.

2.2.2 Federation patterns

When federation is set up, the participants have to set a contractual relationship with each other and to agree on such issues as what user information is delivered on federation and what are the rights and obligations of federation partners. Windley has identified the following three different contractual patterns of federation [Win05] and these were analyzed further by Linden [Lin09]:

- Ad hoc federation is formed by one to one federation contracts.
- Hub and spoke federation consist of a central organization controlling the federation and managing the contracts and policies.
- Identity network is a federation based on independent federation network that is governed together by the federation network partners, instead of having one central organization controlling it, as in hub and spoke federation, or having all organizations having to set separate federation rules and contracts to manage between each other, as in ad hoc federation.

Linden notes that these contractual models are separate from the technical model of the federation [Lin09, Chapter 5.3.2]. A federation that has a contractual model of an identity network could very well be technically set up to use a central identity provider hub. Our analysis of federation scenarios in chapters 4 and 5 is dealing with technical federation models only.

2.3 Security Assertion Markup Language

Security Assertion Markup Language (SAML) developed by Organization for the Advancement of Structured Information Standards (OASIS) Security Services Technical Committee specifies a format for security tokens, a number of protocols to exchange and manage these tokens and a format for exchanging service description metadata. There are three official versions of SAML the first one 1.0 [SAML-1.0] was succeeded by version 1.1 [SAML-1.1], which introduced a number of corrections and additions to the standards [SAML-Diff-11]. The latest version is 2.0 [SAML-core, SAML-Diff-20] in which a lot of the work done by Liberty on ID-FF set of standards [Liberty-ID-FF] was moved to SAML 2.0.

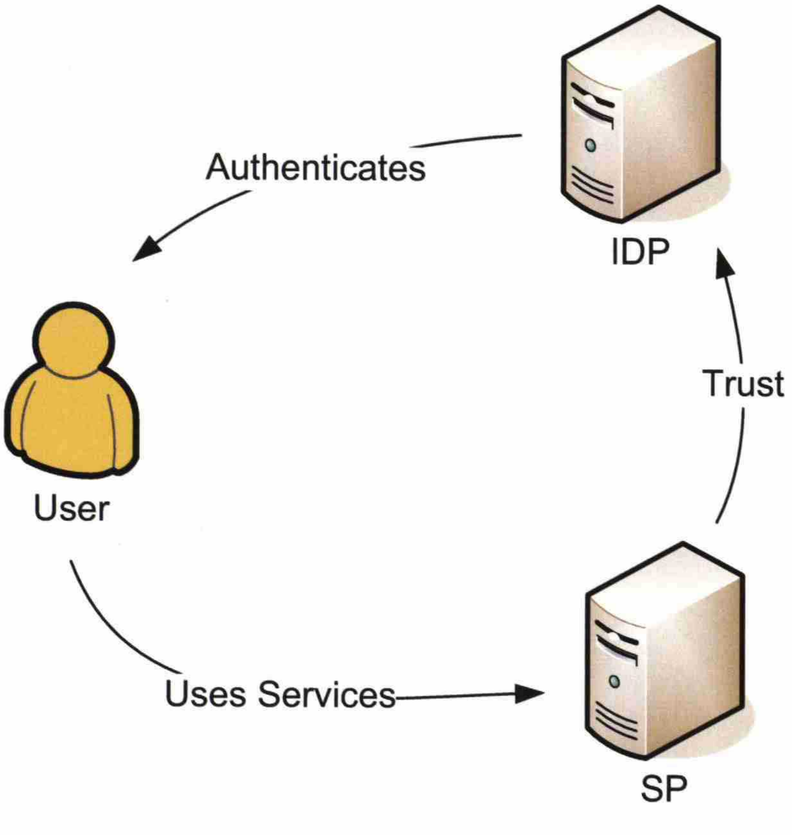

Figure 2.1: The trust model in SAML. The IDP authenticates the user, who uses services at the SP, and the SP trusts the assertions the IDP makes about the user.

One of the most important services that SAML was designed to provide is Single Sign-On (SSO), where a user that has signed on to one service can move directly to another without the need to log on again. The trust model of SAML is based on a relationship between an Identity Provider (IDP) and a number of SPs. The user signs on to the IDP using whatever method the IDP supports, which could be username and password, smartcards or anything else. The actual methods of authentication are specifically left out of scope. After authenticating to the IDP, the user receives an assertion that he/she can use to sign on to the SPs. See Figure 2.1.

The standards have essentially three different levels of services:

1. the assertions containing information about the user [SAML-core]
2. a number of protocol profiles and bindings for requesting and managing these assertions [SAML-bindings, SAML-profiles]
3. a metadata format for describing the offered services [SAML-metadata]

Each of these services builds on top of the earlier, meaning that the assertions can be used alone without the protocol profiles and bindings, but not the other way around. The metadata is meaningful only when used with the protocols.

2.3.1 Liberty Alliance

Liberty alliance has created a set of standards for linking the identities of the users between different services and delivering information about the users in a standard way. The Liberty standards are divided in three main parts.

1. ID-FF [Liberty-ID-FF] was the basis of Liberty's single sign-on and federation framework. It was built on top of SAML 1.1. Now SAML 2.0 has replaced ID-FF
2. ID-WSF [ID-WSF-authn, ID-WSF-client-profiles] is Liberty's federation framework for web services. It supports the development of identity-based services on top of other client programs besides web browsers.
3. ID-SIS [Liberty-tech] is a collection of identity web service specifications. The various specifications include services for requesting and providing users' personal or professional profile information and managing users' contacts online.

Liberty also started the Liberty Interoperable^{TM}program to create a confidential environment in which technology providers could test their adherence to Liberty's specifications and their interoperability. As SAML 2.0 has replaced ID-FF, Liberty now provides test sets for a number of different SAML 2.0 conformance levels [Liberty-Interop].

2.4 Ubilogin Authentication Server

Ubilogin SSO is an Single Sign-On (SSO) and access control solution family that supports modern federation protocols. It is developed by Ubisecure Solutions in Finland. The core component of Ubilogin SSO is the Ubilogin Authentication Server (UAS), which provides authentication, SSO, authorization and federation protocol support for web applications. Nykänen describes the old web application protocol used in earlier versions of UAS [Nyk02]. Käpynen describes how this protocol was later succeeded by SAML [Kä08].

Ubilogin SSO solution family also includes Web Service Identity Provider (WSIDP) support for Web Services as described by Kari-Koskinen [KK07] and a number of integration modules for different application servers. The management and configuration of Ubilogin SSO is done in central Lightweight Directory Access Protocol (LDAP) [Zei06] based Ubilogin Directory. In this study we focus on web applications and UAS, also covering Ubilogin Directory when the actual configuration model is relevant to discussion.

Chapter 3

Identity and access management scenarios

In this chapter we present use scenarios, that we will use for our analysis of UAS. Käpynen [Kä08, Chapter 3] has identified a number of stakeholders in identity federation including users, service providers, government and public sector, private sector, technology providers, and identity providers. In our analysis each identity and access management scenario is characterized by the needs of the two key stakeholders: the SP and the user.

The scenarios we have chosen for this analysis are B2E, B2B, B2C, and G2C scenarios, because we have found them to be amongst the most interesting and relevant ones in our work with SSO and federation systems. This is by no means an exhaustive list of federation scenarios. Especially Government to Government (G2G) services are an important and interesting segment that is right now being addressed in VIRTU project here in Finland [VIRTU]. This study is limited to these four scenarios in order to keep the size of this work manageable.

In this chapter we present the background for each of these scenarios and after that a number of use cases we have derived from the scenario requirements. At the end of this chapter we present evaluation criteria which will be used to analyze how well UAS implements the use cases.

3.1 Business-to-Employee

3.1.1 Background for scenario

One of the simplest identity and access management scenario is that of a business offering services to its own employees. Employees logging into company network are getting used to the idea of a seamless single sign-on experience, where all services in company's intranet and extranet are accessible without separate logins. For example Windows domains provide kerberos-based single sign-on for all domain users [Windows-SSO].

B2E scenario is characterized by the fact that user identities stay inside the same identity domain. Thus the goal is to achieve single sign-on, but identity federation which was defined as identity information transfer between identity domains by Ihalainen in [Iha07] is not part of this scenario. For the purposes of this thesis we assume, that the fact that users stay within the same domain, also means that users do not have additional privacy concerns arising from identity information transfer and federation. After all, all services used in this scenario are offered by the employer who has access to the company's human resources database anyway. In large heterogenous enterprise networks, the users privacy could become an issue, but that scenario is left out of scope of this thesis.

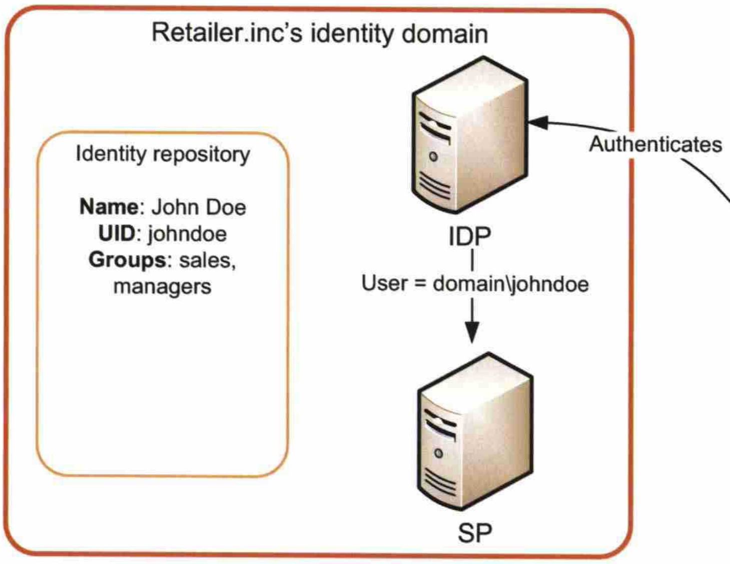

Figure 3.1: B2E authentication use cases, note that both the IDP and SP are in the same identity domain and only the user account name is delivered with authentication.

3.1.2 Use cases

The use cases in this scenario are based on use of out of band information transfer and user identification based on known identifiers, in this case the username of the employee in local domain. In [SAML-tech-overview, chapter 5.4.2] a similar type of identity information transfer, but in their case between different identity domains, is known as "Federation Using Out-of-

Band Account Linking". Figure 3.1 shows the IDP and SP both in the same identity domain of the example organization Retailer.inc.

Use case B2E.1: Web application single sign-on from intranet Retailer.inc offers a web based application to its employees, so that the service may be used from intranet, preferably with the single sign-on session of the operating system. Information that the identity provider passes to the web application is the LDAP user account name of the employee.

Use case B2E.2: Web application offered to employees from public internet Employees can use the same service as in use case 1 also from public internet, using strong authentication. Information that the identity provider passes to the web application is always the LDAP user account name of the employee, regardless of how the user signed in.

3.2 Business-to-Business

3.2.1 Background for scenario

When corporations offer services to other corporations, the model is called Business to Business. Ash has noted that business process integration with outside business partners can optimize the overall B2B value chain and can drive the costs down [Ash01]. Ash has divided this B2B scenario further to Business to Supplier and Business to Corporate Customer scenarios, but for the purposes of this analysis the stakeholders in these subscenarios have similar identity and privacy demands, so they are treated here as the same B2B scenario.

Here the SP is not that much interested in who the user is, but rather about which company the user represents and with what "authorizations" the user has to act on behalf of the company. According to Käpynen [Kä08, chapter 3.4] a B2B service might ignore completely who the user is and determine the user's permissions entirely based on what company the user presents and what roles the user has been assigned in this company. Because of this focus on the company, not the individual user, as the stakeholder, the users in this scenario do not have privacy concerns as individuals. Rather, the corporations the users present might have privacy concerns, such as limiting other users of the service from seeing that they also use it. For example a corporation might find it embarrassing to be publicly associated with a consultant agency specializing in planning layoffs.

Of course not all services offered to business partners can make use of identity attributes and authorization information delivered from an external

IDP. Especially legacy services in corporate extranets might have their own proprietary user database and role management. In these cases local identities and their associated roles at the SP could be linked to federated identifiers using account mapping as described by Käpynen in [Kä08, Chapter 4.1] to achieve single sign-on, if not fully externalized authorization. However this use case is left out of scope of this analysis.

3.2.2 Use cases

The use case 1 below is based on "Federation via Identity Attributes" as presented in [SAML-tech-overview, chapter 5.5]. We use example corporations Importer.inc and Retailer.inc, where importer offers extranet services to its retailers, see figure 3.2.

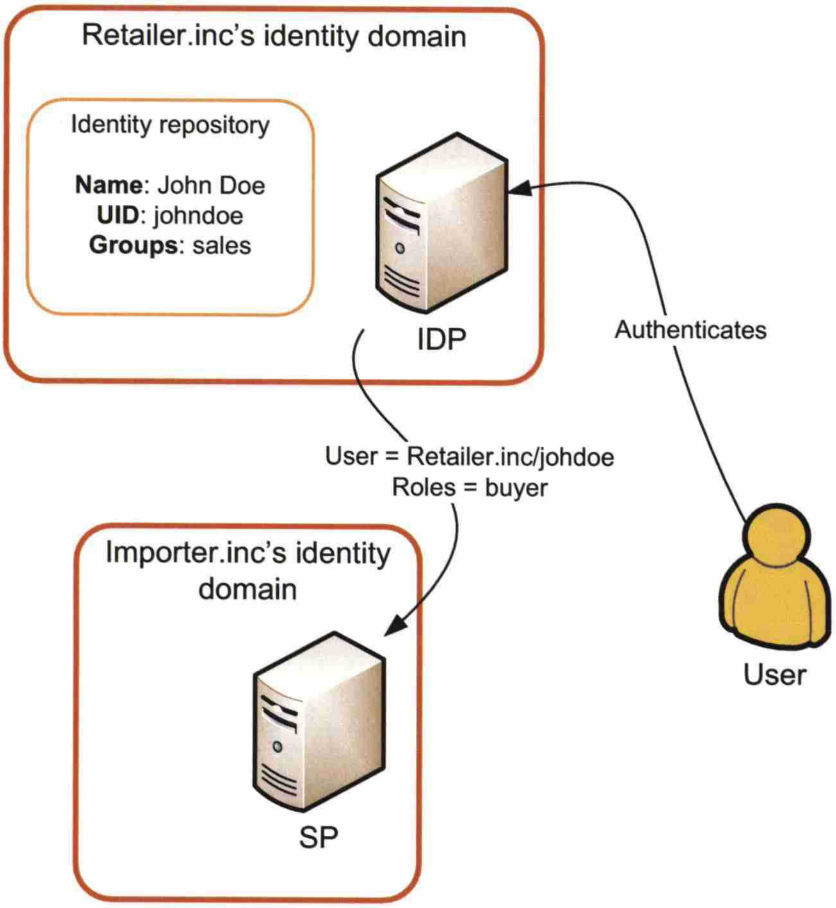

Figure 3.2: B2B authentication scenario, note that IDP and SP are in different identity domains and users roles are delivered with authentication.

