uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Mansour Alsaleh

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

Enhancing Consumer Privacy in Identity Federation Architectures

TITRE DE LA THÈSE / TITLE OF THESIS

Liam Peyton

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

Carlisle Adams

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

A. El Saddik

A. Somayaji

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

Enhancing Consumer Privacy in Identity Federation Architectures

By

Mansour A. Alsaleh

A thesis submitted to

the Faculty of Graduate Studies and Research

in partial fulfillment of

the requirements for the degree of

Master of Computer Science

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering

University of Ottawa

Ottawa, Ontario, Canada

Your file Votre référence
ISBN: 978-0-494-25739-5
Our file Notre référence
ISBN: 978-0-494-25739-5

Abstract

Internet usage has been growing significantly, and the issue of online privacy has become a correspondingly greater concern. Several recent surveys show that users' concern about the privacy of their personal information reduces their use of electronic businesses and Internet services; furthermore, many users choose to provide false data in order to protect their real identities. Identity federation aims to assemble an identity virtually from a user's personal information stored across multiple distinct identity management systems. Liberty Alliance is one of the most recognized projects in developing an open standard for federated network identity. Although one of the key objectives of the Liberty Alliance project is to enable consumers to protect the privacy and security of their network identity information, we believe that this objective is biased towards the business goals of the Liberty architecture rather than giving priority to privacy from the consumer's perspective. In fact, consumer privacy is among the main usage barriers that reduces the adoption of identity federation architectures. In this thesis, we aim to enhance consumer privacy in the Liberty Alliance architecture. In the first place, we identify and analyze previously-unknown potential privacy breaches and concerns within the Liberty identity federation frameworks and propose improvements and recommendations for the identified breaches. We also propose three new services to the Liberty Alliance identity federation architecture that will help in enhancing consumer privacy: user privacy preferences service, privacy seal service, and audit trail service. We present several use case scenarios to demonstrate the effectiveness of the new services and to show how they could merge effectively with the Liberty identity federation frameworks. Furthermore, we propose using the XPref language for our proposed Liberty user privacy preferences service and identify some APPEL issues in this context.

Acknowledgements

I would like to thank both of my supervising professors, Dr. Carlisle Adams and Dr. Liam Peyton, for all their patience, support and guidance throughout this Master's thesis. It would not have been possible without their substantial amount of help and suggestions all over my work.

I would like to thank the reviewers of my research publications for their valuable feedback on my published papers during my Master's thesis research. In particular, a special thank-you to Paul Madsen, a Liberty Alliance Project Editor from NTT group. At the same time I would like to extend my thanks to my Master's thesis committee.

This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Ontario Research Network for Electronic Commerce (ORNEC).

Content

List of Figures

List of Tables

List of Acronyms

AP	Attribute Provider
APPEL	A P3P Preference Exchange Language
CoT	Circle of Trust
DLL	Dynamic Linked Library
DPA	Data Protection Act
DS	Discovery Service
EIS	Enterprise Information System
EPA	Enterprise Privacy Architecture
EPAL	Enterprise Privacy Authorization Language
EPIC	Electronic Privacy Information Center
EUDPD	European Union Data Protection Directive
FIM	Federated Identity Management
FIPA	BC Freedom of Information and Privacy Association the Privacy Act
FTC	Health Insurance Portability and Accountability Act
FOIA	the Freedom of Information Act
ID-FF	Liberty Identity Federation Framework
IdP	Identity Provider
ID-SIS	Liberty Identity Service Interface Specifications
ID-WSF	Liberty Identity Web Services Framework
IIS	Internet Information Services
IMA	Identity Management Architecture
IS	Interaction Service
IT	Information Technology
LE	Liberty-Enabled
OASIS	Organization for the Advancement of Structured Information Standards
P3P	Platform for Privacy Preferences

PDP	Policy Decision Point
PEP	Policy Enforcement Point
PET	Privacy Enhanced Technologies
PI	Personal Information
PII	Personal Identifiable Information
PIPEDA	Personal Information Protection and Electronic Documents Act
PIR	Private Information Retrieval
PKI	Public Key Infrastructure
PPCS	Privacy Policy Compliance System
PRC	Privacy Rights Clearinghouse
RBAC	Role-Based Access Control
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SP	Service Provider
SSO	Single Sign-On
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WSDL	Web Services Description Language
WS-Security	Web Services Security
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language
Xpref	An XPath-based Preference Language for P3P

Chapter 1

Introduction

This chapter provides a brief introduction to the thesis and outlines what motivates our research. It also discusses thesis objective and contributions, scope, publications, and how the rest of this thesis is organized.

1.1 Introduction

Internet users maintain many separate accounts on different Internet businesses and services. Web sites almost always keep a profile for each visitor; this profile will contain more personally identifiable information when the user registers his/her information. Users usually need distinct authentication credentials (e.g. user name and a password) to access their profiles. This collection of credentials becomes difficult to manage and control over time offering inroads for identity theft. Identity federation enables users to link, assemble and control an identity virtually from separate accounts where a user can control sharing of his/her identity attributes between service providers. Identity federation defines mechanisms for enterprises to share identity information between domains. These mechanisms include single sign-on (SSO), authorization, identity mapping and account linking, and directory services. The single sign-on mechanism reduces redundant logons whereby a user can login once with a member of a federate group and gain access to resources of multiple members among the group without signing-on once more.

Identity federation created what is sometimes called a federated identity. An "identity" can simply refer to a set of traits, attributes, and preferences for which a person may receive personalized services. The "federation" is a group of several

businesses/organizations that have agreed to allow a user from one federation partner to seamlessly access resources from another partner in a secure and trustworthy manner. Federated Identity Management (FIM) is the set of procedures and protocols necessary to build a framework that enables identity federation mechanisms and functions. In identity federation architecture, the user can be a member of one or more Circles of Trust (CoT) which is defined as a group of service providers that share linked identities and have pertinent business agreements in place regarding how to do business and interact with identities. Once a user has been authenticated by a CoT identity provider (using SSO mechanism), that individual can be easily recognized and take part in targeted services from other service providers within that CoT. In addition to the SSO benefit of fewer redundant logons, identity federation has the advantages of reducing the administrative costs of user profiles for the users (fewer identities and more manageable) and for service providers (off load the expensive part of the user management to one business partner which is the identity provider in the CoT) and keeping more accurate and up-to-date information about users. Furthermore, identity federation helps businesses in increasing their customers' affiliation and sharing of their customers' personal information (under user control) [18, 54, 55].

For the identity federation concept to become a reality, however, there needs to be a common set of technical and business standards to enable different businesses and organizations to integrate easily in building such a federated architecture. One of the early identity federation frameworks is Microsoft's Passport. It deploys a centralized framework where there is just one identity provider (Microsoft). Any time a user logs into a Passport-participating site, the site is immediately able to access the information in the user's Passport account. Users' privacy concerns (users have no privacy with respect to the identity provider which is Microsoft) and the concept of a single trusted third party are some of the reasons that led to limited adoption of this architecture [45]. Microsoft carried out a significant upgrade to Passport and changed the service name to Windows Live ID. The new service overcomes the limitation of not supporting multiple identity management by utilizing a new Microsoft model, InfoCard, which is an identity selector that enables users to manage and exchange their digital identities [17]. InfoCard supports more than one identity provider (not just Microsoft) and the identity provider can be the

user machine itself. One major drawback of InfoCard is that users lose their digital identities when using a different machine (unless the user uses an external security token such as a smartcard).

Set up at the instigation of Sun Microsystems in 2001, the Liberty Alliance is a consortium of technology vendors and consumer-facing enterprises formed to develop an open standard for federated network identity. The Liberty Alliance project is based on the concept of enabling users to connect multiple sets of personal information which exist across several e-commerce providers into a single easy-to-manage federated identity. This allows for the convenience of an SSO mechanism as well as easier administration of personal information across multiple service providers [54, 93]. Liberty Alliance is one of the most prominent and widely accepted federated identity standard proposals (several other identity federation architectures exist, including, Shibboleth and WS-Federation; we will discuss the different existing architectures in chapter 2). The Liberty architecture has obtained support from over 150 well-known companies and organizations in the last few years. Therefore, the focus of this thesis is on the Liberty Alliance identity federation architecture.

1.2 Motivation

Privacy seems to be of concern to Internet users. Collecting personal information without users' awareness, sharing personal information between businesses without users' consent, using personal information for purposes other than stated, and the inability to access, change, or delete personal information, are among the main privacy concerns for many users. In 2003, industry watchdog Gartner Group predicted that by 2006, the first barrier to electronic business and commerce will be user concerns over information privacy [37]. A March 2000 Business-Week/Harris Poll shows that 86% of users want a Web site to obtain opt-in consent before collecting user name, address, phone number, or financial information. The same poll shows that 88% of users support opt-in as the standard before a Web site shares personal information with others [19]. A Pew Internet & American Life Project survey in 2000 found that 86% of Internet users support opt-in privacy policies before companies use personal information, 54% believe that web site

tracking of users is harmful and privacy invasive, 24% of Internet users reported giving false information to a web site and 20% gave alternative or secondary e-mail addresses to web sites [36].

In fact, privacy concerns were one of the reasons that led to limited adoption of Microsoft Passport identity federation architecture. Not just from the user privacy perspective (the user has no privacy with respect to the identity provider which is Microsoft) but from the service provider privacy perspective as well. In the Passport architecture, the service providers need to deal with and trust only a single third party (Microsoft) that can monitor most of their customers' transactions. This made many Internet businesses avoid using the architecture at all and prefer instead to interact with their customers directly without the burden of managing a business relationship with Microsoft. Although one of the key objectives of the Liberty Alliance project is to enable consumers to protect the privacy and security of their network identity information, the multidiscipline specifications Liberty covers make it vulnerable to a variety of privacy breaches. Moreover, we believe that this objective is biased towards business goals of the Liberty architecture rather than giving priority to privacy from the consumer's point of view. Knowing the advantages of identity federation and the need for a set of technical and business standards to regulate and facilitate the different federation mechanisms, Liberty Alliance identity federation architecture is an attractive choice that is evolving as a widely-accepted open standard for federated network identity. However, any successful identity federation architecture has to satisfy sufficiently user confidentiality and privacy in particular in addition to online service providers' privacy. Inventing a new privacy-aware architecture or making major changes to existing ones like Liberty will probably lack adoption and standardization. Consequently, enhancing consumer privacy in the Liberty architecture is a better approach that will increase consumer trust (in using Liberty-enabled providers' services) and thus will lead to greater adoption of Liberty standards.

1.3 Thesis Objective and Contributions

As we indicated above, our objective in this thesis is to enhance consumer privacy in the Liberty identity federation architecture that will increase consumer trust in using Liberty-enabled providers' services and hence will lead to greater adoption of Liberty identity federation standards. The contributions of this thesis can be summarized in the following main points:

- Documenting previously unknown privacy breaches and concerns in Liberty Alliance identity federation frameworks and proposing solutions and recommendations to the identified privacy breaches to enhance consumer privacy (section 3.3).
- Architectural proposal to Liberty Alliance Project by introducing new services to Liberty frameworks:
    - User Privacy Preferences Service (section 4.3.1)
    - Privacy Seal Service (section 4.1)
    - Audit Trail Service (section 4.2)
- Proposed extension to P3P terminology to make it appropriate for identity federation privacy requirements (section 4.3.2).

1.4 Discussion of the Contributions

As part of the first contribution, we will go through a comprehensive privacy analysis for the Liberty Alliance frameworks:

- The Liberty Identity Federation Framework (ID-FF)
- The Liberty Identity Web Services Framework (ID-WSF)
- The Liberty Identity Service Interface Specifications (ID-SIS)

For each identified privacy breach or concern, we will propose improvements that could help in enhancing consumer privacy. The privacy breaches are identified in the following components in section 3.3:

A. Identity Federation
B. Single Sign-On

C. Discovery Service

D. Interaction Service

E. User Attribute Access Control

The proposed new services in the second point aim to enhance consumer privacy in the first place (discussed in section chapter 4). Although the new services will be part of the Liberty ID-WSF, they will have interactions with the other frameworks. Among the new three services, the focus in this thesis will be on the user privacy preferences service that enhances user privacy in many cases (section 4.3.1). In particular: attribute request at the attribute provider, SP attribute request from the user and service providers introduction at the IdP. The key advantage of the proposed privacy preferences service is that some access permission requests can be directed first to this service to find out whether the user's default preferences allow the requested access. This service gives the consumer more control over his identity information without compromising identity federation ease-of-use advantage. The Liberty seal service assures the user and other providers how the SP privacy policies comply with the Liberty identity federation privacy requirements (section 4.1). The Liberty seal has four different levels of compliance: "very compliant", "Compliant", "Some Compliance", and "Not certified". Finally, the audit trail service (section 4.2) helps the user to know if any privacy violation has been committed by any SP allowing him to take the appropriate action for future transactions and permitting him to update his privacy preferences accordingly. We will demonstrate the effectiveness of the new services by presenting several use case scenarios.