Use case B2B.1: Identity federation based on user roles in partner organization Importer.inc offers a web application to its corporate customers, so that the services the user can access are personalized based on the company the user is coming from and role the user has there. The role is assigned from a given set (buyer, auditor, support personnel). Even though permissions in the Importer.inc's web application are based entirely on the company the user is coming from and the role the user has, some kind of identifier of the specific user is also needed for auditing purposes.

If possible, Importer.inc would like to keep it secret which partner organizations' IDPs are trusted from other partners.

3.3 Business-to-Consumer

3.3.1 Background for scenario

When a person orders books from Amazon, buys a flight from an airline's web-page or uses an internet bank, these are all examples of corporations offering services to consumers. This scenario is the one most people are personally most familiar with. Käpynen notes in [Kä08, chapter 3.4] that in B2C services the identity information needs of the SP can vary and while many services require just enough identity information to deliver the expected service and/or charge expenses, some services such as ebanking may require unique trackable identifiers.

Karine Barzilai-Nahon et al present in [BNS07] that customer privacy and clear policies over information disclosure between 3rd parties were among of the main security concerns of eCommerce services targeted for consumers. As users are presenting themselves as individuals in this scenario, it raises more potential privacy concerns than for example in B2B scenario. The use cases presented on Liberty Alliances Architecture Overview [Liberty-ID-FF] focus on this issue, by giving users control over linking user accounts and asking user permission for account linking and information exchange between identity domains.

3.3.2 Use cases

The use cases in this scenario are based on Liberty Alliance's example business to consumer scenario [Liberty-ID-FF], using Airline.inc and Car-Rental.inc as example corporation websites, see figure 3.3. The first use case links user accounts on both services, based on persistent pseudonyms and the second use case is user initiated termination of this federation. These use cases are presented as "Federation Using Persistent Pseudonym Identifiers" and "Federation Termination" in [SAML-tech-overview, chapters 5.4.3 and 5.4.5]. The third use case is based on "Federation Using Transient Pseudonym Identifiers" use case from [SAML-tech-overview, chapter 5.4.4].

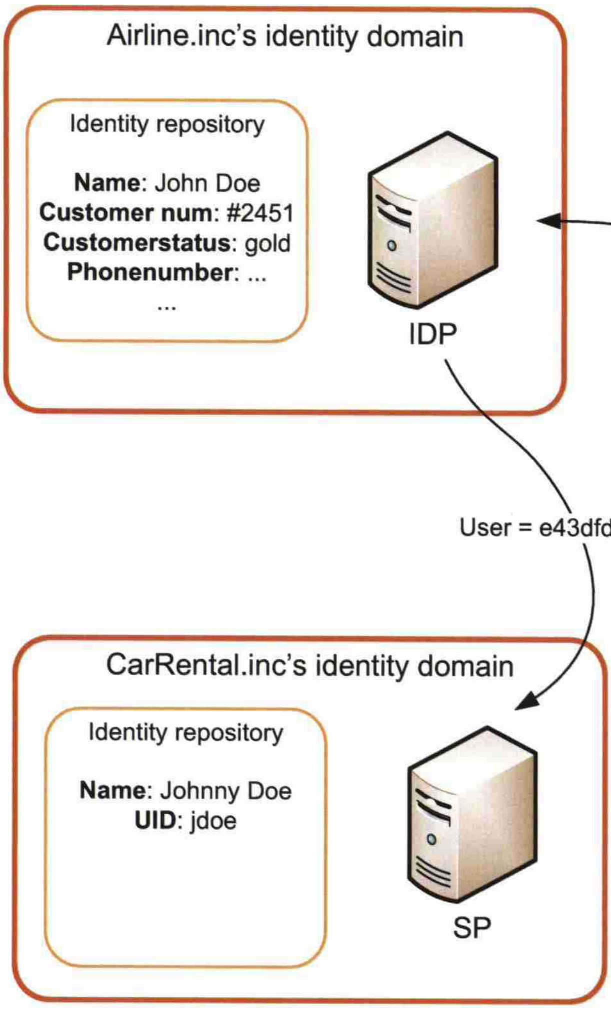

Figure 3.3: B2C authentication scenario, note that IDP and SP are in different identity domains and only an opaque random identifier, a pseudonym, is delivered with authentication.

Use case B2C.1: Identity federation and linking based on persistent pseudonyms The user is a registered customer at the web sites of Airline.inc and CarRental.inc, having a username and a password for logging in at both. At the CarRental.inc's website, the user wants to access a resource that requires authentication. The user chooses to login using identity federation from Airline.inc and is redirected to Airline's web site.

Airline.inc authenticates the user and asks user's permission to federate his account to CarRental.inc. If the user accepts the identity federation, he is redirected back with an opaque persistent pseudonym identifier, so that no personal information is sent from Airline.inc to CarRental.inc.

At CarRental.inc the user is asked to authenticate again, using CarRental.inc's username and password, thereby linking his/her account to the persistent pseudonym from Airline.inc. On subsequent authentications, the federation has already been formed, and user permission or separate authentication at CarRental.inc are no longer needed and account is linked automatically.

Use case B2C.2: Identity federation termination User no longer wishes to keep his accounts on Airline.inc and CarRental.inc federated (for example, if his/her account on CarRental.inc has been disabled.) User clicks a defederation link on CarRental.inc's web site and after that Airline.inc will no longer automatically accept authentication requests for this user from CarRental.inc.

Use case B2C.3: Identity federation based on transient pseudonyms Airline.inc's frequent flyers are entitled to a number of discounts from Hotel.inc's services. Hotel.inc does not want to show the discounted prices, unless they can verify that user is indeed a frequent flyer. On the other hand just for seeing the discounted prices Hotel.inc does not want to force users to do full blown federation as in use case 1, nor force them to create an account at Hotel.inc.

When a user clicks a link "show Airline.inc frequent flyer discounts" at Hotel.inc, the user is redirected to Airline.inc for authentication. After authenticating, the Airline.inc will only send a temporary transient pseudonym about the user back to Hotel.inc. No personal information is sent from Airline.inc to Hotel.inc.

3.4 Government-to-Citizen

3.4.1 Background for scenario

Governments are increasingly offering services for citizens on internet. In Finland examples of such services are Tax Administration's Tax Card Online and lomake.fi. Karine Barzilai-Nahon et al note that G2C and B2C scenarios are rather similar on both technological and administrative level. However they differ on some key points, one of them being the higher level of attention to security and privacy issues. While B2C services are focusing on avoiding customer retaliation and giving users control and clear policies on information disclosure, G2C services on the other hand do not focus on individual users, but instead are more focused on limiting unauthorized information disclosure in general [BNS07].

Camp [Cam04b] has analyzed the identity needs of G2C services, coming to conclusion that a unique trackable identifier is needed on services targeted for citizens. The Danish National IT & Telecom Agency has come to the same conclusion on [DK-SAML] noting that most G2C service providers will need to know users' Danish civil registry number (CPR) or Danish electronic identity number (OCES). This is in contrast to B2C services, where user privacy can usually be protected by using opaque pseudonym identifiers, as presented in [Liberty-overview, chapter 2].

In Finland there are two such unique trackable identifiers: PIN and FINUID. There is a number of IDPs in Finland offering either one or both of these identifiers, including TUPAS offered by banks and described in [TUPAS] and national certificate service. So the IDPs offering unique trackable identifiers are already available, but the problem is that none of these IDPs has a very large portion of the identity market on their own. For example most people have a bank account only in one bank and many still do not have a national identity certificate. Having each SP trust each of these IDPs separately means making and maintaining a lot different federation connections and possibly handling many different federation protocols.

This opens a market for an IDP proxy that offers authentication information from different IDPs so that SPs only need to directly trust and manage connection to this central IDP proxy. Examples of such IDP proxies in Finland are the G2C identity providers tunnistus.fi and Vetuma. They offer authentication information from a number of banks and also from the national certificate service and they both use a identity profile that delivers a user's PIN to the SP.

The Danish National IT & Telecom Agency notes that although most government organizations probably use Danish civil registry number or electric identity number for linking users' identities, the architecture should not mandate this. Therefore [DK-SAML] defines a "Persistent Pseudonym Attribute Profile" for creating identity federations with an opaque persistent pseudonym instead of using the Danish equivalents of Finnish PIN and FINUID. This profile is similar to the pseudonym profiles in B2C services (see previous chapter 3.3), but support for transient identifiers and federation termination protocol has been left out of scope.

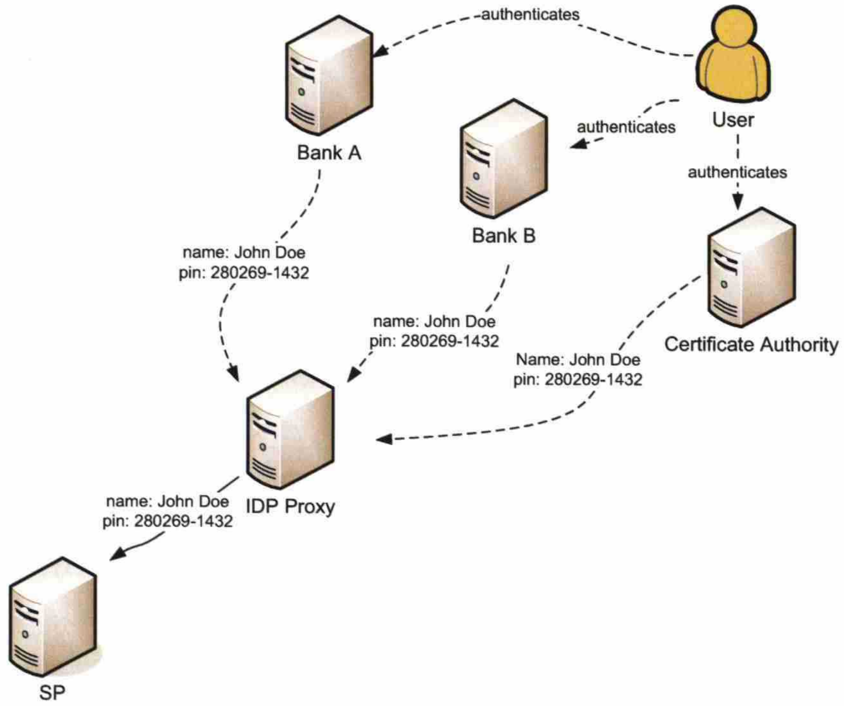

Figure 3.4: G2C authentication scenario. User may authenticate through any of the first level IDPs, but the SP only needs to communicate with the IDP proxy. As the different first level IDPs might use different protocols, this simplifies the authentication procedure for the SP considerably.

3.4.2 Use cases

Use case G2C.1: Proxied identity federation with globally unique identifier Tax administration offers a service for ordering tax card online. When user arrives to service, he/she is redirected to an IDP proxy for authentication. The service needs to know the user's PIN and name

The proxy IDP itself does not have the capability for authenticating users, but instead offers user a choice of other federated identity providers, including online banks and national certificate service. These services send the user's identity information in different formats and protocols. The attribute names might for example be different. The IDP converts the identity information from these different federated identity providers to the format expected by SP.

Use case G2C.2: Proxied identity federation with persistent pseudonym Portal.gov offers various services to citizens, which require the users to register and authenticate using strong authentication, but none of the services needs to know the users' PIN. Also some of the users of the SP are immigrants and foreign visitors who don't even have a PIN. Therefore the service prefers to use persistent pseudonyms for identity federation. This way it can accomplish strong authentication while protecting the privacy of the users and allowing the use of authentication methods that don't provide the users' PIN. For example the, TUPAS protocol supports authentication of foreign users who have a Finnish bank account, but don't have a PIN. [TUPAS]

3.5 Evaluation criteria

In order to evaluate the implementation of these use cases in UAS we need objective criteria based on the needs of the stakeholders. The needs of the two key stakeholders, the user and the SP, differ from scenario to another, but some issues like the ease of use and simplicity of configuration are universal requirements in all scenarios.

The evaluation criteria are divided in two groups: the needs of the user and the needs of the SP.

3.5.1 User criteria

Each of the presented scenarios have slightly different user requirements. In the B2E use case, a user just wants a simple and effortless SSO experience, where he is ideally not even aware of an identity federation taking place. On the other hand, in the B2C use cases the user privacy is more important and account federation needs explicit user consent. Therefore the first and most important evaluation criteria is based on the scenario specific use case descriptions:

Does the implementation fulfill the user requirements stated in the scenario and use case descriptions? An implementation that does not fulfill the stated user requirements will be inadequate for the purpose and in that case the evaluated UAS version can not or should not be used in this scenario.

Is the implementation easy and usable for the user? How many dialogs is the user showed during the authentication procedure? How difficult questions is the user faced with? Is the user required to understand how the federation process works? Can the user make a mistake during the federation? Can the federation end up in a dead end due to user mistake?

3.5.2 SP criteria

As with the user criteria, we will start with the scenario specific requirements. Beyond those basic requirements, the SP would prefer to do as little configuration management and setup as possible. These issues form the basis of the second part of SP criteria below.

Does the implementation fulfill the SP requirements stated in scenario and use case descriptions? An implementation that does not fulfill the stated SP requirements will be inadequate for the purpose and in that case the evaluated UAS version can not or should not be used in this scenario.

Is the implementation easy and scalable for the SP? Does the attribute namespace and user account information that the SP receive from the IDP remain the same regardless of the authentication method used? Does the attribute namespace stay the same regardless of the identity domain from which the user originally is federated? Are the network setup and the networks from which users can connect the SP somehow limited by the implementation? When a new IDP is added to to federation network, do all the SPs in the identity domain need to be configured individually to trust the new IDP or can some of the configuration work be automated or offloaded to the local IDP at that identity domain?

3.5.3 Evaluation methodology

Some of the evaluation criteria are rather subjective questions, like the ease of use for the user. Ideally the implementation could be tested in a usability laboratory and the SP criteria could be evaluated by collecting feedback from clients. However, in the scope of this thesis we don't have the resources to do such an objective evaluation. In chapter 5, we will evaluate the implementation of the use cases based on our analysis in that chapter and chapter 4, using these criteria as guidelines on which issues to focus on. Although not purely objective and measurable in absolute numbers, we think that the evaluation will be illustrative of the general strengths and weaknesses of the implementation.

Chapter 4

Handling identity management scenarios in UAS

In this chapter we answer the first two research questions "Analyze how UAS can handle the use cases presented in each of the given scenarios" and "Recommend how UAS could handle the use cases better" based on our analysis of UAS version 4.1. We go through all the scenarios and on each use case we present how the information transfer and federation were handled by the UAS version analyzed. For some use cases, this previous UAS version was found inadequate. Recommended changes are given at the end of each scenario, thus answering the second research question of how the system could be improved.

4.1 Business-to-employee use cases

4.1.1 Introduction

B2E use cases were introduced in chapter 3.1 and they are characterized by the fact that user identities stay inside the same identity domain and the SP only needs to know the LDAP username of the employee.

4.1.2 Use case B2E.1

In use case B2E.1 Retailer.inc offers a web based application to its employees, so that the service may be used from intranet, with the single sign-on session of the operating system. This use case can be accomplished with UAS using SAML authentication request protocol where Windows domain login session is used for authenticating the user and then passing this LDAP username in assertion to the SP.

Figure 4.1 depicts the message flow during authentication:

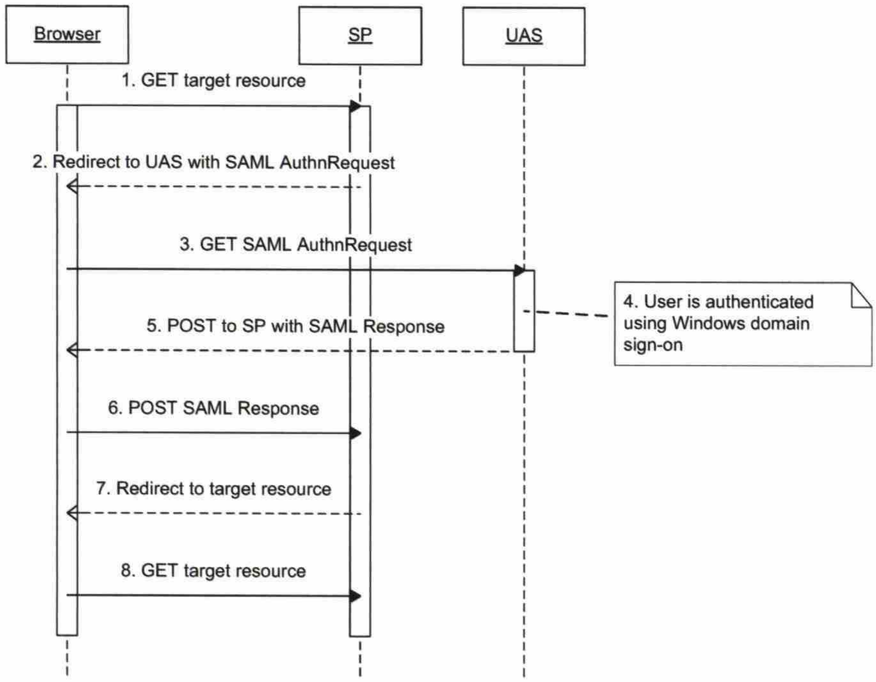
Browser
SP
UAS
1. GET target resource
2. Redirect to UAS with SAML AuthnRequest
3. GET SAML AuthnRequest
4. User is authenticated using Windows domain sign-on
5. POST to SP with SAML Response
6. POST SAML Response
7. Redirect to target resource
8. GET target resource

Figure 4.1: B2E authentication use case.

1. User request a resource at the SP that requires user authentication
2. The SP redirects the user to the UAS with a SAML authentication request
3. Browser forwards the authentication request to UAS
4. UAS validates the authentication request, and if user does not already have a SSO session, authenticates the user using integrated Windows authentication, based on kerberos single sign-on as described in [Windows-SSO]. This makes the authentication automatic and invisible to the user.
5. UAS finds the user account from local LDAP directory creating a directory identity as described in [Kä08, Chapter 5.3.3] and sends the LDAP directory name of the user to the SP in a SAML response message. Also UAS can send attributes and LDAP group information about the user if needed.
6. Browser sends the SAML response to the SP. User can access the resource.

An example Extensible Markup Language (XML) listing of the SAML response sent by the UAS can be seen in listing 1 in appendix 6.2. Listing 4.1 shows the subject element from that full SAML response. Note that the UAS uses name id format "X509SubjectName" in the "NameId" element, when the user account can be found from a local LDAP directory. The attribute "NameQualifier" is used to specify the user directory. This information can be used by the SP to find the same user account from the local user directory if needed.

Listing 4.1: A partial XML listing for B2E scenario. Note that the name format is of type "X509SubjectName" and the name qualifier specifies an LDAP url of the directory from which the user is found.

```
--- cut ---
  <saml:Subject>
   <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:
      nameid-format:X509SubjectName" NameQualifier="
      ldap://retailer.inc/dc=directory,dc=retailer,dc=
      inc">cn=jdoe,ou=users,dc=directory,dc=retailer,dc
      =inc</saml:NameID>
   <saml:SubjectConfirmation Method="urn:oasis:names:tc
      :SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData Address
       ="195.197.205.34" InResponseTo="
       _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
       NotOnOrAfter="2010-03-30T10:15:39.595Z"
       Recipient="https://retailer.inc/internal/spsso/
       saml2/AssertionConsumerService"/>
   </saml:SubjectConfirmation>
  </saml:Subject>
--- cut ---
```

4.1.3 Use case B2E.2

In this use case employees of Retailer.inc can use the same service as in use case B2E.1 also from an internet terminal outside of the corporate network, this time strong authentication is required. As in the previous use case the LDAP user account name of the employee is sent to the SP, thus making it irrelevant to the SP whether user authenticated using operating system's single sign-on from intranet or strong authentication from public internet.

The login sequence is the same as in Figure 4.1, except for step 4, where User is authenticated. UAS supports a number of strong authentication methods that can be used to authenticate users from an LDAP directory.

These include such as ETSI MSS [ETSI-MSS] and One-Time Password (OTP) authentication. Depending on the authentication method that is used, either the authentication method is authorized to assert users in the specified ldap directory [Kä08, Chapter 5.3.3], or the user account is searched from LDAP directory based on identity attributes using account mapping as described in [Kä08, Chapter 6.3.1]. Both of these methods for finding the user from LDAP directory will result in the same directory user identity as in use case B2E.1.

4.1.4 Handling of business-to-employee use cases

Both use cases of this scenario were handled well by the analyzed UAS version and no development needs were identified.

4.2 Business-to-business use cases

4.2.1 Introduction

B2B use case was introduced in chapter 3.2 and in this use case the SP is interested in the company the user is coming from and role the user has there.

4.2.2 Direct federation versus central identity provider proxy

In B2B use cases users from different partners' identity domains can use the SPs at Importer.inc. These SPs can all be configured to directly trust and accept assertions from various different IDPs. We call this model direct federation. This model is seen in figure 4.2.

The UAS version analyzed also supported another model where instead of each SP communicating with partner IDPs directly, the SPs at Importer.inc only communicate with the Importer.inc's IDP and that local IDP handles all federations with partners. See figure 4.3. We call this model central identity provider proxy. Linden has analyzed the need for these two models also in federations between identity domains, calling them decentralized and centralized technical setups of federation. He separates this technical setup from the contractual setup of the federation [Lin09, Chapter 5.3.2 and Figure 17], which confirms our analysis that these models are contractually interchangeable and choice between them should be done based on technical and lower level operational decisions. These two models are compared and analyzed further below.

Direct federation

Direct federation model is presented in figure 4.2. The figure shows identity information transfer for a user who has role "sales" in different services at Importer.inc. If we look at SP A in the figure we see that the local IDP sends roles in local namespace and SP A gets the role name "sales". However IDP A sends a slightly different name for the role "Sales Manager", which SP A needs to map to the role "sales". This means that SP A needs to do an IDP specific role name mapping for each IDP it accepts assertions from.

Identity information sent by IDP B has again slightly different name for the role, but also the assertion includes an additional role "admin". However, this role could possibly be a role that SP A would allow only Importer.inc's local administrators to use. This brings up another IDP specific rule: role filtering.

Overall each SP needs mapping and filtering rules for each partner IDP. When the number of SPs at Importer.inc identity domain is denominated by n and the number of IDPs by m the number of federation role mapping and filtering rulesets to keep up becomes $n \cdot m$.

Compared to identity provider proxying, which is presented next, this high decentralization has an advantage in availability and performance, because there are no single points of failure or bottlenecks.

Central identity provider proxy

Central identity provider proxy model is shown in figure 4.3. As in the previous example this figure shows the identity information transfer for a user who has the role "sales" in different services at Importer.inc. Instead of each SP communicating directly with each IDP, the SP only communicates with Importer.inc's local IDP and authentication requests and responses are proxied through it. Now the role mapping and filtering can be done at the Importer.inc's IDP.

In this model the Importer.inc's IDP needs to have mapping and filtering rules for each partner IDP and then another set of rules for configuring what roles to send to each SP. When the number of SPs is denominated by n and the number of IDPs by m the number of federation role mapping and filtering rulesets to keep up becomes $n + m$.

This centralized approach has the disadvantage that it creates a single point of failure that could compromise all services at Importer.inc's extranet. Also if the usage loads become very high, the local IDP might become a performance bottleneck.

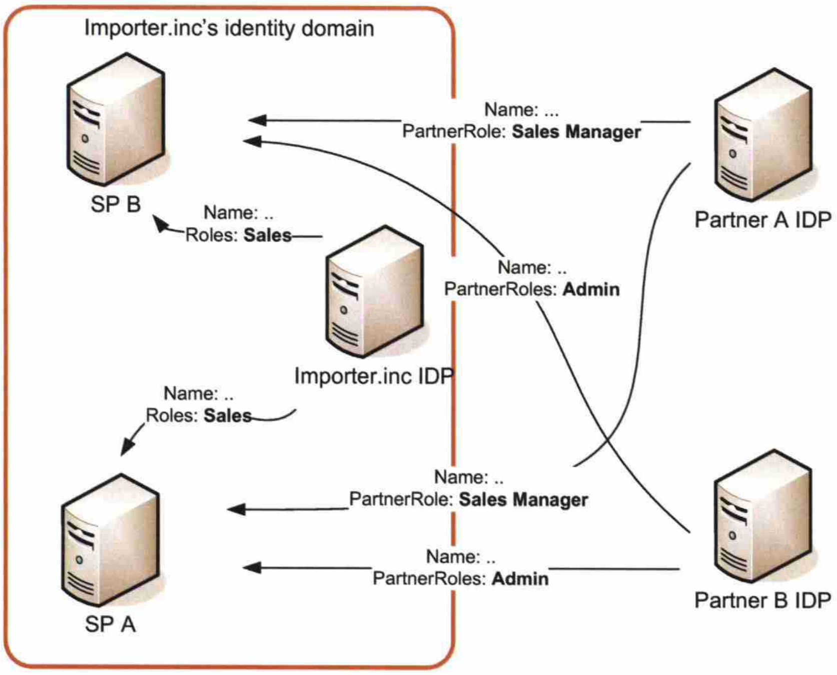

Figure 4.2: B2B authentication use case with direct federation, where each SP trusts all IDPs directly. Arrows present an example of identity information transfer for a user that has role "sales" in both SPs. Note that each IDP sends the roles with slightly different names and also that IDP B sends an additional role "admin".

Model	Advantages	Disadvantages
Direct Federation	better availability through decentralization	complexity $n \cdot m$
Central IDP proxy	lower complexity $n + m$	Centralization leads to single point of failure and a possible bottleneck

Table 4.1: Advantages and disadvantages of direct federation and identity provider proxying

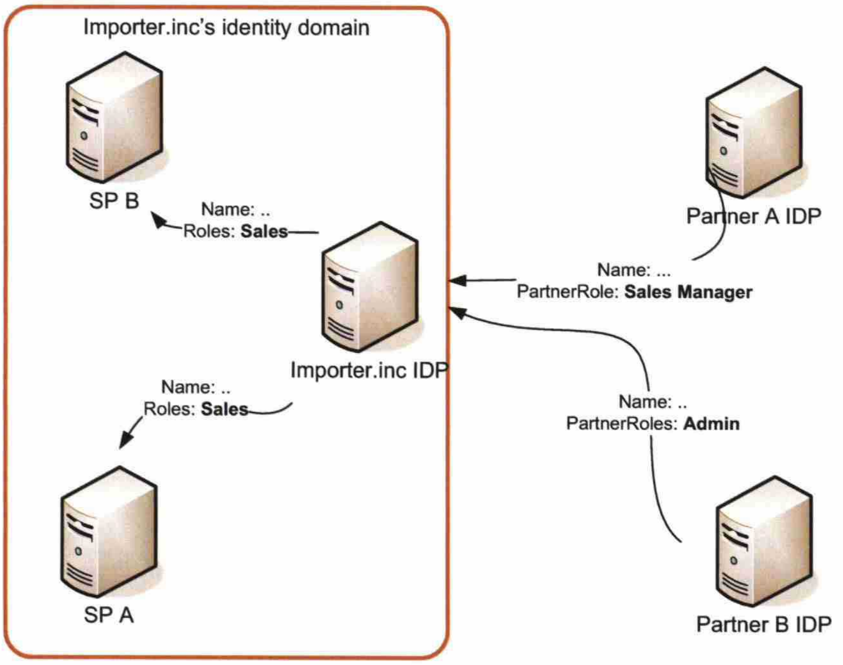

Figure 4.3: The central identity provider proxy model model of handling the B2B authentication use case. Instead of each SP trusting each IDP directly, all SPs only trust the local IDP which will forward authentication requests to partner IDPs as needed. In this model local IDP can make role name mapping and filtering for SPs.