In the third contribution, we will go further in specifying the user privacy preferences service by proposing an extension to P3P to accommodate identity federation privacy requirements. This extension consists mainly of adding new elements to the P3P terminology. We also propose the use of XPref (section 4.3.2), an XPath-based Preference Language for P3P, as an appropriate candidate to specify user privacy preferences in our proposed Liberty user privacy preferences service. In addition, we will compare XPref and APPEL (A P3P Preferences Exchange Language) showing how some privacy requirements for the new user privacy preferences service are met by XPref and not by APPEL.

Throughout this research, we desire to make sure that our proposed privacy enhancements to Liberty frameworks are compliant with Liberty specifications as much as possible. Furthermore, any proposed new service to Liberty frameworks must significantly enhance consumer privacy and must merge easily with the current specification.

1.5 Thesis Publications

The following is the list of the publications that I co-authored during my Master thesis research (both papers are accepted and published):

1. Alsaleh, M. and Adams, C. "Enhancing Consumer Privacy in the Liberty Alliance Identity Federation and Web Services Frameworks". In *Proceedings of the 6th Workshop on Privacy Enhancing Technologies (PET 2006)*, Cambridge, United Kingdom, June 28 - 30, 2006.
2. Peyton, L., Hu, J., Liu, H., Alsaleh, M., and El Saddik, A. E., "A Collaborative Approval Process for Accessing Sensitive Data". The *International Journal of Computer Applications in Technology*, 2005.

1.6 Organization of the Thesis

This thesis is organized as follows: Chapter 2 provides an overview of current privacy legislation and advocates, discusses online privacy, and provides a detailed taxonomy of privacy enhanced technologies (PET): communication privacy enforcement, local privacy enforcement, web privacy enforcement, and organizational privacy enforcement. This PET detailed overview will help in choosing and deploying the appropriate PETs in our proposals and improvements to the Liberty identity federation architecture. This chapter also provides a classification and overview of the identity federation architectures available. Chapter 3 goes through our privacy analysis of the Liberty Alliance architecture and identifies and analyses consumer privacy breaches and concerns. It also proposes improvements to the identified breaches to enhance consumer privacy. Chapter 4 discusses the proposed new services to the Liberty architecture that aim to enhance consumer privacy. In particular, this chapter focuses on the Liberty user privacy preferences service. Moreover, we propose using the XPref language for the Liberty user privacy preferences service, extend P3P, and identify some APPEL issues in this context.

Chapter 5 provides some use case scenarios that demonstrate the effectiveness and the benefits of the proposed services. This chapter also analyzes how the proposed services integrate and are compatible with Liberty, and demonstrates that the proposed services are not Liberty specific but are also applicable to other identity federation architectures. Chapter 6 discusses possible future directions for this work and draws conclusions.

Chapter 2

Background

This chapter starts by providing an overview of current privacy legislation and advocates that will help us in recognizing which privacy breaches contravene defined laws. We also provide a detailed taxonomy of privacy enhancing technologies (PETs): communication privacy enforcement, local privacy enforcement, web privacy enforcement, and organizational privacy enforcement. This PET detailed overview will help in choosing and deploying the appropriate PETs in our proposals and improvements to the Liberty identity federation architecture. This chapter also provides a classification and overview of the available identity federation architectures for better understanding of how these architectures work and how they fit our proposed services.

2.1 Privacy Legislation and Advocates

There is significant several privacy legislation and numerous privacy protection laws around the world. Although there is an overlap between these laws, each law has its own scope, objectives, requirements, and obligations. The European strategy tends to burden governments in enacting comprehensive strict privacy protection laws. Europe enacted comprehensive regulations that establish a basic set of privacy principles for the collection and processing of personal information in both public and private sectors. On the other hand, U.S. enacted federal legislation that covers most federal government departments and federal agencies or statutory corporations to place limits on the collection, use and disclosure by federal agencies of personal information. Nevertheless, the U.S. strategy encourages businesses to be self-regulated pertaining to privacy

protection while the government has limited involvement. The U.S. Department of Commerce in consultation with the European Commission developed a "Safe Harbor" framework (approved by the EU in July of 2000). Safe Harbor is a compromise that enables U.S. companies to avoid experiencing interruptions or facing prosecution in their business dealings with the EU by European authorities according to their legislation. Safe Harbor certification assures that EU organizations know that the company provides "adequate" privacy protection, as defined by the European Commission's Directive on Data Protection [89].

The Canadian approach involves a mix of privacy federal legislation, provincial legislation and industry self-regulation. In 1999, Canada passed the Personal Information Protection and Electronic Documents Act (PIPEDA) that initially applied to federally regulated companies only, but was extended in January 2004 to all business and commercial activities in those provinces that do not have equivalent laws. The obligations imposed on companies operating in Canada by PIPEDA are stricter than Safe Harbor principles [71].

Different other privacy laws and legislation are around the world with different requirements, scopes, and strictness. In this section, we will briefly go through the major privacy legislation in Europe, U.S. and Canada.

2.1.1 Privacy legislation in Europe

Europe has become the international leader in data privacy protection for a long time. In Europe, data privacy laws were firmly in place prior to the information technology revolution of the late 1990s. Personal information is much less readily available than in comparable sectors in the US and consumer trust of on-line environments has received considerable public policy attention. Likewise, the law enforcement and executive agencies within Europe face unique constraints imposed by the stringent privacy regime. In Europe, privacy legislations involve a combination public and private sectors legislations. There are two major national policies in Europe pertaining to data protection. The first is the Council of Europe's Convention on Data Protection, and the second is the European Union Data Protection Directive (EUDPD) [66].

The Council of Europe Convention sets forth the data subject's right to privacy, enumerates a series of basic principals for data, and calls for mutual assistance between parties to the treaty including the establishment of a consultative committee and a procedure for future amendments to the convention. Moreover, this convention aims to secure in the territory of each party for every individual, whatever his nationality or residence, respect for his rights and fundamental freedoms, and in particular his right to privacy, with regard to automatic processing of personal data relating to him [30].

The EUDPD requires all EU countries to implement strict personal information privacy safeguards domestically and prohibits exporting of personal data to non-EU countries that do not ensure adequate levels of protection. EUDPD 95/46/EC went effective in October 1998 with the objectives of protecting individuals with respect to the processing of personal information and to ensure the free movement of personal information within the EU through harmonization of national laws. EUDPD regulates the collection, use, and transfer of individually identifiable personal information about employees and consumers, (e.g. name or contact address). Special handling and restrictions are mandated for "sensitive" data (e.g. racial or ethnic origins or health information). All methods of collection and processing are encompassed: manual, automatic, online, and offline. A key provision of the Directive restricts the transfer of individual personal information to a non-EU country unless that country has been found to provide an "adequate" level of data protection [33].

According to the directive, personal data should not be processed at all, except when certain conditions are met: transparency, legitimate purpose and proportionality.

- Transparency: The data subject has the right to be informed when his personal data are being processed. The controller must provide his name and address, the purpose of processing, the recipients of the data and all other information required to ensure the processing is fair. The data subject has the right to access all data processed about him. The data subject even has the right to demand the rectification, deletion or blocking of data that is incomplete, inaccurate or is not being processed in compliance with the data protection rules.

- Legitimate Purpose: Personal data can only be processed for specified, explicit and legitimate purposes and may not be processed further in a way incompatible with those purposes.
- Proportionality: Personal data may be processed only insofar as it is adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed. The data must be accurate and, where necessary, kept up to date.

2.1.2 Privacy legislation in U.S.

Most privacy laws and acts in U.S are concerning government rather than the private sector. There is no single law in the U.S that provides a comprehensive data protection. There are a number of laws and executive orders dealing specifically with the concept of data protection. The most important and broad based of these laws are the Privacy Act of 1974 and the Computer Matching and Privacy Act. These laws deal exclusively with personal information held by the federal government. The Privacy Act of 1974 resulted from studies by consumer groups and the federal Department of Health, Education & Welfare among others. The Privacy Act is a companion to and extension of the Freedom of Information Act (FOIA) of 1966. FOIA protects the rights of the public to information and makes provisions for individuals to obtain information on the operation of federal agencies. The privacy act was followed by a range of specific federal and state enactments such as the 1970 Fair Credit Reporting Act, 1984 Cable Communications Policy Act, 1974 Family Educational Rights & Privacy Act, and 1986 Electronic Communications Privacy Act. In 1987, the Computer Security Act deals with personal information in federal record systems. It protects the security of sensitive personal information in federal computer systems. The act establishes government wide standards for computer security and assigns responsibility for those standards to the National Institute of Standards. Several other laws followed that restrict the federal government's access to records held by other sources [81].

In 1996, the Health Insurance Portability and Accountability Act (HIPPA) was enacted. The act protects the privacy of the individually identifiable health information by establishing conditions for its use and disclosure. The act is composed of two major

legislative actions: provisions for health insurance reform and requirements for administrative simplification. Complying with all aspects of HIPAA will require providers and virtually all entities within the healthcare industry to make significant changes to their information systems, operations policies and procedures and business practices. HIPAA includes protections for coverage under group health plans that limit exclusions for preexisting conditions, prohibit discrimination against employees and dependents based on their health status, and allow a special opportunity to enroll in a new plan to individuals in certain circumstances [70].

2.1.3 Privacy legislation in Canada

At the federal level Canada has two federal privacy laws, the Privacy Act (PA) and the PIPEDA. The PA took effect on July 1, 1983. This Act imposes obligations on most federal government departments and agencies to respect privacy rights by limiting the collection, use and disclosure of personal information. The PA gives Canadians the right to access and request correction of their personal information held by these federal government organizations. The PIPEDA establishes rules that require all private sector organizations operating in Canada that collect, use or disclose personal information in the course of commercial activity to obtain an individual's prior informed consent to such collection, use and disclosure, subject to certain enumerated exceptions. This informed consent requirement obliges the organization to identify to the individual the purposes for the use and disclosure of the personal information collected and the types of parties to whom the personal information is disclosed. The organization is limited to using and disclosing the information only for the identified purposes and to disclosing the information only to the identified parties [71]. The European Commission has published "frequently asked questions" on its adequacy finding of the PIPEDA and the EU Directives [34].

PIPEDA main principles are based on the Canadian Standards Association's Model Code for the Protection of Personal Information (Code) [21]. The Code contains the following ten principles:

- Accountability: an organization is responsible for personal information under its control and shall designate an individual or individuals who are accountable for the organization's compliance with the remaining principles.
- Identifying purposes: purposes for which personal information is collected shall be identified by the organization at or before the time the information is collected.
- Consent: the knowledge and consent of the individual are required for the collection, use, or disclosure of personal information.
- Limiting the collection: the collection of personal information shall be limited to that which is necessary for the purposes identified by the organization. Information shall be collected by fair and lawful means.
- Limiting Use, Disclosure and Retention: personal information shall not be used or disclosed for purposes other than those for which it was collected, except with the consent of the individual or as required by law. Personal information shall be retained only as long as necessary for the fulfillment of those purposes.
- Accuracy: personal information shall be as accurate, complete and up-to-date as is necessary for the purpose for which it is used.
- Safeguards: personal information shall be protected by security safeguards appropriate to the sensitivity of the information.
- Openness: an organization shall make readily available to individuals specific information about its policies and practices relating to the management of personal information.
- Individual Access: upon request, an individual shall be informed of the existence, use and disclosure of his or her personal information, and shall be given access to that information. An individual shall be able to challenge the accuracy and completeness of the information and have it amended as appropriate.
- Challenging compliance: an individual shall be able to address a challenge concerning compliance with the above principles to the designated individual or individuals accountable for the organization's compliance.

Quebec is the only province that has been found by the Privacy Commissioner of Canada to have substantially similar legislation to PIPEDA in place. Alberta and British

Columbia have introduced omnibus Bills to protect personal information in private sector which came into effect on January 2004. The Privacy Commissioner has yet to determine whether or not these acts are substantially similar to PIPEDA. The PIPEDA provides that express consent is the highest form of consent, but also contemplates reliance on implied consent, depending on the sensitivity of the information. In Quebec, the consent to use or disclosure is valid only if it is free, enlightened, manifest and given for a specific purpose; it is also time-limited. The Implicit consent is considered valid in Quebec only in relation to the use and communication of a nominative list (e.g. a list of names or addresses) for the purpose of commercial or philanthropic prospecting. The British Columbia and Alberta Bills permit reliance on deemed consent in certain circumstances [53].

2.1.4 Privacy Advocates and Commissioners

Privacy Advocates are either individuals or groups that seek to protect privacy through some of the following ways:

- Advocacy and support in development and implementation of federal and provincial policies that protect personal privacy.
- Effective advocacy of both public and private point of view.
- Conduct reviews of federal and provincial authorities' policies and recommend statutory changes to the privacy legislations.
- Warn for any privacy violation or contravention in a governmental or private organization with the help of the media.
- Educate and assist individuals in exercise of their privacy-related rights.
- Carry out social activities such as arranging meetings between individuals and the staffs of relevant authorities to obtain information or discuss complaints regarding policies, practices, and procedures relating to privacy and PII.