Conclusion

Whether to use direct federations between SPs and IDPs or to use a central identity provider proxy comes down to weighing the advantages and disadvantages of both models in the scenario at hand. Especially when the number of SPs and IDPs becomes larger, the additional work in adding and removing IDPs and SPs increases in the direct federation model, but remains constant in the proxy model.

As the local IDP already might be needed to access and manage the services locally, it already is such a critical part of the local infrastructure, that whether the partners can or can not access the services while it is down in our opinion should not be a major factor in deciding the federation mode used. The performance bottleneck might be a problem in some high traffic services, but this has to be decided case by case. UAS supports clustering which can help to improve throughput and availability. Performance testing is left out of scope for this analysis.

Despite the possible performance shortcomings, the identity provider proxy model was chosen for analysis because it is less complex to manage and easier to add federation links.

4.2.3 Use case B2B.1

In use case B2B.1 Importer.inc offers a web application to its corporate customers, so that the services the user can access are based on the company the user is coming from and the role the user has there. This use case has an additional goal of keeping the list of trusted partner IDPs confidential, if possible.

We could see UAS in two different roles here. First UAS can handle the role of the local IDP through which all federations are proxied and second UAS can be in role of a partner IDP. The role of partner IDP is not essentially different from the role local IDP in B2E use cases (see previous chapter.) To partner IDP the Importer.inc's IDP looks like any other SP. The fact that it is in another identity domain does not matter from its point of view. The only additional requirement to those from use cases B2E.1 and B2E.2 is that the IDP has to send user roles with authentication. Käpynen has covered this in [Kä08, Chapter jotain], documenting the implementation of these features. Therefore we will focus our analysis on the role of local central IDP proxy.

Figure 4.4 depicts the message flow during authentication:

1. User request a resource at the SP that requires user authentication

2. The SP redirects the user to the UAS with a SAML authentication request

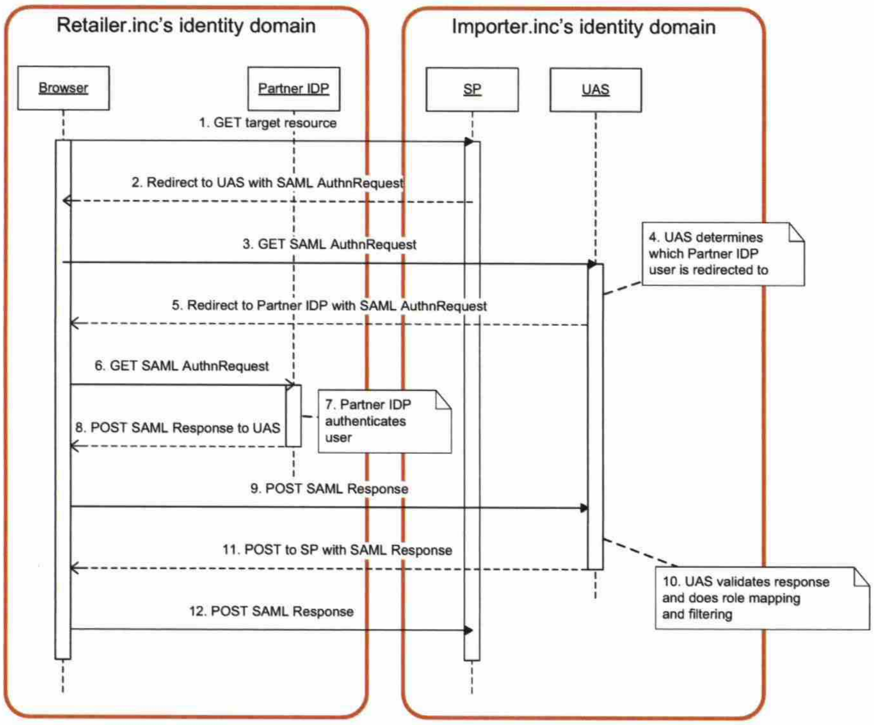
Retailer.inc's identity domain
Importer.inc's identity domain
Browser
Partner IDP
SP
UAS
1. GET target resource
2. Redirect to UAS with SAML AuthnRequest
3. GET SAML AuthnRequest
4. UAS determines which Partner IDP user is redirected to
5. Redirect to Partner IDP with SAML AuthnRequest
6. GET SAML AuthnRequest
7. Partner IDP authenticates user
8. POST SAML Response to UAS
9. POST SAML Response
11. POST to SP with SAML Response
10. UAS validates response and does role mapping and filtering
12. POST SAML Response

Figure 4.4: B2B solicited authentication use case.

3. Browser forwards the authentication request to UAS

4. UAS validates the authentication request, and if the user does not already have a SSO session, has to forward the user to a partner IDP for authentication. The problem here is how UAS is going to choose the correct partner IDP. One possibility is to show a list of all partner IDPs, but this would violate the confidentiality requirements in the use case description. This is discussed in more detail in chapter 4.2.4.

5. UAS redirect the user to the partner IDP with a new SAML authentication request.

6. Browser forwards the authentication request to the partner IDP.

7. The partner IDP verifies the authentication request and authenticates the user.

8. The partner IDP creates a SAML response message for UAS and includes the user's username and roles to the assertion.

9. Browser sends the SAML response to the UAS.

10. UAS validates the SAML response and, if needed, does role name mapping and filtering to convert the roles to the local namespace.

11. UAS creates a response for the SP.

12. Browser sends the SAML response to the SP. User can access the resource.

Overall, the authentication procedure works as described in the use case, except for choosing the partner IDP in step 4. This can be problematic, especially if the list of trusted partner IDPs is confidential. This is discussed in more detail in next chapter.

4.2.4 Choosing correct federation partner

The message flow during B2B authentication is shown in figure 4.4. In step 4 the UAS has to decide which partner IDP it is going to send the user to. If there is only one possible partner IDP or if the SP can tell in the authentication request which IDP to use, then user can be forwarded automatically. If there are more than one partner IDPs, the UAS has a number choices it could do:

1. Show user a list of trusted federation partners to choose from

2. Force partner organizations' users to login using unsolicited federation.

3. Use OpenID-like model where user types in an Uniform Resource Locator (URL) of their IDP.

4. Try to somehow deduce from IP-address, user cookies or other methods what the user's IDP could be.

The first option of showing a list of partner IDPs obviously works always, but has no confidentiality. The second option of using so called unsolicited federation is described later in chapter 4.2.5. Unsolicited federation however has the disadvantage of forcing users to start the authentication procedure themselves, before entering the service at SP. This can be a usability problem, as the users can not for example bookmark pages at service and then return there later, unless they remember to first start the unsolicited authentication before following the bookmark.

The third option is using a model similar to what OpenID does. OpenID solves the discovery problem by asking the user to write an URL that reveals the user's IDP [OpenID]. This model solves the privacy problem by keeping the list of trusted providers hidden, but it has been criticized because of its inherent vulnerability to phishing attacks [Hod07].

The fourth and last option is that of trying to automatically deduce what the user's IDP is. There are a number of different methods for this. OASIS has defined Common Domain Cookie (CDC) as cookie based method of remembering users' previously used IDP [SAML-profiles]. This method works if all the IDPs support CDC and the user has not cleared cookies from his browser since the last authentication to local IDP. This method will not work every time. If the user clears the cookies, or uses a new browser and has not authenticated to local IDP before trying to use service, the cookie will not be set and UAS will have to use some other method to find the correct IDP.

Another method of choosing the federation partner is based on users IP-address. This works if the partners' have a well known and limited number of IP-addresses that the users are coming from. However, if the users are connecting the service from outside their company network, this method will not work.

There might also be other heuristic methods that are not mentioned above and it is possible to use a combination of these methods as well. As it seems that there is no single solution that would work for every scenario, the local IDP should support configuring and customizing this step.

Of course this problem will not arise if there is only one possible partner IDP or if the SP is able to request the correct partner IDP from UAS. This could be achieved for example if the SP has a different URL path for each trusted partner IDP. This is not possible in all situations however.

Of these methods the analyzed UAS version supported listing all or a subset of partner IDPs in a list user can choose from, unsolicited authentication, and automatically choosing the IDP based on client's IP-address. These can handle many federation scenarios quite well, but clearly not all and adding new methods of choosing the IDP could not be easily done as customer specific customization, as this is part of the core of the UAS.

OASIS has defined also another protocol for choosing the IDP called Discovery Profile, where the IDP instead of choosing the federation partner redirects the user to a discovery service with a discovery request [SAML-discovery]. The discovery service will choose the partner IDP and send the user back to the IDP with the chosen partner in response. This of course does not solve the problem of choosing the correct partner IDP, but merely delegates it to discovery service. However, this allows making the possible customizations that are needed only to the discovery service and leave the IDP uncustomized.

4.2.5 Use case B2B.1 with unsolicited federation

Use case B2B.1 can also be configured to use unsolicited federation. Unsolicited federation is defined in SAML specification as a authentication response message that is sent unsolicited, that is without a request from the receiver of the response [SAML-profiles, Chapter 4.1.5]. With unsolicited federation, the problem of choosing the correct partner IDP at the Importer.inc's services is avoided.

Figure 4.5 depicts the message flow during unsolicited authentication:

1. User launches the unsolicited authentication from his local IDP, specifying the the Importer.inc's service that he wants to access

2. The partner IDP authenticates the user

3. The partner IDP creates an unsolicited SAML response message for UAS and includes the user's username and roles in the assertion. Also, included is the URL of the Importer.inc's service the user want's to access.

4. Browser sends the unsolicited SAML response to the UAS.

5. UAS validates the SAML response and, if needed, does role name mapping and filtering to convert the roles to local namespace. Also, UAS checks that the service where user should be redirected is trusted local service.

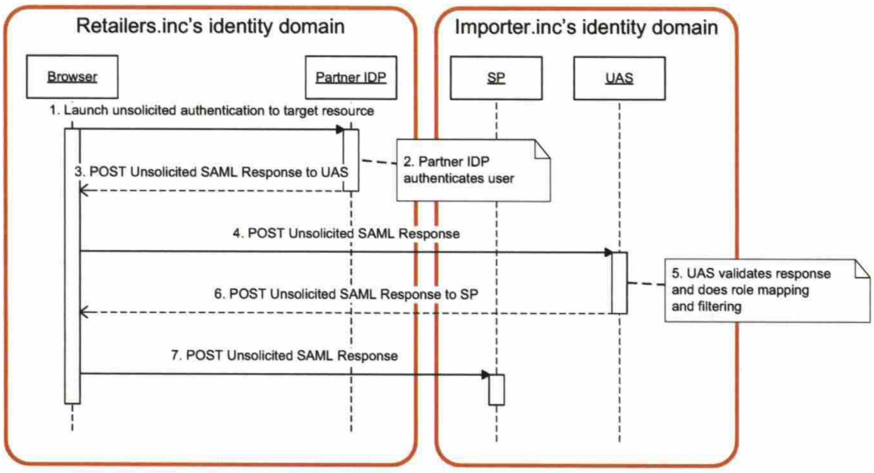
Retailers.inc's identity domain
Importer.inc's identity domain
Browser
Partner IDP
SP
UAS
1. Launch unsolicited authentication to target resource
2. Partner IDP authenticates user
3. POST Unsolicited SAML Response to UAS
4. POST Unsolicited SAML Response
5. UAS validates response and does role mapping and filtering
6. POST Unsolicited SAML Response to SP
7. POST Unsolicited SAML Response

Figure 4.5: B2B unsolicited authentication use case.

6. UAS creates a response for the SP.
7. Browser sends the SAML response to the SP. User can access the resource.

Besides avoiding the problem of choosing the correct federation partner, unsolicited federation has also the advantage of having fewer redirects. Normal solicited federation in figure 4.4 needs 14 steps to finish the authentication and might need even more in step 4 where the user might have to be redirected to some external service to choose the correct IDP. In contrast the unsolicited federation in figure 4.5 only needs 9 steps, of which only the authentication in step 2 might require user interaction.

Nevertheless unsolicited federation has a number of problems. First the user must choose to start the unsolicited federation process from their local IDP or portal. If the user tries to access the target service directly without an existing session, the authentication either has to fallback to normal solicited model or fail. Other problem is that there is no standard way of specifying what is the target URL that the user want to access. The SAML specification only defines a free text relay state field that can be sent along the SAML response. The UAS version analyzed supported passing a target URL in the relay state field, but it is unknown if this non-standard functionality is supported among other IDP technology providers.

4.2.6 Handling of business-to-business use case

The analyzed UAS version handled the use case well, except for the problems with the protecting privacy of partner organizations. Analysis in chapter 4.2.4 suggests a number of solutions, but none of them seems be a catch all solution for all B2B scenarios.

The analysis mostly focused on the role of UAS as a local IDP proxy, but in the role of partner IDP, choosing the federation partner with UAS would be smoother in some cases if it had supported common domain cookies. As for the role of IDP proxy, we suggested implementing support for IDP discovery profile, so that UAS could send the user to an external discovery service to choose to the correct federation partner. This would allow possible customer specific customizations or existing 3rd party discovery services to be used without changes to the core UAS authentication process.

4.3 Business-to-consumer use cases

4.3.1 Introduction

The B2C use cases in chapter 3.3 concentrate on protecting the customers' privacy with the help of opaque pseudonyms. Use cases B2C.1 and B2C.2 use SP specific persistent pseudonyms, which stay the same across different SSO sessions, whereas use case B2C.3 uses transient pseudonym which is different for each SP and each session.

4.3.2 Handling of business-to-consumer use cases

UAS version 4.1 did not support either kind of pseudonyms and therefore none of the use cases in this scenario could be completed in a satisfactory manner. The first and third use cases could be completed without the use of pseudonyms, using a normal authentication procedure with for example LDAP distinguished name instead of an pseudonym in assertion as in B2E use cases, but this leaves users' privacy completely unprotected.

In order to support the use cases of this scenario, we planned adding support for creating SP specific persistent pseudonyms and session specific transient pseudonyms for each user.

To complete the use case B2C.2 UAS also needed to add support for removing the persistent pseudonym. The SAML technical overview presents a use case describing this procedure, using SAML name id management protocol termination message [SAML-tech-overview, Chapter 5.4.5]. As the analyzed UAS version did not support this protocol, support for that would have to implemented as well.

Besides the missing features mentioned above, the use cases did not have any other new requirements that the previous UAS version could not handle.

Overall the message flow in use cases B2C.1 and B2C.3 is similar to that in use case B2E.1 in figure 4.1.