The privacy advocacies are usually independent non-profit organizations that seek support from both organizational and individual supporters. Some privacy advocacies run for specific privacy concerns, for example Internet privacy concerns. The following are some privacy advocacy groups:

- **Privacy International (PI):** is a human rights group formed in 1990 as a watchdog on surveillance and privacy invasions by governments and corporations. It is based in London, England, and has conducted campaigns and research throughout the world on issues ranging from wiretapping and national security, to ID cards, video surveillance, data matching, medical privacy, and freedom of information and expression [78].
- **Electronic Privacy Information Center (EPIC):** is a public interest research center in Washington, D.C. It was established in 1994 to focus public attention on emerging civil liberties issues and to protect privacy, freedom of expression, and constitutional values in the information age. EPIC pursues a wide range of activities, including policy research, public education, conferences, litigation, publications, and advocacy. EPIC strive to educate Internet users on privacy concerns [32].
- **Privacy Rights Clearinghouse (PRC):** is a consumer organization with a two-part mission - consumer information and consumer advocacy. It was established in 1992 and is based in San Diego, California. PRC's activities include raising consumers' awareness of how technology affects personal privacy, and empower consumers to take action to control their own personal information by providing practical tips on privacy protection [79].
- **BC Freedom of Information and Privacy Association (FIPA):** is a non-profit society dedicated to advancing freedom of information and privacy rights in Canada, FIPA was the major force in getting B.C.'s Freedom of Information and Protection of Privacy Act passed [35].

Privacy commissioners are statutory offices entrusted to protect personal data privacy of individuals and to ensure compliances with the privacy legislations. Privacy commissioners are usually appointed by the government with the task of both privacy-related issues monitoring and consultation, and public education and awareness of privacy issues. In Canada, The Privacy Commissioner is a special ombudsman and an agent of parliament who reports directly to the House of Commons and the senate. The Commissioner works independently from any other party in the government to investigate complaints from individuals with respect to the federal public sector and the private sector. Moreover, it conducts research into privacy related issues [71].

2.2 Online Privacy

In February and March 2000, the United States Federal Trade Commission (FTC) conducted a survey of commercial sites' information practices, using a list of the busiest U.S. commercial sites on the World Wide Web. The survey shows that websites collect a vast amount of personal information from and about consumers. Almost all sites, 97%, collect an email address or some other type of personal identifying information. 88% of websites post at least one privacy disclosure. Only 20% of websites that collect personal identifying information implement at least in part, the fair information practice principles [90]. Another survey of Internet users from Annenberg Public Policy Center of the University of Pennsylvania (in June 2005), shows that 47% believed falsely that online merchants give consumers the opportunity to see their own data; 49% believed falsely that banks send their customers e-mails asking them to verify their account; 50% believed falsely that online merchants allow consumers to erase their personal information from the company's database; 49% believed falsely that online merchants are required to disclose the names of their affiliates before transferring personal information to them; 52% believed falsely that magazines were barred by law from selling their subscription lists [88].

Lack of internet users' trust is a concern to online businesses. This lack of trust is simply because users are unwilling to provide personal information to online businesses. Consumers are concerned about the extent to which businesses collect their identifiable personal information and how they could share that data with third parties without their knowledge. It is necessary that users feel confident that their confidentiality and privacy are secure and protected. For example, clearly and explicitly stated privacy policies on a website, lets the consumer gain more trust in dealing with the website. Consumers will feel more comfortable dealing with a website that gives them access and control of their personal information and the way it can be shared. Once an online business successes in building such trust, this means more commerce and more users. Consumer protection of online data will be an ongoing concern and will require continued precautions and data protections by businesses in order to keep users' trust.

2.3 Privacy Enhancing Technologies (PET)

Privacy Enhancing Technologies (PETs) represents a spectrum of both known and new technologies to reduce the exposure of users' private data. The PET term does not have a widely accepted definition where the scope of the term differs with each usage scenario. PETs can be classified in many different ways. Functionality, technique methodology, and usage environment are some of these ways. Table 2.1 shows a general functionality classification by [51].

	PETs
General	♦ Encryption (storage & communication) ♦ Logical access controls (authentication and authorization) ♦ Biometrics ♦ Quality-Enhancing Technologies ♦ Cryptography-based IDs ♦ Government access facility (e.g. portals)
Separation of data	♦ Profile management ♦ Privacy incorporated database ♦ Blind electronic signature ♦ Personal data safe
Anonymisation	♦ MIX routers ♦ Onion routers ♦ Cookie management tools ♦ File management tools ♦ Smart cards ♦ Biometrics
Privacy management systems	♦ P3P (Platform for Privacy Preference Project) ♦ Privacy Rights Management (based on Digital Rights Management) ♦ Automatic data destruction (retention management) ♦ PISA (Privacy-Incorporated Software Agent) ♦ Privacy ontologies ♦ EPAL (Enterprise Privacy Authorisation Language) ♦ Privacy policy management software

Table 2.1: PETs Classification [51]

Another classification of PETs relies on the seven PET Principles (derived from European Union Data Protection Directive) [14]:

1. **Limitation in the collection of personal data:** privacy-Enhanced systems should first of all comply with article 6, paragraph 1 (c) of Directive 95/46EC. Personal data

colleted should be adequate, relevant and not excessive in relation for the purpose for which they are collected and/or further processed.

2. **Identification/authentication/authorization:** pursuant to article 16 of Directive 95/46/EC any person acting under the authority of the controller, including the processor himself, who has access to personal data must not process them except on instructions from the controller, unless he is required to do so by law.
3. **Standard techniques used for privacy protection:** article 6, paragraph 1 (b) of the Directive 95/46/EC regulates that further processing of data for historical, statistical or scientific purposes shall not be considered as incompatible provided that appropriate safeguards are provided. These adoptions should guarantee that data kept should only be used for these purposes. The best guarantee is achieved by transforming the data in such a way that the data can no longer be used to identify the data subject. This process is often referred to as rendering anonymous or anonymisation.
4. **Pseudo-identity:** the concept of the original PET principle is the use of the pseudo-identity. Pseudo-identity can be used by a combination of PETs. For example, the first step is to create a pseudo-identity as an identifier. This identity is created in three steps. First a string is created of the first 3 characters of the surname, the first initial, the date of birth and the code indicating the gender of the data subject. In the next step a hash of his string is generated. Last the original string is again removed. This pseudo identity thus created is the same or each of the source information systems: the linkability is thus safeguarded.
5. **Encryption:** encryption is a typical example of an existing technology that may aid in data protection.
6. **Biometrics:** it is the science of using the individual biological properties to identify him.her. For example, fingerprints and voice recognition.
7. **Audit ability:** article 28 of Directive 95/46/EC gives the Supervisory Authorities (the Data Protection Authorities) investigative powers such as powers of access data forming the subject matter of processing operations and powers to collect all the information necessary for the performance of their supervisory duties.

Common Criteria for Information Technology (in Part 2: Security functional requirements) categorized PETs to four categories [26]:

1. **Anonymity:** ensures that a user may use a resource or service without disclosing the user's identity. The requirements for Anonymity provide protection of the user identity. Anonymity is not intended to protect the subject identity.
2. **Pseudonymity:** ensures that a user may use a resource or service without disclosing its user identity, but can still be accountable for that use.
3. **Unlinkability:** ensures that a user may make multiple uses of resources or services without others being able to link these uses together.
4. **Unobservability:** ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource/service is being used.

A usage environment classification can be derived from the proposed privacy enforcement architecture by Adams and Barbieri in [2]. In this classification, PETs can be categorized as the following (figure 2.1):

A. **Communication Privacy Enforcement:** PETs that enforce privacy on the transaction communications of the different parties.
B. **Local privacy enforcement:** PETs that enforce privacy on local systems transactions.
C. **Web privacy enforcement:** PETs that enforcing privacy on web-based transactions.
D. **Organizational privacy enforcement:** PETs that enforcing privacy on transactions that are within the organization or between organizations.

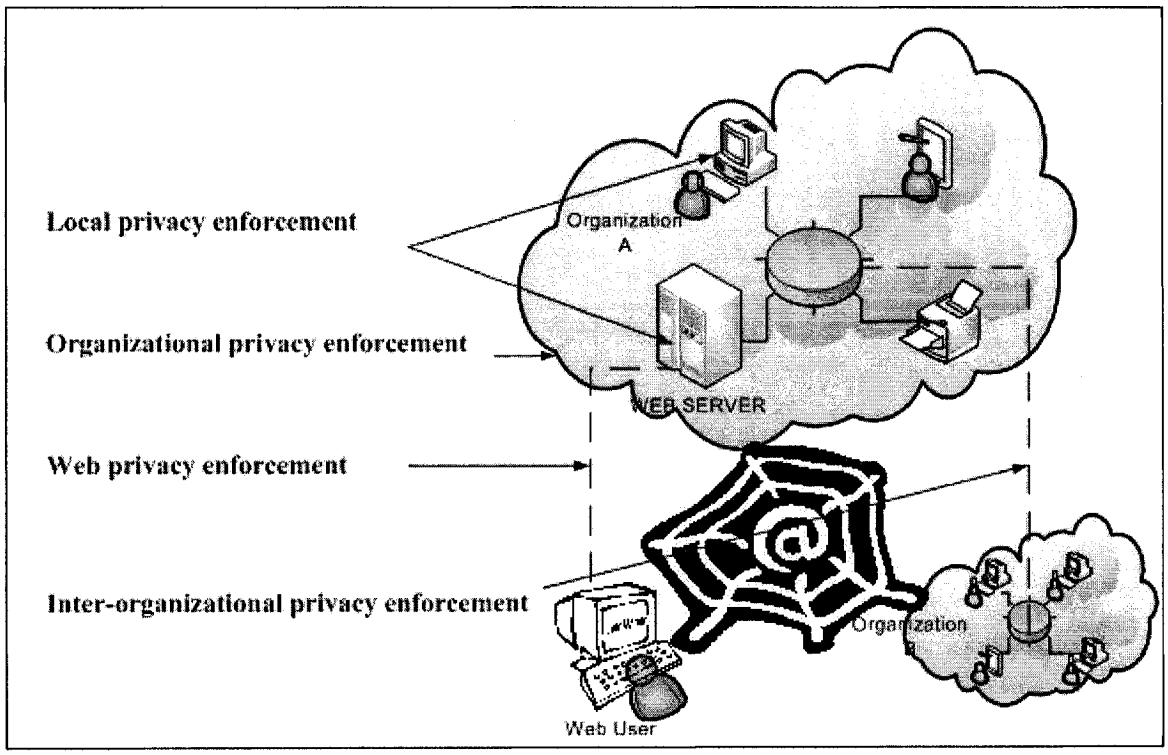

Figure 2.1: PETs classification

2.3.1 Communication Privacy Enforcement

Communication privacy techniques can be classified to two main categories:

I. techniques that limit exposure (i.e. limiting dissemination of identity linkages):
   - Hiding the subject by using anonymous or pseudonymous identifiers: MIXes, Onion Routing, and CROWDS.
   - Hiding the action by burying the action in a large set of actions: Private Information Retrieval (PIR).
   - Hiding the tuple by securing the channel: SSL or SSH.

II. techniques that limit disclosure (i.e. limiting dissemination of identity attributes linkages):
   - Hiding the subject by anonymization of data.
   - Hiding the attributes by randomizing the data.
   - Hiding the tuple by regulating access to data.

In this section, we discuss some important techniques for limiting exposure. In particular, a brief of MIXes, Onion Routing, Crowds, and PIR is given. Explaining the different privacy techniques in detail for limiting exposure and disclosure is out of the scope of this thesis.

2.3.1.1 MIXes

MIX network is a solution to the traffic analysis problem (based on public key cryptography) that allows any single intermediary to provide security for those messages passing through [22]. It also allows messages to be sent or received anonymously (hiding who a participant communicates with as well as the content of the communication). Moreover, the use of a roster of pseudonyms provides limited anonymity. This solution works as following: A participant prepares a message M for delivery to a participant at address A by sealing it with the addressee's public key Ka, appending the address A, and then sealing the result with the mix's public key K1 as following: K1(R1, *Ka(Ro, M), A)* → *Ka(Ro, M), A.* This is the transformation of the input by the mix into the output. The mix decrypts its input with its private key, throws away the random string R1, and outputs the remainder.

The purpose of a mix is to hide the correspondences between the items in its input and those in its output. The order of arrival is hidden by outputting the uniformly sized items in lexicographically ordered batches. If just one item is repeated in the input and is allowed to be repeated in the output, then the correspondence is revealed for that item. Thus, an important function of a mix is to ensure that no item is processed more than once. This function can be readily achieved by a mix for a particular batch by removing redundant copies before outputting the batch.

Since each mix may keep incoming messages in batches until certain amount of messages are available, this could cause significant delay that make this approach not a real-time solution. The sender has the choice either to send his address (encrypted) to the responder or not. In the first case, the sender should include in the sent message its address encrypted with the mixers public keys in reverse order from responder back to him plus a new public key for the session (that the responder use to encrypt his response message).