4.4 Government-to-Citizen use cases

4.4.1 Introduction

In G2C scenario, described in chapter 3.4, the IDP acts as an proxy between the SPs and different first level IDPs offering user authentication. The information sent to the SP should remain the same, or in other words, to use the same attribute namespace, regardless of the authentication service used.

4.4.2 Use case G2C.1

In the first use case users of the SP are authenticated using different banks and national certificate service, but regardless of the actual authentication service used, the central IDP proxy should send the user's PIN and name in attributes with the same names.

The message flow during authentication is similar to the B2B proxied authentication flow in figure 4.4, except that steps 5-9, where the IDP proxy communicates with the first level IDP, might use some other protocol than SAML. For example in Finland the banks that act as first level IDPs use a proprietary TUPAS protocol [TUPAS]. Besides the need to support additional protocols, this use case does not bring any new requirements compared to use case B2B.1. Attribute mapping and filtering were already covered by that use case. UAS supports the protocols used by Finnish banks and national certificate service and it is used in this role as an IDP proxy in tunnistus.fi, the G2C IDP with the highest traffic in Finland.

An example the SAML response sent by the UAS can be seen in listing 3 in appendix 6.2. This example is from testi.tunnistus.fi, where UAS is used an authentication proxy as specified by this use case. Listing 4.2 shows a snippet from that full SAML response. Note that the UAS uses name id format "unspecified'" in the "NameId" element, when the user account is not found from an LDAP directory. The SP should use the attributes "name.ref" and "tfi.custname" to find the user's name and "id.ref" and "tfi.CUSTID" to find the user's PIN.

Listing 4.2: A partial XML listing for G2C scenario. Note that the name format is of type "unspecified" and attributes contain user's name and PIN.

```
-- cut --
  <saml:Subject>
```

```
   <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:
      nameid-format:unspecified">012345-678D</saml:
      NameID>
   <saml:SubjectConfirmation Method="urn:oasis:names:tc
      :SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData Address
       ="195.197.205.34" InResponseTo="
       _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
       NotOnOrAfter="2010-03-30T10:15:39.595Z"
       Recipient="https://example.com/portal/spsso/
       saml2/AssertionConsumerService"/>
   </saml:SubjectConfirmation>
  </saml:Subject>
-- cut --
   <saml:Attribute Name="tfi.CUSTID">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string
       ">012345-678D</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="id.ref">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       tfi.CUSTID</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="tfi.version">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       katso-1.1</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="name.ref">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       tfi.CUSTNAME</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="tfi.custname">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
```

```
        John Doe</saml:AttributeValue>
    </saml:Attribute>
--- cut ---
```

4.4.3 Use case G2C.2

In the second use case the users need to be identified using a persistent pseudonym instead of PIN. As noted in section 4.3, the analyzed UAS version did not support pseudonyms and this use case could not be completed.

4.4.4 Handling of government-to-citizen use cases

The first use case was handled well, but the second use case was lacking support for SP specific persistent pseudonyms. This requirement already came up in B2C use cases and implementing the support for them also will fulfill the needs of this scenario.

Chapter 5

Implementation of new features

In chapter 4 we analyzed how UAS version 4.1 handled the federation scenarios from chapter 3. In all scenarios except for B2E scenario we found some deficiencies in the way UAS handled them and for each some changes were suggested. Most of the changes were implemented to version 5 of UAS. This chapter answers the last research question "analyze how the implemented changes work in next version of UAS". We will go through all the scenarios where problems were identified and their use cases documenting and analyzing the how the use cases are supported now.

5.1 Business-to-Employee scenario

5.1.1 Introduction

As described in chapter 4.1.4, the previous UAS version already handled this use case in full accordance with the requirements of the scenario.

5.1.2 Evaluation of implementation

In this section we evaluate the implementation of this scenario from chapter 4.1 according to the criteria from chapter 3.5.

User's requirements The user's requirements in the B2E use cases were that the user can access the services preferably with the SSO session of the operating system. There were no privacy or user consent requirements. The scenario specific user requirements are therefore fulfilled.

The implementation is easy for the user, as the user is not shown a single dialog when in intranet as UAS uses the operating system SSO session to sing the user in automatically. When singing in from a remote terminal over the internet, the user is prompted for his authentication credentials, but no further dialogs are shown and the only mistakes the user can possibly do, are typing in wrong authentication credentials, such as the wrong one time password. As the authentication interface is offered by the local UAS it can instruct the user as needed.

SP's requirements In the scenario requirements, the SP wanted the user information in same format regardless of whether the user came from local intranet or from a remote terminal over the internet. This requirement is met by the implementation, as the UAS can be configured to send the user's local LDAP account name in both use cases.

The SP does not need to be aware of changes in authentication methods as the identity profile is not dependent on them. As this scenario is limited to use within one identity domain, the SP does not need to handle the addition of new IDPs.

Conclusion of evaluation The previous as well as the current UAS version handle the requirements of both stakeholders well.

5.2 Business-to-Business scenario

This scenario was handled quite well by the previous UAS version and analysis in chapter 4.2 only found possible improvements in the IDP discovery. We suggested implementing support for Common Domain Cookie (CDC) to improve the user experience in federation networks that support it and also Discovery Profile to make it possible to move the step of IDP choice away from the UAS and to an external discovery service.

CDC support was implemented to UAS. This allows better user experience in federation networks that support CDC, but does not alone help to increase the confidentiality of the system, as the CDC can not be trusted as the only method of choosing the user's IDP.

However, we did not implement support for Discovery Profile to UAS version 5. In those cases where the list of trusted partner IDPs has to be kept confidential, it is possible to use unsolicited federation, which is already supported. As noted in chapter 4.2.5, this approach is not without problems, but it does get the job done in most scenarios. Also as noted earlier, the Discovery Profile does not really solve the problem as much as it just delegates the problem to another service.

In the end, our analysis of previous UAS version and its handling of B2B use cases in chapter 4.2.5 is still relevant for version 5 of UAS, as adding support for CDC, did not essentially change how these use cases are supported.

5.2.1 Evaluation of implementation

In this section we evaluate the implementation of this scenario from chapter 4.2 according to the criteria from chapter 3.5.

User's requirements Beyond the requirement that the user can sign in to the partner SP using his local IDP, there were really no user requirements specific to the B2B scenario, as the user is presenting his company instead of himself. The basic requirements are therefore met, but the story gets more complicated with the ease of use.

As described above, choosing the correct federation partner can be problematic. One choice was to show the user a list of federation partner IDPs. This is not a particularly difficult choice as we can reasonably expect that the user knows which company he works for, but showing the list of federation partners was against the privacy requirements of the SP in this scenario.

Another option that keeps the list of partner IDPs secret, is using the user's IP address to determine the correct IDP. This option is very easy for the user, as he is not prompted at all when making the choice, but this is against the SP requirement of ease and flexibility of network setup, as this sets limitations on from which networks the user may login.

Last option is that of unsolicited federation, which fulfils all of the SPs requirements, but is against the requirements of the user, as now the user may end up in a dead end, when trying to access the SP directly, instead of logging in through his local IDP.

SP's requirements The SPs requirements were to get the user's roles, organization and some kind of a user specific unique id, while keeping the list of trusted partner IDPs secret. The other requirements were met, but the last one is met conditionally as described above in user's requirements.

While the choice of correct federation partner causes it's own problems, the other aspects of the implementation are quite easy for the SP. The configuration of new federation partners can be done centrally at the local IDP, which can also handle role filtering and mapping.

Conclusion of evaluation The use case is fairly straight forward to implement with the current UAS version, except when the federation partners need to be kept secret. There are many different ways of accomplishing that and they all fail some of the evaluation criteria. So fundamentally it comes down to weighing the advantages and disadvantages of each implementation strategy based on the specific requirements of the stakeholders.

5.3 Business-to-Consumer scenario

The B2C scenario and three relevant use cases were analyzed in chapter 4.3, noting that the support for pseudonyms and protocols for managing them were missing from the analyzed version 4.1 of UAS. The pseudonym support was thereafter implemented with the goal of also passing interoperability certification of Liberty Alliance. Liberty Alliances certification process is described in chapter 2.3.

The test case A in Liberty Alliance's test criteria [Liberty-Interop] includes test steps for both pseudonym creation and termination, thus covering the functionality for use cases B2C.1 and B2C.2. Test case G includes transient pseudonym handling and therefore describes how to implement use case B2C.3. However, these test cases also include many other new functionalities that the B2C scenario in chapter 3.3 does not require. For example test case A from Liberty test criteria document also includes name identifier management steps where the persistent pseudonym is changed. These functionalities were also implemented to UAS 5 to pass the certification, but are not documented here as they are not necessary to complete the use cases from chapter 3.3.

In B2C use cases UAS can be in two different roles. First, UAS can be the originating IDP for pseudonyms. This means that UAS has to be able to create temporary transient pseudonyms and persistent pseudonyms for each user and SP pair and to support termination requests for persistent pseudonyms. How UAS handles this role is analyzed in chapter 5.3.1.

Second possible role is that of an central IDP proxy (see chapter 4.2.2). To handle this role, a new tool called Federation Manager was created to supplement the functionality of UAS. This setup is analyzed in chapter 5.3.2.

5.3.1 UAS as an originating IDP

Use case B2C.1 requires that the SP can request for a persistent pseudonym as user identifier from the IDP and that on subsequent authentications the persistent pseudonym for that user stays the same. Use case B2C.2 extends this by requiring support for federation termination requests, so that SP can request the IDP not to send the same persistent pseudonym again for that user.

To handle the role of IDP, UAS needed support for pseudonym creation and termination. To enable configuring pseudonyms into use and to store them the data model of Ubilogin directory was extended to support defining a name identifier format mapping table for each SP. Figure 5.1 shows the configuration model using syntax from Unified Modelling Language (UML) [UML]. Each SP has a name identifier mapping entry, that defines

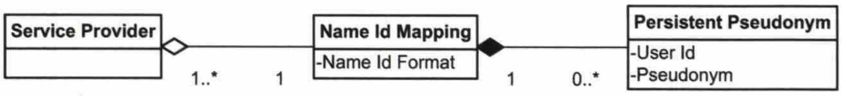

Figure 5.1: Name Identifier Mapping data model. Note that each SP has exactly one mapping attached to it by aggregation and each mapping may have a number of pseudonym entries that are exclusively attached to it by composition.

the used name identifier format. Supported format types include persistent pseudonyms, transient pseudonyms and Ubilogin internal formats. If the format is persistent, this mapping table has a list of user name and persistent pseudonym pairs. When a UAS creates a new authentication assertion for an SP that has been configured to use persistent name identifier format, it looks up if the authenticated user has an entry in this mapping table, and if not creates a new persistent pseudonym entry.

The SAML specification allows the SP to request a certain name identifier format to be used in the response. In theory this would make it possible to implement dynamic name identifier formatting, where the IDP would support sending any type of identifier the SP requests for. However this would be problematic in regards of user privacy; if the SP can request for the user's clear text username at will, why to use pseudonyms in the first place? Therefore, at least some restrictions on supported name identifier formats for each SP are needed.

The name identifier mapping configuration in UAS is implemented as static and exclusive, meaning that it can only be changed by the UAS administrator and one SP can have only one name identifier mapping configured to it at any given time. This means that if the SP requests a certain type of name identifier format, the UAS will compare it to the one configured for the SP and refuse the request if they do not match. Therefore, one SP will only ever get identifiers of one format. We considered this to be a reasonable restriction. None of the use cases in B2C scenario, nor the test cases in Liberty Alliances test criteria require the same SP to be able to request for identifiers of different formats. If a service needs to be able to receive identifiers of two different formats, for example persistent and transient pseudonyms, this can be configured to UAS by configuring the same service as two different SP entries. This however requires that the service can be configured to act as two different SPs with different SAML entity identifiers.

Figure 5.1 also shows that any number of SPs can be connected to same name identifier mapping table. The SAML specification allows the creation of affiliations, where a number of SPs will receive the same persistent pseudonym for the same user [SAML-core, Chapter 8.3.7]. In UAS this is configured by associating the same mapping table for two or more SPs, making this configuration also static and exclusive.

To enable transient pseudonyms for use case B2C.3 the administrator sets the name identifier format in mapping entry to "transient". In that case there are no user to pseudonym mapping entries saved to Ubilogin Directory, as these identifiers are only used for one session.

Use case B2C.1

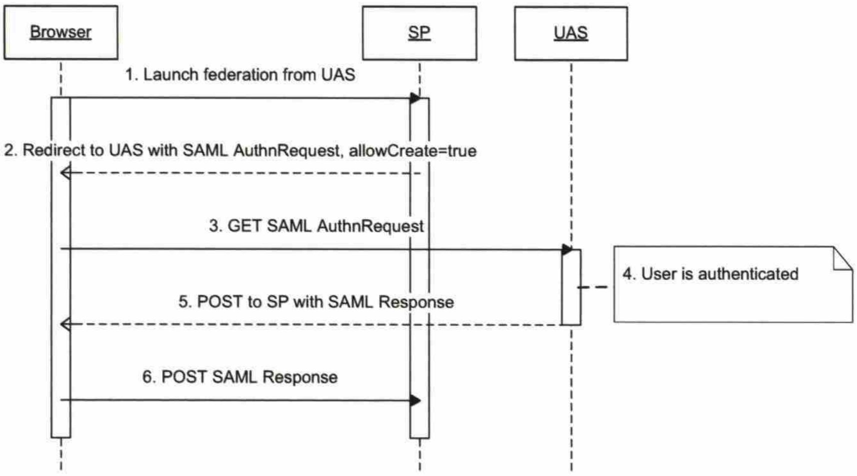
Browser
SP
UAS
1. Launch federation from UAS
2. Redirect to UAS with SAML AuthnRequest, allowCreate=true
3. GET SAML AuthnRequest
4. User is authenticated
5. POST to SP with SAML Response
6. POST SAML Response

Figure 5.2: B2C authentication use case.

As noted in the previous chapter, to use the persistent pseudonyms, the administrator has to configure the persistent name identifier format to Ubilogin Directory for that SP. Figure 5.2 shows the authentication sequence for use case B2C.1. The sequence is very much like the one in B2E use cases in figure 4.1. There are two noteworthy changes to authentication sequence. First in step 2, the SP has to define in authentication request whether the IDP is allowed to create a new persistent pseudonym for the user if one does not already exist. This "allowCreate" -attribute is defined in SAML core specification [SAML-core] and support for it is required in the Liberty certification program [Liberty-Interop, Test case A]. Secondly in step 5, UAS sends the user's persistent pseudonym back to SP, or if one does not already exist and SP allowed the creation of new one in step 2, creates a new persistent pseudonym and sends that back.