2.3.1.2 Onion Routing

The essential goal of Onion Routing is to provide private, traffic analysis resistant communications (which is bidirectional and real-time) over a public network at reasonable cost and efficiency. Communications are intended to be private in the sense that both the public network itself and any eavesdropper on the network cannot determine the contents of messages flowing from initiator and responder, and he cannot tell that this particular initiator and responder are communicating with each other. A secondary goal is to provide anonymity to the sender and receiver, so that the responder may receive messages but be unable to identify the sender, even though he may be able to reply to those messages. Using a layered encryption scheme (using public key cryptography), Onion Routing guarantee that only intended recipients of each layer can decrypt it with their private keys. So each hop along the route then only knows about the previous hop where the onion (that includes the message) came from and the next hop where it will forward the onion [80].

2.3.1.3 Crowds

Crowds is based on the idea of hiding one's actions within the actions of many others. A user first joins a crowd of other users. The user's request to a web server is first passed to a random member of the crowd. That member can either submit the request directly to the end server or forward it to another randomly chosen member, and in the latter case the next member chooses to submit or forward independently. When the request is eventually submitted, it is submitted by a random member, thus preventing the end server from identifying its true initiator. Even crowd members cannot identify the initiator of the request, since the initiator is indistinguishable from a member that simply forwards a request from another [82].

Comparing to mixes, crowds provide sender anonymity against collaborating crowd members. In contrast mixes do not provide sender anonymity but do ensure sender and receiver unlinkability. Moreover, mixes provide sender and receiver unlinkability against a global eavesdropper. Crowds does not provide anonymity against global eavesdroppers. Another difference is that mixes typically rely on public key encryption.

2.3.1.4 Private Information Retrieval (PIR)

PIR aims to protect user privacy from a server by allowing a user to retrieve a record from a database while hiding the identity of the record from a database server. PIR is a general problem of private retrieving the i-th bit out of an N-bit string stored at the server. Private means that the server does not know about i, that is, the server does not learn which bit the client is interested in. Motivating examples for this problem include databases with sensitive information (e.g. medical records or stocks rates) in which users are likely to be highly motivated to hide which record they are looking for. The main cost measure in PIR is communication complexity [10].

2.3.2 Local Privacy Enforcement

In this section, we briefly describe the most important privacy-enhancing technologies that enforce privacy on local systems transactions. In particular, we go through Hippocratic Databases and Sticky Policies.

2.3.2.1 Hippocratic Databases

Almost always, privacy-aware techniques use purpose as a central concept around which privacy protection is built. For the transparency requirement, enterprises should declare in their privacy policies the purpose for which data is collected, who can receive it, the length of time (retention time) the data can be retained and the authorized users who can access it. Looking at such policies customers would be able to understand how their personal data will be used and, in case they agree, disclose them. Developed by IBM Almaden Research Center, Hippocratic databases [3, 4] use "purpose" as a central concept and consider it as a special attribute occurring in every table forming the database and associated with each piece of data stored in the database. Therefore, any data that is collected must be collected for a specific purpose and the purpose must be stored with the data. Any query to a Hippocratic database should declare the purpose of accessing the data; the access will be granted only if the query purpose matches the requested-data purpose that is stored in the database.

The authors in [4] employed the current privacy legislations and best practices to derive the necessary privacy principles for Hippocratic databases: purpose specification,

consent, limited collection, limited use, limited disclosure, limited retention, accuracy, openness and compliance.

Hippocratic databases can be used to add enforcement dimension to the P3P initiative (described in section 2.3.3.1). A P3P policy essentially describes the purpose of the collection of information along with the intended recipients and retention period for the collected information. The policy description uses data tags to specify the data items for which the policy is being stated. P3P's concepts of purpose and retention map directly to analogous concepts in Hippocratic databases. It is easy to map each of P3P recipient types into external recipients or authorized users. Thus, P3P policies can be processed through the privacy metadata processor, and generate the corresponding data structures in the Hippocratic database system.

Hippocratic database is one step forward to privacy-aware database system. It sustains the idea of making the data itself privacy–aware and reduces the need for an external agent to control access according to the use purpose.

2.3.2.2 Sticky Policies

HP Labs Bristol in [12] proposed sticky policies that associate "tamper resistant" privacy polices to obfuscated data, along with trusted tracing services. When submitting data to an enterprise, a user consents to the applicable privacy policies along with selected opt-in and opt-out choices. Sticky policies are strictly associated to users' data and drive access control decisions and privacy enforcement.

The concept of sticky policy relies on the concept of coupling the data privacy policy with the data itself on the operating system or even hardware levels. According to [12], it is necessary to express privacy policies that are enforceable across various environments at different levels of computer system: at the operating system or trusted platform level, through the encryption mechanisms such as Identity Based Encryption (IBE) and through Trusted Third Parties (TTPs), and within applications or services. The enforcement mechanisms potentially dictate what type of conditions and rules a policy language should be able to express. The constructs are necessary that make it possible to express the characteristics of possible execution environments, the platform expectations, and operations on different types of data transfer. Environments where confidential data is

used dictate and influence mechanisms that enforce privacy policies. Sticky policies require other privacy protection methods after the data is decrypted by the requester such as trusted platforms and operating systems.

2.3.3 Web Privacy Enforcement

In this section, we briefly describe the most important privacy-enhancing technologies that enforce privacy on web transactions. In particular, we go through P3P, APPEAL, Semi-Automated Derivation of Personal Privacy Preferences, and Privacy seal program.

2.3.3.1 P3P

The Platform for Privacy Preferences Project (P3P) was started in 1998 (first public working draft) by the World Wide Web Consortium (W3C) to define a means to publish website policies in a machine-readable way (in XML format). According to W3C [94], P3P is "emerging as an industry standard providing a simple, automated way for users to gain more control over the use of personal information on Web sites they visit. At its most basic level, P3P is a standardized set of multiple-choice questions, covering all the major aspects of a Web site's privacy policies. Taken together, they present a clear snapshot of how a site handles personal information about its users. P3P-enabled Web sites make this information available in a standard, machine-readable format. P3P enabled browsers can read this snapshot automatically and compare it to the consumer's own set of privacy preferences".

In order to perform a P3P agreement, illustrated in figure 2.2, the visited website must be P3P-enabled site, where two files are stored: a reference file and a policy file. The reference file tell which policy file applies to which resource and the policy file state the privacy policy for the whole site or just a specific part of the site. The user should have a user-agent that can act on his behalf. This user-agent either is built into his browser or is a plug-in to his browser (or it can be a proxy located somewhere else). In addition, the user usually has his privacy preferences stored in his local machine where the user-agent has access to. The privacy preferences are written in a format called APPEL (described in next section).

A typical P3P scenario (figure 2.2) can be summarized in the following steps:

A. The user sets generic preferences, upon which his agent automatically acts.

B. The User encounters a web site with exceptional practices outside his generic preferences.

C. The user-agent requests and gets the website privacy policy reference file as in steps 1 and 2 respectively.

D. The user-agent search in the received reference file for the corresponding visited page privacy policy file, send a request to the web server to get the privacy policy file as in step 3, and get the file as in step 4.

E. The user-agent reads the visited web site P3P policy file automatically and compares it to the consumer's own set of privacy preferences (written in APPEL language) and then gives warning in case of discrepancies as in step 5. The user is prompted if he wishes to consider other alternatives, consent to the exceptional practice, or to go elsewhere.

F. Whether the website privacy policy matches user privacy preferences or the user received discrepancies warning and agree to go forward, the user-agent will take the user to the desired webpage as in step 6 and 7.

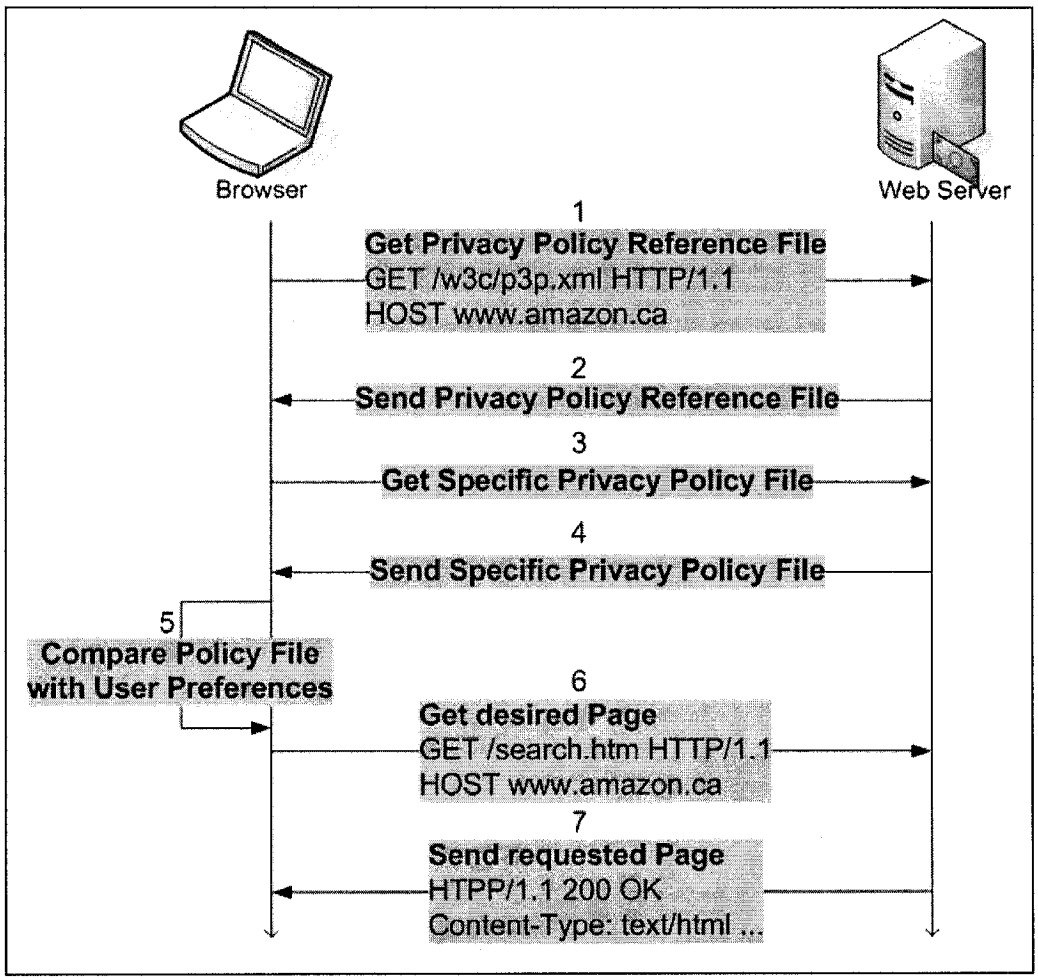

Figure 2.2: P3P user scenario

The following is a typical P3P privacy policy with the definition of each element:

```
<!--Policies: is the root element of policy files; it contains one or more P3P policies -->
<POLICIES xmlns="http://www.w3.org/2002/01/P3Pv1">
    <!-- contains a complete P3P policy -->
    <POLICY name="policy">
        <!-- Entity: gives a precise description of the legal entity making the representation of the
        privacy practices-->
      <ENTITY>
            <!-- contains one or more DATA elements. DATA elements are used to describe the type of
            data that a site collects -->
            <DATA-GROUP>
            <DATA/>
        </DATA-GROUP>
      </ENTITY>
      <!--indicates whether the site provides access to various kinds of information-->
      <ACCESS/>
      <!-- contains one or more DISPUTES elements
      <DISPUTES-GROUP>
```

```
            <!-- Describes the site's procedures for dispute resolution with regard to their privacy
            practices -->
          <DISPUTES>
            <!-- specifies the possible remedies in case a policy breach occurs -->
            <REMEDIES>
          </DISPUTES>
        </DISPUTES-GROUP>
        <!-- describe data practices that are applied to particular types of data -->
          <STATEMENT>
            <!-- Provides a human-readable text summary of the data practices specified in the
            STATEMENT element -->
          <CONSEQUENCE>
            <!-- contains one or more purposes of data collection or uses of data -->
          <PURPOSE/>
          <!-- contains one or more recipients to which the private data may be disclosed  -->
          <RECIPIENT/>
            <!-- indicates the kind of retention policy that applies to the data referenced in that
            statement -->
          <RETENTION/>
        </STATEMENT>
    </POLICY>
</POLICIES>
```

P3P made one step forward in automating websites privacy policies and facilitating automatic privacy matching by user-agent. However, P3P doesn't protect the privacy of users in jurisdictions with insufficient data privacy laws, nor ensures that websites follow their stated privacy policies [59]. Thus, privacy enforcement mechanisms are needed to complement P3P functionality. Other shortcomings of P3P include: no multiple policies per person or across individuals, unclear legal consequences, no mechanism for negotiation, and inability to express accurately detailed or complex privacy policies. Although there are many major websites using P3P, the lack of law enforcement of using P3P and P3P drawbacks limit its adoption tremendously.