An example the SAML response sent by the UAS can be seen in listing 2 in appendix 6.2. Listing 5.1 shows a snippet from that full SAML response.

Note that the UAS uses name id format "persistent" in the "NameId" element, when persistent pseudonyms are used.

Listing 5.1: A partial XML listing for G2C scenario. Note that the name format is of type "persistent" and the value is a random string.

```
--- cut ---
  <saml:Subject>
   <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:
      nameid-format:persistent">
      ijKlJAaKAPrSHbqlbcKWu7JktcKY</saml:NameID>
   <saml:SubjectConfirmation Method="urn:oasis:names:tc
      :SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData Address
       ="195.197.205.34" InResponseTo="
       _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
       NotOnOrAfter="2010-03-30T10:15:39.595Z"
       Recipient="https://importer.inc/service/spsso/
       saml2/AssertionConsumerService"/>
   </saml:SubjectConfirmation>
  </saml:Subject>
--- cut ---
```

Use case B2C.2

In use case B2B.2 the created persistent pseudonym is removed. SAML specification calls this terminating the federation and has specified name identifier management protocol for it [SAML-core]. The Liberty test criteria has several test cases where federation is terminated using this protocol. Broadly these tests fall into two categories: the front channel management messages [Liberty-Interop, for example Test case A] and back channel management messages [Liberty-Interop, Test case B].

Figure 5.3 shows the message flow on front channel management request. As in previous authentication examples, the messages are routed through user's browser using Hypertext Transfer Protocol (HTTP) redirects or POSTs. In figure 5.4, we see back channel management message flow. Note that this message exchange is done directly from SP to IDP and user's browser is not involved. Either protocol could be used to terminate the persistent pseudonym, but as both were required by the Liberty Alliance's test cases, both were also implemented.

When UAS receives a termination request, it removes the relevant entry from the mapping table (see figure 5.1) and sends a response message back to requester. When there is only one SP using the name identifier map-

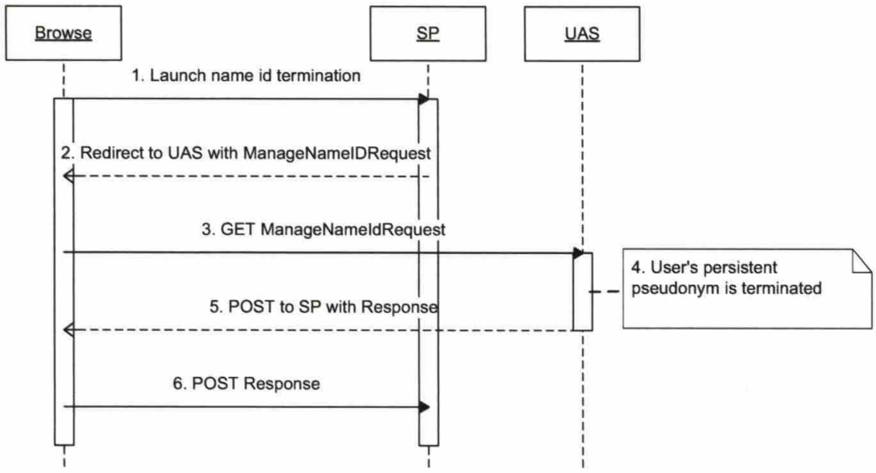
Browse
SP
UAS
1. Launch name id termination
2. Redirect to UAS with ManageNameIDRequest
3. GET ManageNameIdRequest
4. User's persistent pseudonym is terminated
5. POST to SP with Response
6. POST Response

Figure 5.3: Front channel name identifier management request for pseudonym termination. The message flow is similar to that of front channel authentication in for example figure 5.2.

ping this is a fairly straight forward matter, but as described earlier in chapter 5.3.1 there might be more than one SP in an affiliation that shares the same pseudonyms. The SAML specification does not mention what should be done when an IDP receives a name identifier management request from one affiliation member. The IDP could either simply remove the pseudonym and only respond back to the original requester or the IDP could inform all affiliation members by sending an equivalent name identifier management request to each other affiliation member. The specification does mention on logout requests, that these should be propagated to each SSO session member, so if we view an affiliation as kind of a "pseudonym session", it could be argued that the IDP should send the management message to each member.

However, comparing name identifier management to SSO session logout is problematic. In an active SSO session we can be fairly sure that all session member SPs are in fact active and reachable because the user has managed to login to them during the active session. When using front channel messaging knowing that all participants can be reached is very important, because any SP that fails to handle the request will disrupt the message flow and user's browser will stay at the SP showing possibly an error message instead of being redirected to the next SP and continuing the process.

With name identifier management messages it is not nearly as likely that all affiliation member SPs are running and ready to handle requests. In an affiliation with 20 member SPs, if even one of them is undergoing maintenance or connectivity problems a front channel request would fail leaving the user's browser at the URL where the chain broke. This problem can be avoided using back channel messaging. Back channel requests can fail without causing problems with the requests to other session members. On the other hand back channel requests will not work if the SP and IDP do not have direct connection or if one of them does not support SOAP requests [SOAP-part1, SOAP-part2]. The support for SOAP binding is not required for IDP-lite and SP-lite certification in Liberty Alliances test cases [Liberty-Interop].

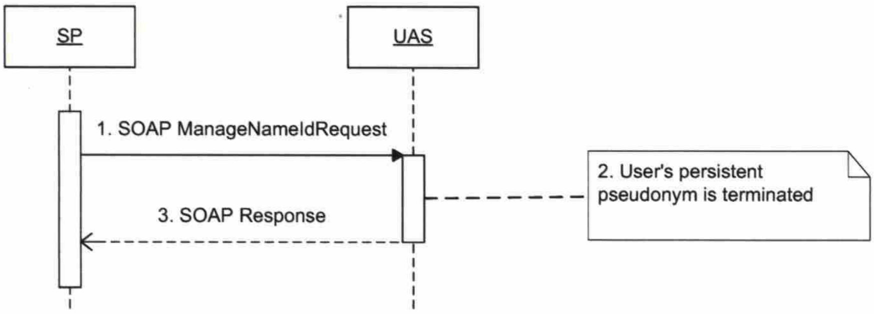
SP
UAS
1. SOAP ManageNameIdRequest
2. User's persistent pseudonym is terminated
3. SOAP Response

Figure 5.4: Back channel name identifier management request for pseudonym termination. Note that the user's browser is not involved in message exchange.

Because back channel binding can not be always used and front channel configuration does not allow propagating the message to all affiliation members due to the risk of possible disruptions in message flow, the name identifier management implementation in version 5 of UAS does not propagate any management requests to other affiliation members.

One way to think about affiliations is that because the SPs are sharing the same pseudonym they must be exchanging identity information already in back channel or maybe even using the same identity database. This implies that the affiliation members are responsible for sharing the management information as they see fit and UAS indeed does not need to propagate the management requests. Whether this is reasonable restriction on name identifier management handling remains to be seen as we get more experience with affiliation needs of real customers.

Use case B2C.3

Use case B2C.3 was simple authentication as seen in for example use case B2E with the exception that the assertion contains a new transient pseudonym for each authentication session. With the implementation of transient pseudonym support, the version 5 of UAS handles this use case as described in chapter 3. The authentication sequence follows the same pattern as in figure 4.1.

5.3.2 UAS as IDP proxy

We discussed the benefits and downsides of SPs directly communicating with all IDPs in chapter 4.2.2. If there is a large number of SPs in the same identity domain it can simplify the configuration management to use a local IDP proxy also in B2C use cases. There are two different subscenarios to this proxy model:

1. UAS works as a proxy without account linking capability. This means that UAS simply passes the pseudonyms as they are to the SP and it is the SP's responsibility to create account linkings and to provide authentication methods and user databases needed for linking accounts.

2. UAS will handle account linking and authenticates the users with local authentication methods as needed. This makes the federation invisible to the SPs and reduces the integration work.

UAS 5.0 works in the first model like the earlier version worked in B2B use cases as IDP proxy. This is described in chapter 4.2.3. However, this model leaves the account linking to the SPs, requiring a local user database and linking functionality for each SP.

The second option makes federation using pseudonyms considerably easier for the SPs. The account linking is done at the UAS level where the local user account are handled and the SP does not need a local user database. To support such a configuration a new UAS integrated tool was created called Federation Manager.

To partner IDPs Federation Manager is seen as the external interface of UAS and to local SPs it is completely invisible. But when examining how it works internally, it is somewhat more complicated. Federation Manager is configured to acts as two different SPs and as one IDP, as shown in figure 5.5. First it acts as an SP towards the external partner IDP and then to link the user to a local account from the UAS it acts as an SP to request for authentication. Finally when the accounts are linked it acts as an IDP to UAS.

Figure 5.5 shows the message flow during federation:

1. Partner IDP sends the user to Federation Manager's external interface with an unsolicited SAML response that includes a persistent pseudonym. Federation Manager then searches its database for that persistent pseudonym. If the pseudonym is found and the user is therefore already federated, Federation Manager goes to step four and

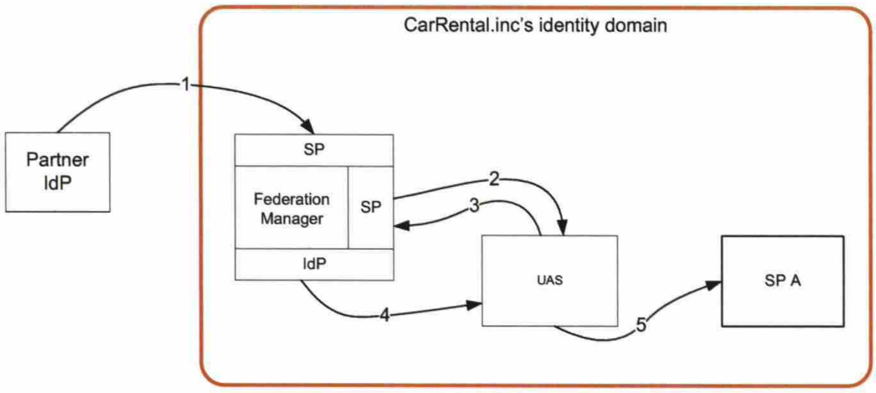

Figure 5.5: B2C authentication use case with federation manager.

sends an unsolicited response to UAS. If pseudonym is not found and user has not federated accounts yet Federation Manager sends an authentication request to UAS.

2. Because the user's pseudonym was not found in Federation Managers database, it sends a SAML authentication request to UAS to find a local account to which link the user's pseudonym.

3. UAS authenticates the user and sends back a response to Federation Manager. Federation Manager will save the identity information from the response and the original pseudonym to local database, so that next time the same user returns the account will be found automatically.

4. Federation Manager now acts as an IDP towards UAS sending an unsolicited SAML response with the user's local identity information. The original pseudonym received in step 1 is not part of the information sent to UAS.

5. UAS will create a SSO session for the user and send an unsolicited SAML response to the target service. SP will process the response and grant access to user with the local identity.

Federation manager does the account linking for the SPs so that their developers don't have to think about linking pseudonyms to local accounts. The SP does not need to know that the user signed in with a federated authentication using a pseudonym instead of authenticating locally. Both cases look the same from its point of view.

However, federation Manager does have two significant restrictions. First, it only supports unsolicited authentication and the local SP can not request for authentication. The process has to always start from the partner IDP. Secondly, the analyzed version of Federation Manager does not support name identifier management protocol, which was discussed in chapter 5.3.1 above. The need for automatic account termination was not seen as absolutely necessary for most federation scenarios. This is also acknowledged by Liberty Alliance as their IDP-lite and SP-lite certifications do not require support for name identifier management [Liberty-Interop]. System administrator can destroy the link between a pseudonym and user account by hand if needed.

Therefore with Federation Manager the use case B2C.1 can be handled in central IDP proxy role with account linking taken care of for the SPs, with the restriction that the federation process always has to start from the partner IDP. Because of the lack of support for name identifier management the use case B2C.2 could not be completed with the current version. If support for use case B2C.2 is required then instead of using Federation Manager the SPs need to communicate directly with the partner IDP.

5.3.3 Evaluation of implementation

In this section we evaluate the implementation of this scenario according to the criteria from chapter 3.5.

User's requirements User's requirements in B2C scenario were to be able to link account from one identity domain to another, while keeping the amount of identity information that is transferred between the domains to the minimum. Also the user requires an option to delete the link between the accounts.

In this chapter above we have analyzed different ways of implementing the use cases in this scenario. These basic needs of the user are met in all of them, with the exception of account link deletion when using federation manager.

The usability is reasonably good, although some extra dialogs are needed compared to previous scenarios. When not using federation manager, the user may access the target SP directly and the choice of correct IDP should not be too difficult (assuming that we can expect the user to know in which partner services he has an account). The act of linking the accounts requires the user to authenticate to both identity domains, but after that the SSO works.

When the federation manager is used, the setup and management becomes easier for the SP, but users are forced to use unsolicited federation and user initiated federation termination does not work. These can be major drawbacks on some situations, but as noted before, using federation manager is optional and all the user requirements of the scenario can be met without it.

SP's requirements The SP's basic requirements in this scenario are met by the current UAS version regardless whether the federation manager is used or not. But when it comes to configuration complexity there are significant differences.

When not using the Federation Manager, the SP must make the account linking and federation management by itself. This means keeping a local database of user pseudonyms and also providing the necessary user interfaces and authentication methods for the federation. On the other hand when using Federation Manager, all these issues are taken care of by the UAS. But as noted above, that might interfere with the user requirements.

Conclusion of evaluation The basic requirements of both stakeholders can be met, but with varying levels of user or SP satisfaction in other requirements. As with the B2B scenario earlier, there are different ways of implementing the use cases in this scenario and they all have their strengths and drawbacks. Weighing the stakeholders requirements needs to be done case by case in order to choose the correct approach.

5.4 Government-to-Citizen

5.4.1 Introduction

In G2C scenario the lack of pseudonym support in UAS 4.1 made use case G2C.2 impossible to handle in satisfactory manner. UAS 5.0 now handles this use case as described in chapter 5.3.1 (see Figure 5.2). Therefore UAS 5.0 now fully supports both use cases of this scenario.

5.4.2 Evaluation of implementation

In this section we evaluate the implementation of this scenario according to the criteria from chapter 3.5.

User's requirements The users in the G2C use cases don't have really scenario specific requirements except for the privacy requirements in use case G2C.2. As persistent pseudonyms are now fully supported, this requirement is met. Usability of the implementation is also very good, as the user is only prompted to choose which bank he is a customer to or to choose to use the certificates if he has the required certificate card and reader. This should be an easy decision for the user and if he chooses the wrong bank he can still backtrack to same view and choose correct first level IDP.

SP's requirements The scenario specific requirements of the SP were to always get the user's identity attributes in same namespace regardless of the first level IDP the user chose and in the second use case to be able to use persistent pseudonyms to identify users. Both of these requirements are met by the new UAS version.

The implementation is also easy to configure and scalable to the SP. As the central IDP proxy handles converting the identity attributes of new first level IDPs, adding new authentication services does not require any changes to the SP.

Conclusion of evaluation The previous UAS version did not handle the privacy requirements of use case G2C.2 but the new version fulfills all user and SP requirements well.

Chapter 6

Conclusions

In this thesis we evaluated how Ubilogin Authentication Server (UAS) handled a number of key federation scenarios. We suggested a number of improvements and based on those a new version on UAS was implemented and evaluated again. At the end of this chapter we discuss possible future developments.

6.1 Results of this thesis

A number of federation scenarios and relevant background to them were introduced in Chapter 3. These scenarios and use cases based on them were used as an evaluation criteria for analyzing how Ubilogin Authentication Server (UAS) handles different stakeholder needs. In chapter 4, we analyzed the previous UAS version, noting room for improvement in B2B, B2C and G2C scenarios. Also two different models for handling the external federation links, direct federation and central IDP proxy, were analyzed and as central proxy was found a useful model in many situations, use cases were also analyzed with UAS in this role.

In B2B scenario we found a possible problem in choosing the correct federation partner and especially problematic was making that choice in an environment where the list on federation partners is confidential. Although unsolicited federation was introduced as a solution to confidentiality problem, we suggested a more complete support for customization by implementing Common Domain Cookie (CDC) and Discovery Profile support. In both B2C and G2C use cases support for pseudonyms and their management was needed.

In chapter 5, we analyzed the latest version of UAS, where many of these improvements were implemented. Handling of B2B scenario had not changed, as only CDC had been implemented and the other one of the suggested improvements, the Discovery Profile, had been left out. Handling of pseudonyms in the new UAS version was discussed, also presenting some limitations on details that the SAML specification had left open. Analysis of UAS as the originating IDP found that both scenarios were handled well and all stakeholder requirements from the use cases were achieved. For the role of central IDP proxy, a new tool called Federation Manager was introduced and found to be useful in simplifying handling of the B2C scenario for the SPs, although it could not handle the second use case of federation termination. The latest version of UAS was also taken to Liberty Alliances interoperability testing, where the handling of pseudonyms was tested and successfully passed.

6.2 Future development

The scenarios chosen for this analysis were limited in order to keep the size of this work manageable. However especially Government to Government (G2G) scenario would require further analysis now that Virtu project [VIRTU] is advancing in Finland and introducing its own SAML profile [Virtu-SAML].

Also the analysis on B2B scenario left non RBAC applications out of scope and did not include use cases with account mapping. Analysis on different account mapping and also account provisioning use cases would give a much more realistic picture of B2B use cases in real world legacy systems.

Further improvements to UAS are also needed. Support for Discovery Profile in UAS is also still lacking and work on Federation Manager could be extended by analyzing whether support for name identifier management and solicited authentication could and should be added to it.

List of Figures

Bibliography

[507/1993]
Ministry of the Interior. Väestötietolaki, 1993. http://www.finlex.fi/fi/laki/ajantasa/1993/19930507, cited 6th May 2009.

[886/1993]
Ministry of the Interior. Väestötietoasetus, 1993. http://www.finlex.fi/fi/laki/ajantasa/1993/19930886, cited 6th May 2009.

[And01]
Ross Anderson. *Security Engineering*. Wiley, 2001.

[Ash01]
Colin G. Ash. e-Business with ERP: A primary study of e-ERP implementations. In *Proceedings of the 2nd Working of e-Business Conference*, pages 364–375, Perth, Australia, 2001. Edith Cowan University. http://www-business.ecu.edu.au/schools/man/media/pdf/0008.pdf, cited 8th May 2008.

[BNS07]
Karine Barzilai-Nahon and Hans J. (Jochen) Scholl. Similarities and Differences of E-Commerce and e-Government: Insights from a Pilot Study. In *HICSS '07: Proceedings of the 40th Annual Hawaii International Conference on System Sciences*, page 92, Washington, DC, USA, 2007. IEEE Computer Society.

[Cam04a]
J.L. Camp. Digital identity. *Technology and Society Magazine, IEEE*, 23(3):34–41, Fall 2004.

[Cam04b]
L. Jean Camp. Identity in Digital Government. *SSRN eLibrary*, 2004.

[CIA2009]
Central Intelligence Agency. CIA World Fact Book, 2009. https://www.cia.gov/library/publications/the-world-factbook/geos/xx.html, cited 1st December 2009.

[Dep85]
Department of Defense. *Department of Defense Trusted Computer System Evaluation Criteria*, 1985. DOD 5200.28-STD (supersedes CSC-STD-001-83).

[DK-SAML]
Center for Services Orientered Infrastructure Danish National IT & Telecom Agency, editor. OIO Web SSO Profile V2.0.5, March 2008. http://www.oiosaml.info/OIOWebSSOProfileV205-review.pdf, cited May 2008.

[ETSI-MSS]
European Telecommunications Standards Institute, editor. ETSI TS 102 204 v1.1.4 Mobile Commerce (M-COMM): Mobile Signature Service: Web Service Interface, August 2003.

[Hod07]
Jeff Hodges. Technical Comparison: OpenID and SAML - Draft 06, January 2007. http://identitymeme.org/doc/draft-hodges-saml-openid-compare.html, cited 12th May 2009.

[ID-WSF-authn]
Jeff Hodges, Robert Aarts, Paul Madsen, and Scott Cantor, editors. Liberty Identity Web Services Framework (ID-WSF) Authentication, Single Sign-On, and Identity Mapping Services Specification v2.0, October 2006. http://www.projectliberty.org/liberty/content/download/871/6189/file/liberty-idwsf-authn-svc-v2.0.pdf, cited 6th May 2007.

[ID-WSF-client-profiles]
Robert Aarts, Jukka Kainulainen, and John Kemp, editors. Liberty ID-WSF Profiles for Liberty enabled User Agents and Devices, version v2.0, October 2006. http://www.projectliberty.org/liberty/content/download/874/6198/file/liberty-idwsf-client-profiles-v2.0.pdf, cited 6th May 2007.

[Iha07]
Petteri Ihalainen. Federation - Ubilogin Whitepaper, August 2007.

[KK07]
Yrjö Kari-Koskinen. Implementing a Web Service Identity Provider in a Production Environment. Master's thesis, Helsinki University of Technology, May 2007. http://peruna.fi/~ykk/dippa/implementing_wsidp-2s.pdf.

[Kou05]
Jim Kouri. Social Security Cards: De Facto National Identification, November 2005. http://www.americanchronicle.com/articles/view/3911, cited 15th May 2009.

[Kä08]
Arto Käpynen. Processing Identity Information in Federated Single Sign-On System. Master's thesis, Helsinki University of Technology, March 2008.

[Liberty-glossary]
Jeff Hodges, editor. Liberty Technical Glossary, October 2006. http://www.projectliberty.org/liberty/content/download/868/6180/file/liberty-glossary-v2.0.pdf, cited 6th May 2007.

[Liberty-Interop]
Kyle Meadors, editor. Test plan for Liberty Alliance SAML test event test criteria SAML 2.0, version 3.1, 2008. http://www.projectliberty.org/liberty/content/download/4160/27946/file/Liberty_Interoperability_SAML_Test_Plan_v3.1.pdf, cited 6th May 2009.

[Liberty-overview]
Conor Cahill. Liberty Technology Overview, August 2006. http://www.projectliberty.org/liberty/content/download/800/5730/file/SpecsOverviewAOL.pdf, cited 6th May 2007.

[Liberty-tech]
Conor Cahill, Carolina Canales-Valenzuela, Frederick Hirsch, Paul Madsen, Prateek Mishra, Rob Philpott, Jeff Smith, Eric Tiffany, and Greg Whitehead. Liberty Technology Tutorial, June 2005. http://www.projectliberty.org/resources/LibertyTechnologyTutorial.pdf, cited 7th March 2006.

[Liberty-ID-FF]
Thomas Wason, editor. Liberty ID-FF Architecture Overview, Version: 1.2-errata-v1.0, November 2003. http://www.projectliberty.org/liberty/content/download/318/2366/file/draft-liberty-idff-arch-overview-1.2-errata-v1.0.pdf, cited 24th May 2009.

[Lin09]
Mikael Linden. *Organisational and Cross-Organizational Identity Management*. PhD thesis, Tampere University of Technology, January 2009.

[Nyk02]
Toni Nykänen. Secure Cross-Platform Single Sign-On Solution for the World-Wide Web. Master's thesis, Helsinki University of Technology, May 2002.

[OpenID]
David Recordon and Drummond Reed. OpenID 2.0: a platform for user-centric identity management. In *DIM '06: Proceedings of the second ACM workshop on Digital identity management*, pages 11–16, New York, NY, USA, 2006. ACM.

[PG07]
John Palfrey and Urs Gasser. *Digital Identity Interoperability and eInnovation*. Berkman Center for Internet & Society, 2007. http://cyber.law.harvard.edu/interop, cited 10th Otober 2009.

[Ren05]
Karen Renaud. Evaluating authentication mechanisms. In Lorrie Faith Cranor and Simson Garfinkel, editors, *Security and Usability*, pages 103–128. O'Reilly Media, 2005.

[SAML-1.0]
Phillip Hallam-Baker and Eve Maler, editors. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML), May 2002. http://www.oasis-open.org/committees/security/docs/, cited 30th April 2009.

[SAML-1.1]
Eve Maler, Prateek Mishra, and Rob Philpott, editors. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1, September 2003. http://www.oasis-open.org/committees/documents.php?wg_abbrev=security, cited 30th April 2009.

[SAML-bindings]
Scott Cantor, Frederick Hirsch, John Kemp, Rob Philpott, and Eve Maler, editors. Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf, cited 6th May 2007.

[SAML-core]
Scott Cantor, John Kemp, Rob Philpott, and Eve Maler, editors. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf, cited 6th May 2007.

[SAML-Diff-11]
Prateek Mishra, editor. Differences between OASIS Security Assertion Markup Language (SAML) V1.1 and V1.0, May 2003. http://www.oasis-open.org/committees/documents.php?wg_abbrev=security, cited 30th April 2009.

[SAML-Diff-20]
saml.xml.org, editor. Differences between SAML 2.0 and 1.1, January 2008. http://saml.xml.org/differences-between-saml-2-0-and-1-1, cited 22nd May 2009.

[SAML-discovery]
Scott Cantor and Rod Widdowson, editors. Identity Provider Discovery Service Protocol and Profile, March 2008. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf, cited 30th April 2009.

[SAML-metadata]
Scott Cantor, Jahan Moreh, Rob Philpott, and Eve Maler, editors. Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf, cited 6th May 2007.

[SAML-profiles]
John Hughes, Scott Cantor, Jeff Hodges, Frederick Hirsch, Prateek Mishra, Rob Philpott, and Eve Maler, editors. Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf, cited 6th May 2007.

[SAML-tech-overview]
Nick Ragouzis, John Hughes, Rob Philpott, and Eve Maler, editors. Security Assertion Markup Language (SAML) V2.0 Technical Overview, Committee Draft 2, March 2008. http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf, cited 23rd April 2008.

[SFK00]
Ravi Sandhu, David Ferraiolo, and Richard Kuhn. The NIST model for role-based access control: towards a unified standard. In *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*, pages 47–63, New York, NY, USA, 2000. ACM.

[SOAP-part1]
Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau,

and Henrik Frystyk Nielsen, editors. Simple Object Access Protocol (SOAP) Version 1.2 part 1: Messaging framework, June 2003. http://www.w3.org/TR/2003/REC-soap12-part2-20030624/, cited 6th May 2007.

[SOAP-part2]
Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, and Henrik Frystyk Nielsen, editors. SOAP Version 1.2 part 2: Adjuncts, June 2003. http://www.w3.org/TR/2003/REC-soap12-part2-20030624/, cited 6th May 2007.

[SWE 481/1991]
Sweriges Riksdag. Folkbokföringslag, 1991. http://www.notisum.se/rnp/SLS/lag/19910481.htm, cited 12th May 2009.

[TUPAS]
Suomen Pankkiyhdistys, editor. Pankkien Tupas-varmennepalvelu palveluntarjoajille, October 2005. http://www.fkl.fi/asp/ida/download.asp?prm1=wwwuser_fkl&docid=11266&sec=&ext=.pdf, cited March 2009.

[UML]
Object Management Groupd, editor. OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.2, February 2009. http://www.omg.org/spec/UML/2.2/Infrastructure, cited May 2009.

[VIRTU]
Petri Suvila and Mika Komu, editors. VIRTU-suunnitteluohje, Ohje Virtu-yhteensopivan palvelun kehittäjälle, April 2008. http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/03_muut_asiakirjat/20080421Virtul/03_suunnitteluohje-20080412.pdf, cited May 2009.

[Virtu-SAML]
Valtiovarainministeriö, editor. Virtu SAML 2.0, Comparison with Oasis Conformance Levels, 2008. http://www.vm.fi/vm/fi/04_julkaisut_ja_asiakirjat/03_muut_asiakirjat/20080421Virtul/01_virtu-saml-20080418.pdf, cited April 2009.

[Wal03]
Elspeth Wales. Identity Theft. *Computer Fraud & Security*, 2003(2):5 – 7, 2003. http://www.sciencedirect.com/science/article/B6VNT-480BG67-6/2/b5cfc90224c775762fcf8c8355f35b69.

[Win05]
Phillip Windley. *Digital Identity: Unmasking Identity Management Architecture* (IMA). O'Reilly & Associates, 2005.