2.3.3.2 APPEL

A P3P Preferences Exchange Language (APPEL) is a separate W3C specification designed to provide a standard language to express user privacy preferences. Using the APPEL language, a user can express preferences in a set of preference-rules called a ruleset, which can then be used to make automated or semi-automated decisions regarding the acceptability of machine-readable privacy policies from P3P enabled websites [95].

According to W3C, APPEL aims to achieve the following:

- Sharing and installation of rulesets: sophisticated preferences may be difficult for end-users to specify, even through well-crafted user interfaces. An organization can create a set of recommended preferences for users.
- Communication to agents, search engines, proxies, or other servers: servers of various kinds may wish to tailor their output to better meet users' preferences, as expressed in a ruleset.
- Portability between products: the same ruleset will work with any P3P-APPEL enabled product.

Agrawal et al in [5] showed the limited expressiveness of APPEL. APPEL can specify only what is unacceptable and not what is acceptable for a user. They demonstrate how the APPEL constructs interact with the P3P policy language in unintended ways, making it non-trivial to get even simple preferences right. So it is easy to write a preference that appears correct and find that it does not accomplish the intended goal. Moreover, they showed how simple preferences are surprisingly difficult to express with APPEL and that writing APPEL preferences is error prone. Their proposed solution is XPref, an XPath based language for user preferences that solve some of APPEL's problems.

2.3.3.3 Semi-Automated Derivation of Personal Privacy Preferences

A semi-automated derivation of a personal privacy policy is the use of mechanisms that may be semi-automated to obtain a set of privacy rules for a particular use. Yee and Korba in [96] presented two approaches for such derivations. The first approach relies on third party surveys of user perceptions of data privacy. The second approach is based on retrieval from a community of peers. The authors found that consumers must be able to derive their personal privacy policies easily. They showed how these two approaches are accomplished by comparing to the current privacy preferences languages such as APPEL. Both approaches reflect the privacy sensitivities of the community, giving the consumer confidence that his privacy preferences are interpreted with the best information available. Moreover, in the second approach, the authors suggested having privacy commissioners' offices take responsibility of being privacy watchdog for the consumer.

In their approaches, consumers will be able to continuously update personally identifiable information and privacy policies, based upon policy update feedback to the policy provider from consumers. In addition, users will be changing their policies to suite their desired needs over time, when interacting with different service providers and for different services. At the same time, policy providers can monitor the changes made to policies to find out what needs to be changed in their published privacy policies to match consumers' preferences.

2.3.3.4 Privacy Seal

In a privacy seal program, a trusted third party provides assurance to consumers that personal information is handled according to the businesses privacy policy. Privacy seal providers do not set privacy policies, but rather ensure that a business's privacy policy accurately states what personal information the business gathers and how it is handled.

TRUSTe (founded in 1996) is a non-profit organization that provides privacy seals to web sites that follow a set of privacy specifications. TRUSTe seal sign on a service provider web site presents assurance to consumers that their personal information is handled according to the business privacy policy [11]. TRUSTe investigated many well known businesses for privacy complaints in their websites and granted them the privacy assurance certificate; however, it never revoked the TRUSTe certification from a business even when some privacy violations happened [60]. For instance, Yahoo handled users' personal information collected previously under a different privacy policy and the TRUSTe seal remained valid. Several other companies and organizations as well offer privacy seals. The main privacy seals beside TRUSTe are WebTrust and BBBOnline. Although WebTrust has been in existence for almost as long as TRUSTe, WebTrust has not been widely adopted because of its privacy certificates cost (only 31 recipients by January 2002). In contrast, TRUSTe and BBBOnline have many well-known customers [65].

In July 2000, Cheskin Research reported that TRUSTe seal increased the confidence in a Web site for 55% of the sample and 40% for BBBOnline seal. This is because the same study showed that 69% of Web users recognized the TRUSTe seal (up from 10% in January 1999), and 37% recognizing the BBBOnline seal (up from 18%) [23]. Moores

and Dhillon in [65] highlighted that there are continued calls for legislation to ensure that the good sense outlined by the provisions of each seal becomes a legally binding contract. They suggested that any legislation must ensure that a Web site is legally bound by an acceptable standard of privacy. Privacy legislations may need to enclose businesses obligation to use privacy seals to comply with their published privacy practices together with the need to legislate laws for the seal programs themselves (e.g. when to revoke the privacy seal).

2.3.4 Organizational Privacy Enforcement

In this section, we briefly describe the most important privacy-enhancing technologies that enforce privacy on transactions that are within the organization or between organizations. In particular, we go through EPAL, XACML, and some other purpose-based access control models.

2.3.4.1 EPAL

The Enterprise Privacy Authorization Language (EPAL) developed by IBM is a formal language to specify fine-grained enterprise privacy policies that are exchangeable between applications or enterprises. It concentrates on the core privacy authorization while abstracting from all deployment details such as data model or user authentication [9]. An EPAL policy defines lists of hierarchies of data-categories, user-categories, and purposes, and sets of (privacy) actions, obligations, and conditions. Figure 2.3 shows how EPAL can be used to represent organization's privacy policies in the policy decision point inside the organization.

User-categories are the entities (users/groups) that use collected data. Data-categories define different categories of collected data that are handled differently from a privacy perspective. Purposes model the intended service for which data is used. Actions model how the data is used. Obligations define actions that must be taken by the environment of EPAL. Conditions are Boolean expressions that evaluate the context. These elements are then used to formulate privacy authorization rules that allow or deny actions on data-categories by user-categories for certain purposes under certain

conditions while mandating certain obligations. In order to allow for general rules and exceptions, EPAL rules are sorted by descending precedence [9].

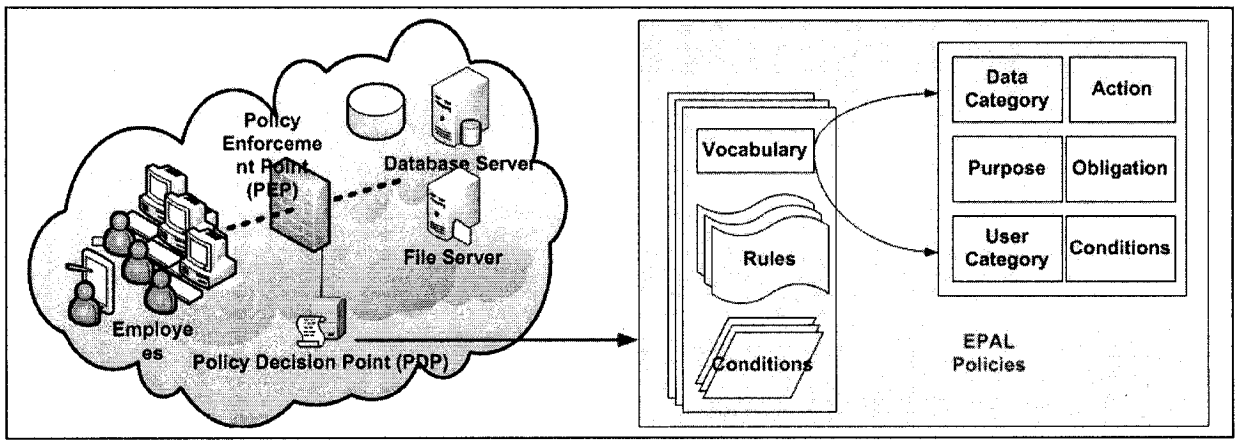

Figure 2.3: EPAL privacy policies in an organization

EPAL is concerned with formalizing enterprise-internal privacy policies only. Nevertheless, an EPAL privacy policy and an engine that evaluates it are not sufficient for privacy enforcement. In addition, EPAL is not intended to set privacy preferences [2]. In a comparison between EPAL and XACML (described in next section 2.3.4.2), Anne Anderson in [8] pointed out some EPAL drawbacks: inconsistent treatment of attributes, limited concept of "role", limited concept of "hierarchical role", only one "vocabulary" per policy, can't express complicated privacy policies, limited hierarchies, and it is not a standard privacy policy language since there is no action by W3C since proposal submission in November 2003. Furthermore, Stufflebeam, et al; in [85] show that EPAL is insufficient for specifying high-level company obligations and it does not allow policy makers to express to whom the disclosure is made, so there are no disclosure recipients. The author in [8] concludes that EPAL contains no privacy-specific features that are not already supported in XACML and that EPAL lacks significant features that are included in XACML and that are important in many enterprise privacy policy situations. Moreover, EPAL supports only a small subset of the functionality offered by XACML while XACML supports most of the EPAL functionality, so XACML can be extended to substitute EPAL.

2.3.4.2 XACML

XACML (eXtensible Access Control Markup Language) is an OASIS open standard XML-based fine-grained access control language. XACML describes an access control policy language, Processing Environment, and a request/response protocol. The policy language is used to express access control policies (who can do what when). The request/response protocol defines queries about whether a particular access should be allowed (requests) and describes answers to those queries (responses). So, XACML includes a policy language and a query language that results in a Permit, Deny, Intermediate (error in query) or Not Applicable response [68]. In fact, it is difficult to have a precise definition for XACML, since it is a general purpose policy system that has several profiles designed to support the needs of different authorization modules. XACML defines four layers to access policy control [68]:

1. The Policy Administration Point (PAP) creates security policies and stores these policies in the appropriate repository.
2. The Policy Enforcement Point (PEP) performs access control by making decision requests and enforcing authorization decisions.
3. Policy Information Point (PIP) serves as the source of attribute values, or the data required for policy evaluation.
4. The Policy Decision Point (PDP) evaluates the applicable policy and renders an authorization decision.

XACML has a domain model expressed in XML Schema. Figure 2.4 shows how XACML processes a service request to retrieve the attributes and policies [68]. The following outlines the data flow in chronological sequence:

1. The policy administrator defines policies and policy sets at the PAP.
2. The service requester issues a request to the PEP to access the specified resource. This requires fetching the attributes and policies associated with the resource, the action, the environment, and tie service requester.
3. The PEP sends the request for access to the XACML context handler in native request format. This may include the details a attributes of the subjects, resources, actions, and environment.

4. The context handler creates an XACML request context and sends a policy evaluation request to the PDP.
5. The PDP queries the context handler for attributes of the subject, resource, action, and environment needed to evaluate the policies.
6. The context handler obtains the attributes either from the request context created in Step 4, or it queries a PIP for the attributes.
7. The PIP returns the requested attributes to the context handler.
8. Optionally, the context handler includes the resource in the context.
9. The context handler returns the requested attributes to the PDP. The PDP continues evaluating the policy as attributes are made available.
10. The policy sends the response context (including the authorization decision) to the context handler.
11. The context handler responds to the PEP, after translating the response context to the native response format of the PEP.
12. The PEP executes any relevant obligations.

The key elements of XACML policy are [68]:

- **Policy Set:** a set of policies or other Policy Sets and a policy-combining algorithm, along with a set of optional obligations.
- **Policies**: represents a single access control policy expressed through a set of Rules. Policies are a set of rules together with a rule combining algorithm and an optional set of obligations. Obligation, are operations specified in a policy or policy set that should be performed in conjunction with enforcing an authorization decision.
- **Rules:** are expressions describing conditions under which resource access requests are to be allowed or denied. They apply to the target, which can specify some combination of particular resources, subjects, or actions. Each rule has an effect ("permit" or "deny") that is the result to be returned if the rule's target and condition are true. Rules can specify a condition using Boolean expressions and a large set of comparison and data-manipulation functions over subject, resource, action, and environment attributes.

- **Target:** is basically a set of simplified conditions for the Subject, Resource, and Action that must be met for a Policy Set, Policy, or Rule to apply to a given request.
- **Attributes:** are named values of know types that may include an issuer identifier or an issue date and time. Attributes are usually considered the characteristics of the Subject, Resource, Action, or Environment in which the access request is made.