[Windows-SSO]
Microsoft Inc. Single Sign-On in Windows 2000 Networks, 2008. http://technet.microsoft.com/en-us/library/bb742456.aspx, cited 17th April 2008.

[Zei06]
K. Zeilenga. Lightweight Directory Access Protocol (LDAP): Technical Specification Road Map. RFC 4510 (Proposed Standard), June 2006. http://www.ietf.org/rfc/rfc4510.txt.

SAML listings

Listing 1: Full XML listing for B2E scenario. The SAML Response message from IDP to SP.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response xmlns:saml="urn:oasis:names:tc:SAML
   :2.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML
   :2.0:protocol" Destination="https://retailer.inc/
   internal/spsso/saml2/AssertionConsumerService" ID="
   _4e2247b5e0a7929bb68901111b6b9699be3fb601"
   InResponseTo="
   _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
   IssueInstant="2010-03-30T10:05:39.595Z" Version
   ="2.0">
 <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:
    nameid-format:entity">https://idp.retailer.inc/uas
    </saml:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
    xmldsig#">
  <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="http://www.w3.
      org/2001/10/xml-exc-c14n#"/>
   <ds:SignatureMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#rsa-sha1"/>
   <ds:Reference URI="#
      _4e2247b5e0a7929bb68901111b6b9699be3fb601">
    <ds:Transforms>
     <ds:Transform Algorithm="http://www.w3.org
        /2000/09/xmldsig#enveloped-signature"/>
     <ds:Transform Algorithm="http://www.w3.org
        /2001/10/xml-exc-c14n#"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org
       /2000/09/xmldsig#sha1"/>
   <ds:DigestValue>K8VwEzP9HVlFi7JNG5RIaLTyEF8=</ds:
      DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue>-- cut --</ds:SignatureValue>
</ds:Signature>
<samlp:Status>
 <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:
    status:Success"/>
</samlp:Status>
<saml:Assertion ID="
   _c3133a88f646f6d3aaf1c22fb69fb889b3fe5a4e"
   IssueInstant="2010-03-30T10:05:39.595Z" Version
   ="2.0">
 <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:
    nameid-format:entity">https://idp.retailer.inc/uas
     </saml:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
    xmldsig#">
  <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="http://www.w3
      .org/2001/10/xml-exc-c14n#"/>
   <ds:SignatureMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#rsa-sha1"/>
   <ds:Reference URI="#
      _c3133a88f646f6d3aaf1c22fb69fb889b3fe5a4e">
    <ds:Transforms>
     <ds:Transform Algorithm="http://www.w3.org
        /2000/09/xmldsig#enveloped-signature"/>
     <ds:Transform Algorithm="http://www.w3.org
        /2001/10/xml-exc-c14n#"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org
       /2000/09/xmldsig#sha1"/>
    <ds:DigestValue>7S1mGifuZ+VTgBpFYY/g5dm6OaE=</ds:
       DigestValue>
   </ds:Reference>
  </ds:SignedInfo>
  <ds:SignatureValue>-- cut --</ds:SignatureValue>
 </ds:Signature>
 <saml:Subject>
  <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:
      nameid-format:X509SubjectName" NameQualifier="
      ldap://retailer.inc/dc=directory,dc=retailer,dc=
      inc">cn=jdoe,ou=users,dc=directory,dc=retailer,dc
      =inc</saml:NameID>
   <saml:SubjectConfirmation Method="urn:oasis:names:tc
      :SAML:2.0:cm:bearer">
    <saml:SubjectConfirmationData Address
       ="195.197.205.34" InResponseTo="
       _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
       NotOnOrAfter="2010-03-30T10:15:39.595Z"
       Recipient="https://retailer.inc/internal/spsso/
       saml2/AssertionConsumerService"/>
   </saml:SubjectConfirmation>
  </saml:Subject>
  <saml:Conditions NotBefore="2010-03-30T10:03:11.237Z"
      NotOnOrAfter="2010-03-30T10:13:11.237Z">
   <saml:AudienceRestriction>
    <saml:Audience>urn:uuid:38acf336-ab10-3e5d-91e4-908
       f58c0a021</saml:Audience>
   </saml:AudienceRestriction>
  </saml:Conditions>
  <saml:AuthnStatement AuthnInstant="2010-03-30T10
     :05:39.564Z" SessionIndex="
     _03b44fc722c2b86548085b36bc2c327c319de515"
     SessionNotOnOrAfter="2010-03-30T11:05:39.595Z">
   <saml:SubjectLocality Address="195.197.205.34"/>
   <saml:AuthnContext>
    <saml:AuthnContextDeclRef>https://idp.retailer.inc/
       uas/saml2/names/ac/domain.auth</saml:
       AuthnContextDeclRef>
   </saml:AuthnContext>
  </saml:AuthnStatement>
 </saml:Assertion>
</samlp:Response>
```

Listing 2: Full XML listing for B2C scenario. The SAML Response message from IDP to SP.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response xmlns:saml="urn:oasis:names:tc:SAML
   :2.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML
   :2.0:protocol" Destination="https://importer.inc/
   service/spsso/saml2/AssertionConsumerService" ID="
```

```
  _4e2247b5e0a7929bb68901111b6b9699be3fb601"
  InResponseTo="
  _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
  IssueInstant="2010-03-30T10:05:39.595Z" Version
  ="2.0">
<saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:
   nameid-format:entity">https://idp.retailer.inc/uas
   </saml:Issuer>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/
   xmldsig#">
 <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3.
     org/2001/10/xml-exc-c14n#"/>
  <ds:SignatureMethod Algorithm="http://www.w3.org
     /2000/09/xmldsig#rsa-sha1"/>
  <ds:Reference URI="#
     _4e2247b5e0a7929bb68901111b6b9699be3fb601">
   <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org
       /2000/09/xmldsig#enveloped-signature"/>
    <ds:Transform Algorithm="http://www.w3.org
       /2001/10/xml-exc-c14n#"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#sha1"/>
   <ds:DigestValue>AIczX/rhLyPdsaYg0zmBaBPWbz8=</ds:
      DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue>-- cut --</ds:SignatureValue>
</ds:Signature>
<samlp:Status>
 <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:
    status:Success"/>
</samlp:Status>
<saml:Assertion ID="
   _c3133a88f646f6d3aaf1c22fb69fb889b3fe5a4e"
   IssueInstant="2010-03-30T10:05:39.595Z" Version
   ="2.0">
 <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:
    nameid-format:entity">https://idp.retailer.inc/uas
    </saml:Issuer>
```

```
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/
   xmldsig#">
 <ds:SignedInfo>
  <ds:CanonicalizationMethod Algorithm="http://www.w3
     .org/2001/10/xml-exc-c14n#"/>
  <ds:SignatureMethod Algorithm="http://www.w3.org
     /2000/09/xmldsig#rsa-sha1"/>
  <ds:Reference URI="#
     _c3133a88f646f6d3aaf1c22fb69fb889b3fe5a4e">
   <ds:Transforms>
    <ds:Transform Algorithm="http://www.w3.org
       /2000/09/xmldsig#enveloped-signature"/>
    <ds:Transform Algorithm="http://www.w3.org
       /2001/10/xml-exc-c14n#"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#sha1"/>
   <ds:DigestValue>2i9N0ev6cwesN3cYwQOf4E/vG+o=</ds:
      DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue>-- cut --</ds:SignatureValue>
</ds:Signature>
<saml:Subject>
 <saml:NameID Format="urn:oasis:names:tc:SAML:2.0:
    nameid-format:persistent">
    ijKlJAaKAPrSHbqlbcKWu7JktcKY</saml:NameID>
 <saml:SubjectConfirmation Method="urn:oasis:names:tc
    :SAML:2.0:cm:bearer">
  <saml:SubjectConfirmationData Address
     ="195.197.205.34" InResponseTo="
     _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
     NotOnOrAfter="2010-03-30T10:15:39.595Z"
     Recipient="https://importer.inc/service/spsso/
     saml2/AssertionConsumerService"/>
 </saml:SubjectConfirmation>
</saml:Subject>
<saml:Conditions NotBefore="2010-03-30T10:03:11.237Z"
    NotOnOrAfter="2010-03-30T10:13:11.237Z">
 <saml:AudienceRestriction>
  <saml:Audience>urn:uuid:38acf336-ab10-3e5d-91e4-908
     f58c0a021</saml:Audience>
   </saml:AudienceRestriction>
  </saml:Conditions>
  <saml:AuthnStatement AuthnInstant="2010-03-30T10
     :05:39.564Z" SessionIndex="
     _03b44fc722c2b86548085b36bc2c327c319de515"
     SessionNotOnOrAfter="2010-03-30T11:05:39.595Z">
   <saml:SubjectLocality Address="195.197.205.34"/>
   <saml:AuthnContext>
    <saml:AuthnContextDeclRef>https://idp.retailer.inc/
       uas/saml2/names/ac/domain.auth</saml:
       AuthnContextDeclRef>
   </saml:AuthnContext>
  </saml:AuthnStatement>
 </saml:Assertion>
</samlp:Response>
```

Listing 3: Full XML listing for G2C scenario. The SAML Response message from IDP to SP.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response xmlns:saml="urn:oasis:names:tc:SAML
   :2.0:assertion" xmlns:samlp="urn:oasis:names:tc:SAML
   :2.0:protocol" Destination="https://example.com/
   portal/spsso/saml2/AssertionConsumerService" ID="
   _4e2247b5e0a7929bb68901111b6b9699be3fb601"
   InResponseTo="
   _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
   IssueInstant="2010-03-30T10:05:39.595Z" Version
   ="2.0">
 <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:
    nameid-format:entity">https://testi.tunnistus.fi/
    ubitp</saml:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
    xmldsig#">
  <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="http://www.w3.
      org/2001/10/xml-exc-c14n#"/>
   <ds:SignatureMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#rsa-sha1"/>
   <ds:Reference URI="#
      _4e2247b5e0a7929bb68901111b6b9699be3fb601">
    <ds:Transforms>
     <ds:Transform Algorithm="http://www.w3.org
```

```
       /2000/09/xmldsig#enveloped-signature"/>
    <ds:Transform Algorithm="http://www.w3.org
       /2001/10/xml-exc-c14n#"/>
   </ds:Transforms>
   <ds:DigestMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#sha1"/>
   <ds:DigestValue>Xjlb6BKy0zOmo+AS7/rxG6DGoKk=</ds:
      DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue>-- cut --</ds:SignatureValue>
</ds:Signature>
<samlp:Status>
 <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:
    status:Success"/>
</samlp:Status>
<saml:Assertion ID="
   _c3133a88f646f6d3aaf1c22fb69fb889b3fe5a4e"
   IssueInstant="2010-03-30T10:05:39.595Z" Version
   ="2.0">
 <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:
    nameid-format:entity">https://testi.tunnistus.fi/
    ubitp</saml:Issuer>
 <ds:Signature xmlns:ds="http://www.w3.org/2000/09/
    xmldsig#">
  <ds:SignedInfo>
   <ds:CanonicalizationMethod Algorithm="http://www.w3
      .org/2001/10/xml-exc-c14n#"/>
   <ds:SignatureMethod Algorithm="http://www.w3.org
      /2000/09/xmldsig#rsa-sha1"/>
   <ds:Reference URI="#
      _c3133a88f646f6d3aaf1c22fb69fb889b3fe5a4e">
    <ds:Transforms>
     <ds:Transform Algorithm="http://www.w3.org
        /2000/09/xmldsig#enveloped-signature"/>
     <ds:Transform Algorithm="http://www.w3.org
        /2001/10/xml-exc-c14n#"/>
    </ds:Transforms>
    <ds:DigestMethod Algorithm="http://www.w3.org
       /2000/09/xmldsig#sha1"/>
    <ds:DigestValue>DIpJaCKegLe4E5bfyDDWlV/Uy3Y=</ds:
       DigestValue>
  </ds:Reference>
 </ds:SignedInfo>
 <ds:SignatureValue>-- cut --</ds:SignatureValue>
</ds:Signature>
<saml:Subject>
 <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:
    nameid-format:unspecified">012345-678D</saml:
    NameID>
 <saml:SubjectConfirmation Method="urn:oasis:names:tc
    :SAML:2.0:cm:bearer">
  <saml:SubjectConfirmationData Address
     ="195.197.205.34" InResponseTo="
     _34d5fe1ac392fe7978b2cd8a8c43580a542bb4a7"
     NotOnOrAfter="2010-03-30T10:15:39.595Z"
     Recipient="https://example.com/portal/spsso/
     saml2/AssertionConsumerService"/>
 </saml:SubjectConfirmation>
</saml:Subject>
<saml:Conditions NotBefore="2010-03-30T10:03:11.237Z"
    NotOnOrAfter="2010-03-30T10:13:11.237Z">
 <saml:AudienceRestriction>
  <saml:Audience>urn:uuid:38acf336-ab10-3e5d-91e4-908
     f58c0a021</saml:Audience>
 </saml:AudienceRestriction>
</saml:Conditions>
<saml:AuthnStatement AuthnInstant="2010-03-30T10
   :05:39.564Z" SessionIndex="
   _03b44fc722c2b86548085b36bc2c327c319de515"
   SessionNotOnOrAfter="2010-03-30T11:05:39.595Z">
 <saml:SubjectLocality Address="195.197.205.34"/>
 <saml:AuthnContext>
  <saml:AuthnContextDeclRef>https://testi.tunnistus.
     fi/ubitp/saml2/names/ac/tupas.test.3</saml:
     AuthnContextDeclRef>
 </saml:AuthnContext>
</saml:AuthnStatement>
<saml:AttributeStatement>
 <saml:Attribute Name="tfi.CUSTID">
  <saml:AttributeValue xmlns:xs="http://www.w3.org
     /2001/XMLSchema" xmlns:xsi="http://www.w3.org
     /2001/XMLSchema-instance" xsi:type="xs:string
     ">012345-678D</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="id.ref">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       tfi.CUSTID</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="tfi.version">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       katso-1.1</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="name.ref">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       tfi.CUSTNAME</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="tfi.custname">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       John Doe</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute Name="id.type">
    <saml:AttributeValue xmlns:xs="http://www.w3.org
       /2001/XMLSchema" xmlns:xsi="http://www.w3.org
       /2001/XMLSchema-instance" xsi:type="xs:string">
       hetu</saml:AttributeValue>
   </saml:Attribute>
  </saml:AttributeStatement>
 </saml:Assertion>
</samlp:Response>
```