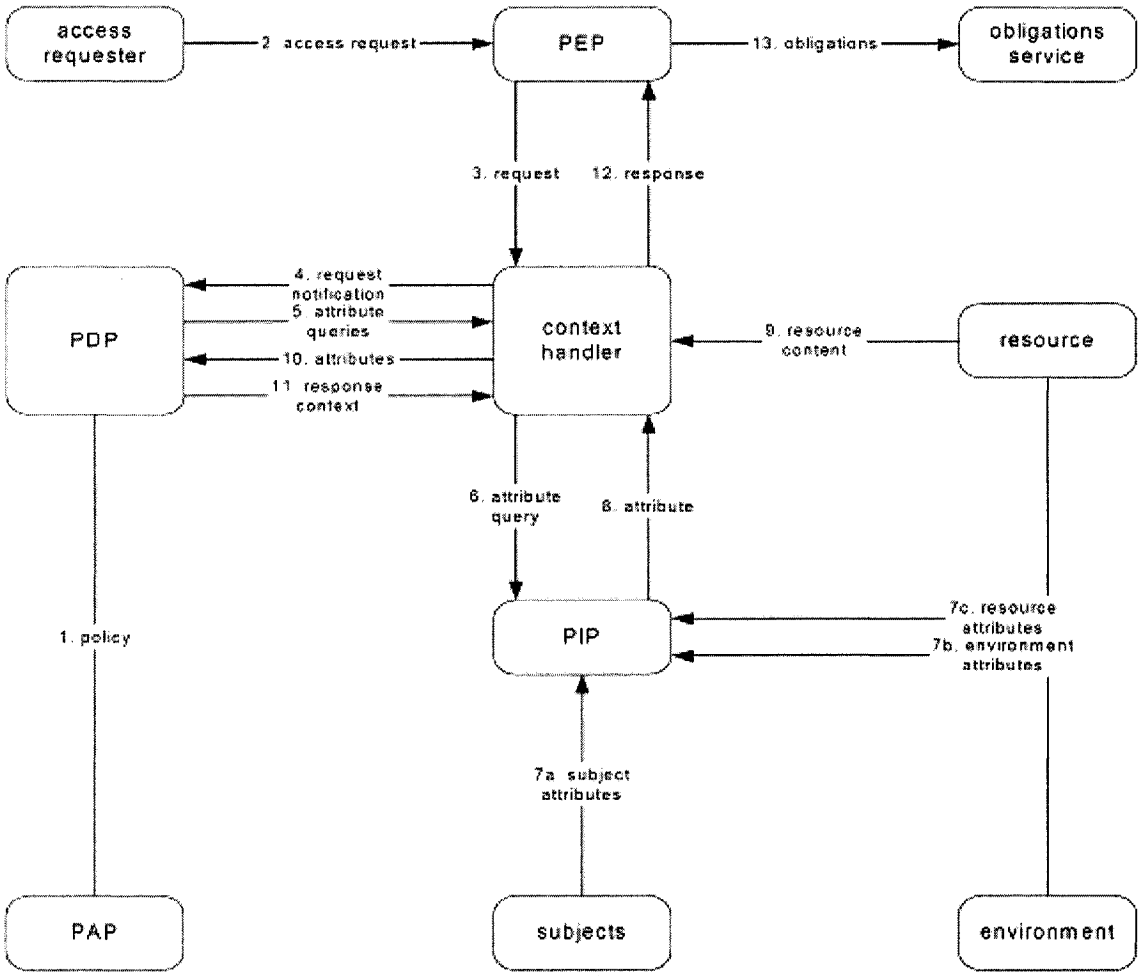

Figure 2.4: XACML Data Flow [68]

XACML 2.0

There are changes in the context schema (e.g. Request may contain more than one Resource), and Changes in the policy schema (e.g. The AnySubject, AnyResource and AnyAction elements no longer exist in XAMCL 2.0; instead, the absence of the

corresponding element will do the same). Most of the syntactic changes aim to make the language more flexible and expressive [28]. Several profiles have been added to XACML 2.0; the following is a brief of the ones that are related to the thesis [68]:

- **SAML Profile:** defines how to use SAML 2.0 (described in section 2.4) to protect, transport, and request XACML schema instances and other information needed by an XACML implementation.
- **RBAC Profile:** The role-based access control profile allows policies to be specified in terms of subject roles instead of individual subject identities. Roles can be nested so that more senior roles inherit the privileges of junior roles.
- **Privacy Profile.** The privacy profile supports specifying data privacy requirements by using two attributes: resource purpose and action purpose. The resource purpose indicates the purpose for which the data resource is collected. The action purpose indicates the purpose for which access to the data resource is requested. This profile is necessary to specify privacy policies using XACML.
- **Multiple Resource Profile.** This profile describes three ways in which a PEP can request authorization decisions for multiple resources in a single request context and how the result of each such authorization decision is represented in the single response context that is returned to the PEP. It also describes two ways in which a PEP can request a single authorization decision in response to a request for all the nodes in a hierarchy.

EPAL vs. XACML

As we referred to in section 2.3.4.1, the author in [8] concludes that EPAL contains no privacy-specific features that are not already supported in XACML and that EPAL lacks significant features that are included in XACML and that are important in many enterprise privacy policy situations.

The following table summarizes the differences between the two languages [8]:

Feature	Difference between EPAL and XACML
Decision request, Rule, Applicability of rules	EPAL supports a subset of XACML
Condition	Equivalent
Nested policies	EPAL does not support
Result conflicts	EPAL supports a subset of XACML
Policy references	EPAL does not support
Vocabulary	XACML supports a subset of EPAL
Attribute mapping	EPAL supports a subset of XACML
Attribute retrieval	Equivalent
XML attribute values	EPAL does not support
Hierarchical roles	EPAL does not support
Hierarchical categories	XACML does not support
Hierarchical resources / XML document resources	EPAL supports a subset of XACML
Subjects with multiple attributes	EPAL supports a subset of XACML
Multiple subjects	EPAL does not support
Purpose attribute	EPAL supports a subset of XACML
Error handling	EPAL does not support
Revision number	Equivalent
Data types, Functions, Obligations	EPAL supports a subset of XACML
Multiple responses	EPAL does not support
Status as a standard	XACML is an OASIS Standard EPAL is not a standard

Table 2.2: Differences between XACML and EPAL [8]

2.3.4.3 Purpose-Based Access Control Models

There are several proposals for improving access control models by making them privacy preserving models. Purpose-based access control models can be used to create privacy policies since purpose is a key element in a privacy policy. Qingfeng He in [42] proposed an extended role-based access control (RBAC) model, called Privacy-Aware Role-Based Access Control (PARBAC) model, for enforcing privacy policies within an organization. He's model combines RBAC, Domain-Type Enforcement (DTE), and privacy protection by modeling business purposes and data policies. Consented consumer privacy preferences are recorded as data policies, which govern how to use actual consumer data. The author suggested enforcing a purpose-binding privacy requirement; so that the actual purpose of a business operation to consumer data is consistent with the purpose consented by the consumer. He's proposal is based on the Dynamic Authorization

Framework for Multiple Authorization Types that combines RBAC, DTE, and E-P3P (from IBM). PARBAC aims to enable enterprises to protect customer data privacy.

Byun, Bertino and Li presented a purpose-based access control suited for hierarchical data [20]. Moreover, they proposed an efficient method for determining access purposes, which uses the notions of role attributes, conditional roles and system attribute. Their proposed solution relies on extending RBAC model.

Karjoth and Schunter proposed a privacy policy model for enterprises which serves as the basis for an internal access control system to handle received data in accordance with privacy standards. Their model protects personal data from privacy violations by means enforcing enterprise-wide privacy policies. They extended Flexible Authorization Framework (FAF) with grantors and obligations and created a privacy control language that includes user consent, obligations, and distributed administration. Conditions impose restrictions on the use of the collected data, such as modeling guardian consent and options. Access decisions are extended with obligations, which list a set of activities that must be executed together with the access request. Grantors allow defining a separation of duty between the security officer and the privacy officer. The proposed model assures data subject providing his personal data that the enterprise receiving the information will handle it according to the stated privacy policy. At the same time, the enterprise may verify that its business practices are not in conflict with the privacy policies posted on its website, which are usually considered a binding contract between the site owner and the people visiting the site [47].

Chiu et al demonstrated the key privacy and access control policies for internal content flow management (such as content editing, approval, and usage) in Financial Enterprise Content Management Systems (FECMS) as well as external access control for Web portal and institutional programmatic users [24].

2.4 Security Assertion Markup Language (SAML)

Security Assertion Markup Language (SAML), an OASIS standard (developed by the Security Services Technical Committee of OASIS), is "an XML-based framework for communicating user authentication, entitlement, and attribute information. It allows

business entities to make assertions regarding the identity, attributes, and entitlements of a subject to other entities, such as a partner company or another enterprise application" [69].

One of the goals of SAML standard is to enhance interoperability of identity management in a secure approach between businesses. SAML can be defined in terms of the following components [69]:

- **Assertion:** a package of information that supplies one or more statements made by a SAML authority where SAML defines three different kinds of assertion statement: authentication, attribute, and authorization decision.
- **Protocols:** a number of request/response protocols that allow service providers to, for example, request or query for an assertion and ask for a subject to be authenticated, create and manage name identifier mappings.
- **Bindings:** mappings from SAML request-response message exchanges into standard messaging or communication protocols. For example, the SAML SOAP Binding defines how SAML protocol messages can be communicated within SOAP messages.
- **Profiles:** defines constraints and/or extensions in support of the usage of SAML for a particular application. For example, the Web Browser SSO Profile specifies how SAML authentication assertions are communicated between an identity provider and service provider to enable single sign-on for a browser user.

SAML standard can be used in different applications. For instance, SAML enables proxy-based true SSO through the communication of an authentication assertion from the identity provider (or authentication provider) to the service provider which, if confident of the origin of the assertion, can choose to log in the user as if they had authenticated directly. Another example of using SAML is in securing web services. SAML assertions can be used within SOAP messages in order to carry security and identity information between actors in web service transactions. The SAML Token Profile of the OASIS WS-Security TC specifies how SAML assertions should be used for this purpose. SAML can also be used in attribute-based authorization model where one web site conveys identity information about a subject to another web site in support of some transaction [69].

SAML is not designed to express privacy policies or to define a new authentication technique. Instead, version two of both XACML and SAML have been designed to complement each other. Therefore, an XACML policy can specify what a provider should do when it receives a SAML assertion, and XACML-based attributes can be expressed in SAML. Moreover, SAML is compatible with several other standards; for example, SAML token profile of WS-Security (OASIS Standard that specifies how SOAP messages can have their integrity and confidentiality ensured) specifies how SAML assertions can be used to provide message security. SAML itself points to WS-Security as an approved mechanism for securing SOAP messages carrying SAML protocol messages and assertions [69].

Many identity federation architectures deploy SAML standard. Liberty Alliance Project had defined its Identity Federation Framework (ID-FF) on the base provided by SAML V1.x, layering additional functionality on top. The Liberty project submitted ID-FF V1.2 back into the OASIS Security Services Technical Committee as input for SAML V2.0 and is already working to base its work on the new technical base formed by the SAML V2.0 OASIS Standard. Another example is Shibboleth architecture that profiled SAML for its particular requirements and built privacy management into its architecture.

Grob in [40] did a security analysis of the SAML single sign-on browser profile, revealing several security flaws in Liberty Alliance specification of this profile. These security flaws allowed several attacks on the protocol. Three attacks that have severe impact are discussed: man-in-the-middle attacks, attacks by information leakage, and message replay (connection hijacking). Countermeasures and solutions to these attacks are proposed. The paper also discusses the vulnerability of an implementation using SSL or TLS channels.

2.5 Identity Federation and Single Sign-On

Users maintain many separate accounts on different Internet businesses and services. Many Websites keep a profile for each visitor; this profile will contain more identifiable information when users register their information (otherwise, the Website may identify the user by other means such as a browser cookie, or an IP address). Users usually need

distinct authentication credentials (e.g. user name and a password) to access their profiles. Managing user profiles is a costly process for both users and service providers. Identity federation aims to assemble virtually an identity from a user's personal information stored across multiple distinct identity management systems. Identity federation defines mechanisms for companies to share identity information between domains. These mechanisms include Single Sign-On (SSO), identity mapping and account linking, directory services, authorization, and management. The SSO mechanism reduces redundant logons whereby a user can authenticate once with a member of a federate group and gain access to the resources of multiple members among the group without signing-on again. In addition to fewer redundant logons, Identity federation has the advantages of reducing the administrative costs of user profiles for service providers and keeping more accurate and up-to-date information about users [84].

Pashalidis gave a taxonomy of SSO systems in [73]. The author classified SSO into two main types, pseudo-SSO and true-SSO. In the former type, the user authenticates herself to the pseudo-SSO component where this component automatically executes the authentication mechanisms that are supported by the different service providers (SPs). Therefore, a separate authentication happens each time the user is logged into a SP. The pseudo-SSO component also manages each SP specific authentication credentials that the user logged in. The pseudo-SSO is categorized further into two types, local pseudo-SSO and proxy-based pseudo-SSO. In the local pseudo-SSO type, the pseudo-SSO component is in the user machine itself. Thus, at the beginning of a session, the user logs in once to the pseudo-SSO component. Later, whenever the user wants to visit one of her previously signed up SPs, the pseudo-SSO component will executes on behalf of the user a SP-specific authentication protocol by giving the corresponding credentials to the SP. in the proxy-based pseudo-SSO, the pseudo-SSO component is on an external proxy server. As in the first type, the user first authenticates herself to the external proxy server. Subsequent user authentication at SPs is redirected to the proxy, which automatically executes the SP specific authentication protocol including supplying the requested credentials. In the second type of SSO system, true-SSO, the user authenticates first to an Authentication Service Provider (ASP). When the user visits a SP and that SP want to verify the user identity, the SP will contact the user ASP and verifies that the ASP did

authenticates the user. In this case, there should be a trust relationship between the user ASP and her SPs. The true-SSO is categorized further into two types, local true-SSO and proxy-based true-SSO. In the local true-SSO, the ASP component is within the user system. So the ASP authenticates the user first and then the user ASP subsequently conveys authentication assertions to relying SPs whenever necessary. This SSO type is rarely used since it requires that user SPs trust the user infrastructure itself. In a proxy-based true SSO architecture, an external server takes the role of the ASP as in figure 2.5. This external server acts as a broker between users and SPs. In this case, both users and SPs have to trust the ASP for the purposes of SSO [73]. The SSO architectures that we will examine in this thesis are from the proxy-based true SSO type.

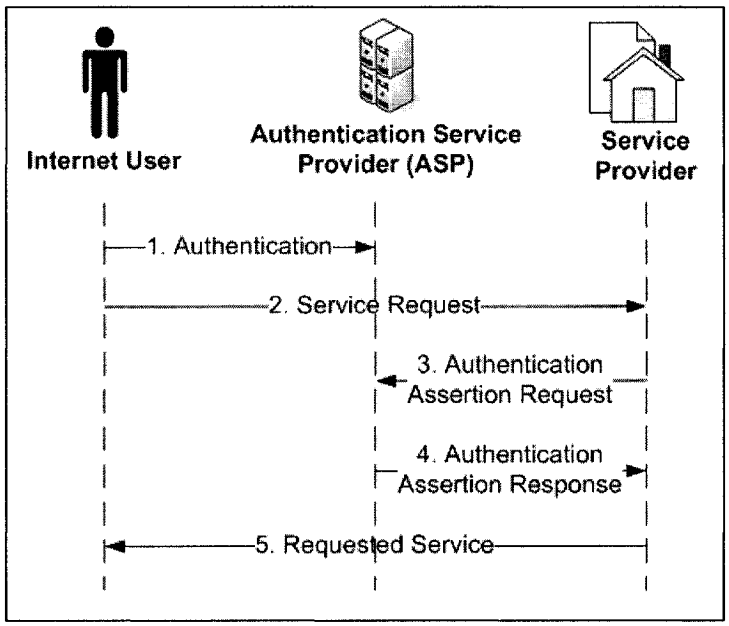

Figure 2.5: Information flows in proxy-based true SSO

2.5.1 Centralized Identity Federation Architectures

In centralized identity federation architecture all user information is kept in one centralized location. In other words, the user has only one identity provider that serves as attribute provider as well. In this case, all the service providers that this user deal with have to contact this single centralized identity and attribute provider in order to authenticate the user or get some of his attributes (as in figure 2.6).

One main advantage of this design is that a user has centralized control and access to his personal information. A user can ensure that his personal information is correct with updating only one location and can control access to his personal information in terms of who can access, under what conditions, for what purposes, and for how long. By having this central repository, it becomes easy to provide complete information to whoever may need it. Another advantage is that personal information can travel with a user, allowing access to that information whenever it is needed. Moreover users can own their personal information [38].

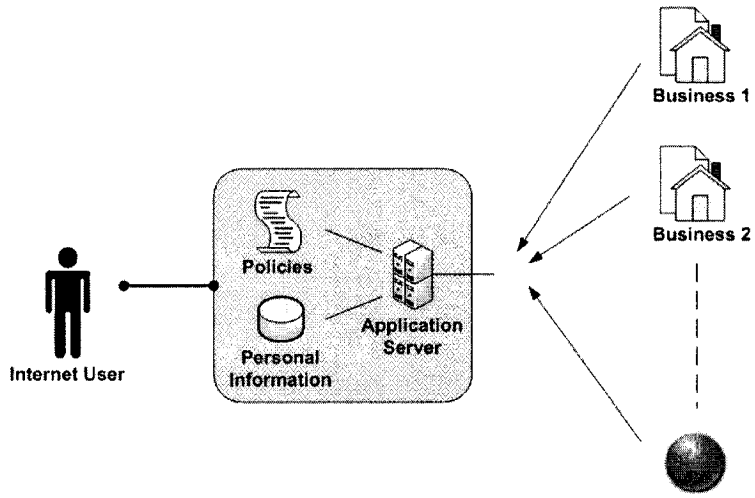

Figure 2.6: Centralized identity federation architecture

On the other hand, there are many disadvantages with this architecture. The centralized identity federation architecture undergoes the single point of failure, having all users' data with one provider. It is also easier for adversaries to perform identity theft since they will get illegal access to all user personal data. Another disadvantage is that the user has the burden of understanding and practicing access control over his data and ensuring its accuracy. Moreover, businesses will no longer be able to perform data mining, unless they have been able to store the information they need [38].

2.5.1.1 Microsoft Passport Network

Microsoft Passport Network which is previously called .NET Passport (the new name is: Windows Live ID), is a centralized SSO service provided by Microsoft corporation that allows users to log in to many websites using only one account which is the Microsoft

one. Microsoft integrated this identity federation architecture in their own operation system (Windows XP), so the users have the option of linking their operating system user account with their other service providers' accounts. Each service provider needs to have business agreement with Microsoft to be a member of Microsoft federate group where Microsoft is the only identity provider [62].

Early in 2001, Deborah Pierce, staff attorney for the Electronic Frontier Foundation, and some privacy advocates pointed out the privacy threat that Passport "terms of use" exposes. The "terms of use" shows that Microsoft has the right of full access and usage to user personal information which contradicts the site's privacy policy. On the same year, Microsoft revised privacy terms. HailStorm, another Microsoft software initiative on top of the Passport, is a way to widely manage personal data as consumers operate computer applications. HailStorm was also criticized because Microsoft's is the only repository of user personal data which make the architecture vulnerable to hacker attacks. On the same time, Microsoft may sell or misuse the users' data [72]. Furthermore, Microsoft Passport does not protect user privacy against what a service provider may do with his personal data. Once the user agrees that a specific service provider share some of his personal data, that service provider can use user data according to its own privacy policy. Although users have access and some control over their personal information that Passport maintains, other personal data that are collected by service providers may not be accessible. This is because each service provider has its own privacy policy that could not enable users to access some of their gathered data. One major drawback of Microsoft Passport is that it does not enable multiple identity management for the user.

Giving the above drawbacks, Microsoft Passport failed to gain user trust or market acceptance. Although Passport has 160 million customers in 2001 [72], the few businesses affiliated to the project (and many well-known businesses withdrawn) and the severe criticism reduce the growth rate of the customers dramatically in the following years. At the end of 2004, Microsoft announced that it was no longer going to pursue Web sites to use its Passport service, which stores consumers' credit-card and other information as Internet users surf from place to place [61]. In addition, Microsoft suspended the use of .NET Passport Wallet, a service to store user's credit card information. In May, 2005, Microsoft rolled out several updates to the Passport service.

Early in 2006, Microsoft rolled out a major upgrade to the Passport services called "Windows Live ID" which we will go throw in the next section.

2.5.1.2 Windows Live ID

Windows Live ID is a new service from Microsoft Corp. that will replace Passport. Windows Live ID will enable the user who has subscribed to multiple Windows Live services to sign in to one, and be simultaneously signed in to all the others. Thus, a person can move from one service to another without repeating the sign-in process [63]. As Passport, Windows Live ID service will be the identity provider and federating authority for third party services and software built on top of it. In April 2006, Microsoft started to replace their customers Passport accounts with Windows Live ID accounts [39]. Windows Live ID service will map federated IDs supplied by other identity providers into a form that works within Microsoft online services. This is done through protocols like WS-Trust, WS-Security (OASIS), and WS-Federation, and industry protocols that can be implemented on any platform. Microsoft is planning to use Windows Live ID as the authentication system for all existing and future Microsoft online services. Windows Live ID accounts can be authenticated using not just traditional user-name/password pairs but different other ways of authentications (e.g. security PIN combinations, and smart cards). The Windows Live ID service also supports specialized mechanisms like the Radius protocol for authentication from cell phones or televisions. Windows Live will be tied with many online services and websites that was not part of Passport service. Windows Live ID gains 300 million potential online users in 2006 (including Passport users) [63].

To overcome the limitation of not supporting multiple identity management in Passport, Windows Live ID service will support multiple identity management by adopting the new service "InfoCard" which is the code name for the Microsoft implementation of an identity selector for their identity federation architecture. Although Windows Live ID service is still considered as a centralized identity federation architecture since there is just one identity provider, InfoCard is designed as the building block of the distributed identity federation architecture. InfoCard will be explained in detail in section 2.5.2.2.

2.5.2 Distributed Identity Federation Architectures

In the distributed identity federation architecture, user personal information is distributed among different Internet organizations and agents. In this design, there is no one identity provider or one attribute provider, so the user can have more than one identity or attribute provider (Figure 2.7).

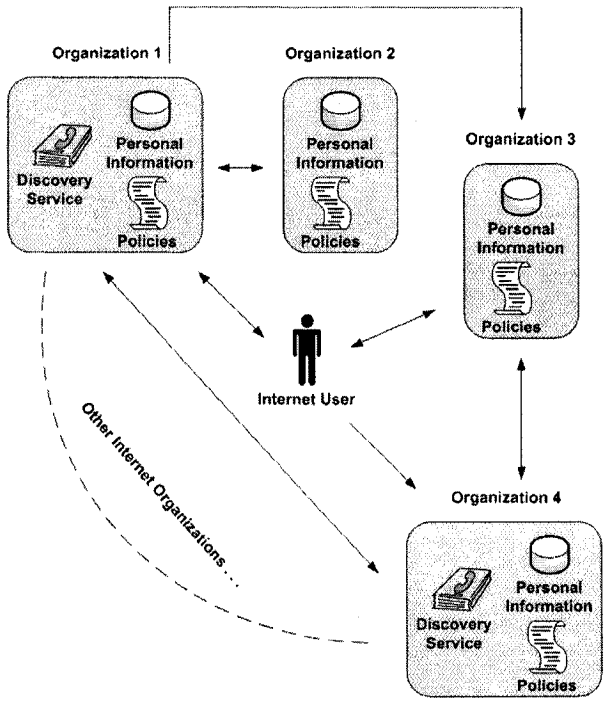

Figure 2.7: Distributed identity federation architecture

Using this architecture, the user can have more than one identity (e.g. one as employee, one as a merchant...). The distributed identity federation architecture has no single point of failure, unless the user uses only a single identity provider and even in this case, the user can still access his local accounts in his service provider websites. Moreover, there is no need for a single trusted third party since the user can have more than one identity provider. As the centralized design, the architecture also enables SSO which reduces redundant logons whereby a user can authenticate once with his identity provider and gain access to the resources in his service providers Web sites.

The user personal information can be scattered among different providers (identity, service, or attribute providers). This burdens the user in updating or regulating access to his personal data. It is also easier for adversaries to perform identity theft once the user credentials with his identity provider are compromised, since adversaries will get illegal access to all related service providers' user accounts.

2.5.2.1 Liberty Alliance Project

The Liberty Alliance project objective is to create open, technical specifications that enable sign-on mechanisms through federated network identification using current and emerging network access devices, and to support a permission-based attribute sharing framework to enable a user's control over the use and disclosure of his or her personal information. The Liberty Alliance project has obtained support from over 150 well known companies and organizations in the last few years and they all were involved in the development of the specifications. The Liberty architecture consists of a multilevel specification set that has three major components. The first component is the Liberty Identity Federation Framework (ID-FF) that defines a framework for federating identities and a mechanism for single sign-on using a federated identity. ID-FF allows a user with multiple accounts at different Liberty enabled sites to link these accounts for future SSO. The second is the Liberty Identity Web Services Framework (ID-WSF) that defines a framework for Web services that allows providers to share users' identities in a permission-based mode. ID-WSF offers features like Permission Based Attribute Sharing, Identity Service Discovery (to discover identity and attribute providers), and Interaction Service (a mechanism to obtain permissions from a user). Third is the Liberty Identity Service Interface Specifications (ID-SIS) that defines service interfaces for each identity-based Web service so that providers can exchange different parts of identity interoperably. These might include services such as registration, contact book, calendar, geo-location, or alerts [87, 93]. We will describe Liberty Alliance project in greater detail in chapter 3.

2.5.2.2 Microsoft InfoCard

InfoCard enables users to manage, control, and exchange their digital identities. Hence, InfoCard will provide users with a single place to manage authentication and payment information as the wallet holds multiple cards (InfoCard characterizes the user electronic identity selector). InfoCard user interface allows a user to choose among several digital identities, each of which is represented visually as a card. InfoCard provides a safer way for users to manage and exchange their digital identity information, helping to protect them from various forms of identity fraud. InfoCard is expected to be available by the

end of 2006 where it will built into Windows Vista, the next version of Microsoft's operating system, and will also be available for Windows XP [16]. The idea of InfoCard was first proposed by Microsoft in March 2005 [41], but the first introductory design document was introduced in April 2006 targeting Microsoft partners to consider using InfoCard in their websites [17].

Although Windows Live ID service (centralized identity federation architecture) depends on using InfoCard for user authorization and attributes exchange, InfoCard is designed as the building block of distributed identity federation architecture. InfoCard supports more than one identity provider not just Microsoft where the identity provider can be the user machine itself. There are two types of InfoCard, one that is issued by an identity provider and one that is issued by the user himself. In the first type, the user visits the identity provider website, the website authenticates the user, and the user enters or specifies some of his personal information to be part of a new identity. The identity provider creates an InfoCard that represents the user new identity and the user either downloads the card from the website or by email. The user can import the card onto any machine from which he want to use it. The InfoCard itself doesn't actually contain any of his identity details. Instead, it carries metadata that describes the shape of the identity. This information includes what claims it has inside it, such as an e-mail address, a home phone, and so forth, and what identity technology it uses such SAML. It also includes who issued it and a unique identifier that the issuer (identity provider) can use to look up the identity and get the values of each of the claims. When the user visits a service provider that needs to know the identity of the user (or may be to authenticate the user), the user needs to choose which card to use for that service provider. The InfoCard system will then make a request to the identity provider who issued the card to get the one claim that the service provider cares about (e.g. the e-mail address). The identity provider returns a SAML token with this information. InfoCard pops up a view that shows the user exactly what information he is about to disclose to the service provider, which is the e-mail address in the SAML token. The user gives his consent which allows InfoCard to pass the token to the service provider. InfoCard is also capable to send only a unique, opaque identifier (Personal Private Identifier (PPID)) that a relying party can use to identify the user without having to know any personal information about him/her [16].

Figure 2.8 shows the interactions between the participating entities and the sequence of message flows between them [64].

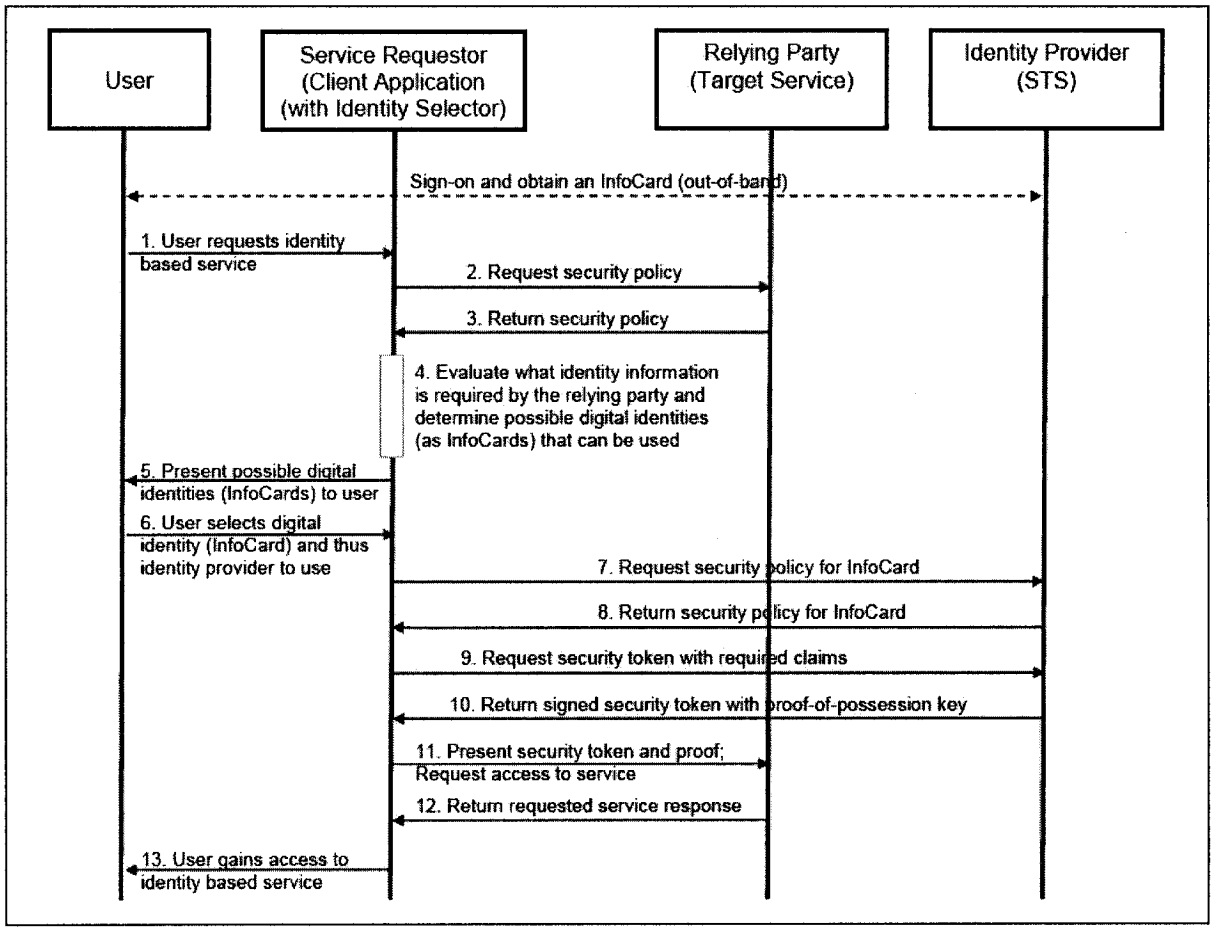

Figure 2.8: InfoCard user case scenario interaction diagram [64]

The second type of InfoCard allows the user to issue his own InfoCards to himself. Traditional Web sites usually allow the user to register with a user name and password. If that Web site accepted self-issued InfoCards, the user could use a self-issued InfoCard to register. There would be no password; just to select a card when prompted, and the Web site would record his PPID and make a note of the long-term key sent in the token. That key would replace the traditional password, and would be unique for every site the user visits. In fact, this key would be computed as a function of a master key, which would be generated randomly for each self-issued InfoCard, and the public key of the relying party. Self-issued cards are a great way to get started with InfoCard. However, using a third-

party identity provider will be much more powerful, because it enables user-constrained sharing of personal information between different systems on the Web [16].

Using InfoCard, the user can get many identity cards (representing different identities) from the same provider and can use the same card with several service providers. InfoCard can be used not just for SSO (mainly authentication) but also to pass some of the user personal data (after his consent). Several drawbacks can be deduced from InfoCard current design. First it is clear that the user loses his digital identities when using a different machine unless he will import his cards in every new machine he will use. Access to the InfoCard client application itself on the user machine is another authentication challenge. It is though too early to evaluate the effect or the success of InfoCards technology knowing that the design is not yet finalized.

2.5.2.3 Shibboleth

Shibboleth is an initiative to develop an open-source implementation, standards-based solution to the needs for organizations to exchange information about their users in a secure and privacy-preserving manner. The organizations that may want to exchange information include higher education, their partners, digital content providers, government agencies, etc. The purpose of the exchange "is typically to determine if a person using a web browser has the permissions to access a resource at a target resource based on information such as being a member of an institution or a particular class. The initiative is facilitated by Internet2, a consortium being led by over 190 universities working in partnership with nearly 100 industry vendors and government to develop and deploy advanced network applications and technologies. Shibboleth architecture is privacy preserving in that it leads with this information, not with an identity, and allows users to determine whether to release additional information about themselves" [44]. Key concepts within Shibboleth include:

- **Federated Administration**: the origin campus provides attribute assertions about that user to the target site. A trust fabric exists between campuses, allowing each site to identify the other speaker, and assign a trust level. Origin sites are responsible for authenticating their users, but can use any reliable means to do this.

- **Access Control Based On Attributes**: access control decisions are made using those assertions. The collection of assertions might include identity, but many situations will not require this.
- **Active Management of Privacy**: the origin site, and the browser user, controls what information is released to the target.
- **Standards Based**. Shibboleth uses OpenSAML for the message and assertion formats, and protocol bindings which is based on SAML.
- **A Framework for Multiple, Scaleable Trust and Policy Sets (Clubs):** Shibboleth uses Clubs to specify a set of parties who have agreed to a common set of policies.

The architecture reduces time needed to manage access to protected resources by enabling SSO capabilities. It focuses on using credential combinations to authenticate while providing the minimum personal identity information transfer possible for the transaction. In general, Shibboleth is related to the high-education community and it is a main contributor to SAML 2.0 specification. We will go through this architecture in more detail in chapter 5.

2.5.2.4 WS-Federation

Web Services Federation Language (WS-Federation) is a specification that defines mechanisms "used to enable identity, account, attribute, authentication, and authorization federation across different trust realms. The specification extends the WS-Trust model to allow attributes and pseudonyms to be integrated into the token issuance mechanism to provide federated identity mapping mechanisms. The models defined in WS-Security, WS-Trust, and WS-Policy provide the basis for federation; WS-Federation extends this foundation by describing how these models are combined to enable richer trust realm mechanisms across and within federations" [27]. BEA, IBM, Microsoft, RSA Security, and VeriSign are involved in the design of this specification. WS-Federation was published on July 2003.

WS-Federation describes how to use the existing Web services security building blocks to provide federation functionality, including trust, single sign-on and attribute

management across a federation. WS-Federation can be divided to three specifications [18]:

- **WS-Federation:** describes how to implement a federation in a Web services world. In particular, WS-Federation focuses on the relationships between parties, and the high-level architecture that supports these relationships.
- **WS-Federation Active:** describes how to implement federation functionality in the active client environment. Leveraging the Web services security stack, WS-Federation Active describes how to implement the advantages of a federation relationship, including single sign-on, in an active client environment.
- **WS-Federation Passive:** describes how to implement federation functionality in a passive client environment. A passive client is one that is not Web services enabled. WS-Fed Passive describes how to leverage the advantages of a federation relationship such as single- sign-on in a passive client environment.

Chapter 3

Privacy Analysis of Liberty Alliance Project

This chapter starts with overview of Liberty Alliance Project. Detailed description of Liberty Alliance frameworks is given in section 3.1. In particular, section 3.1 illustrates the Liberty Identity Federation Framework (ID-FF), the Liberty Identity Web Services Framework (ID-WSF), and the Liberty Identity Service Interface Specifications (ID-SIS). In the next section, 3.2, we go through our privacy analysis for these frameworks. The section begins with listing the main privacy requirements in identity federation systems. Next is a survey of related work in which we summarize and provide discussion of related privacy-analysis of Liberty Alliance identity federation architecture. We then give a detailed use case scenario that integrates the usage of the Liberty frameworks. For each major component in the frameworks, we identify and analyze possible privacy breaches and concerns throughout the scenario. Furthermore, we propose several improvements to the frameworks to enhance consumer privacy.

3.1 Liberty Alliance Project

The Liberty Alliance Project was created in September 2001 (with initial support from Sun Microsystems). The Liberty Alliance is a consortium of technology vendors and consumer-facing enterprises formed to develop an open standard for federated network identity. The Liberty Alliance is delivering specifications and guidelines to enable a network identity infrastructure to resolve several technology and business issues that hindering the deployment of identity-based Web services. The Liberty Alliance project is

based on the concept of enabling users to connect multiple sets of personal information which exist across several e-commerce providers into a single easy-to-manage federated identity. This allows for the convenience of a Single Sign-On (SSO) mechanism as well as easier administration of personal information across multiple service providers. Liberty Alliance is one of the most prominent federated identity standard proposals [54].

The Liberty Alliance project has obtained support from over 150 well known businesses and organizations in the last few years and they were involved in the development of the specifications. The Liberty architecture consists of a multi-level specification set that has three major components. First is the Liberty ID-FF which defines a framework for federating identities and a mechanism for SSO using a federated identity. ID-FF allows a user with multiple accounts at different Liberty-enabled (LE) sites to link these accounts for future SSO. The second component is the Liberty ID-WSF which defines a framework for Web services that allows providers to share users' identities in a permission-based mode. ID-WSF offers features like Permission Based Attribute Sharing, Identity Service Discovery (to discover identity and attribute providers), and Interaction Service (a mechanism to obtain permissions from a user). The third is the Liberty Identity Service Interface Specifications (ID-SIS) that defines service interfaces for each identity-based Web service so that providers can exchange different parts of identity interoperably. These might include services such as registration, contact book, calendar, geo-location, or alerts [50, 52, 54, 93]. The privacy analysis in this paper is for the Liberty ID-FF version 1.2 and ID-WSF version 2.0. The Security Assertion Markup Language (SAML) version 2.0, an OASIS Standard, includes many new features derived from the Liberty ID-FF v1.2 specification that were contributed to the OASIS Security Services TC. Some new key features are the following: the use of pseudonyms, attribute profiles for attribute exchange, single logout, common domain cookie for identity provider discovery, and metadata for expressing SAML configuration. The new added features in SAML V2.0 enable SSO and Identity federation mechanisms. Therefore, SAML V2.0 can supersede the Liberty ID-FF V1.2 [69]. Figure 3.1 shows the different components for each framework. The components that are covered in our privacy analysis are highlighted (ID-FF: Bindings and Profiles, Protocols and Schema; ID-WSF: Data Services Template, Discovery Service, Interaction Service, Authentication

Service, Security Mechanisms, SOAP Binding, Client Profiles). In the following three subsections, we will go through each framework in details.

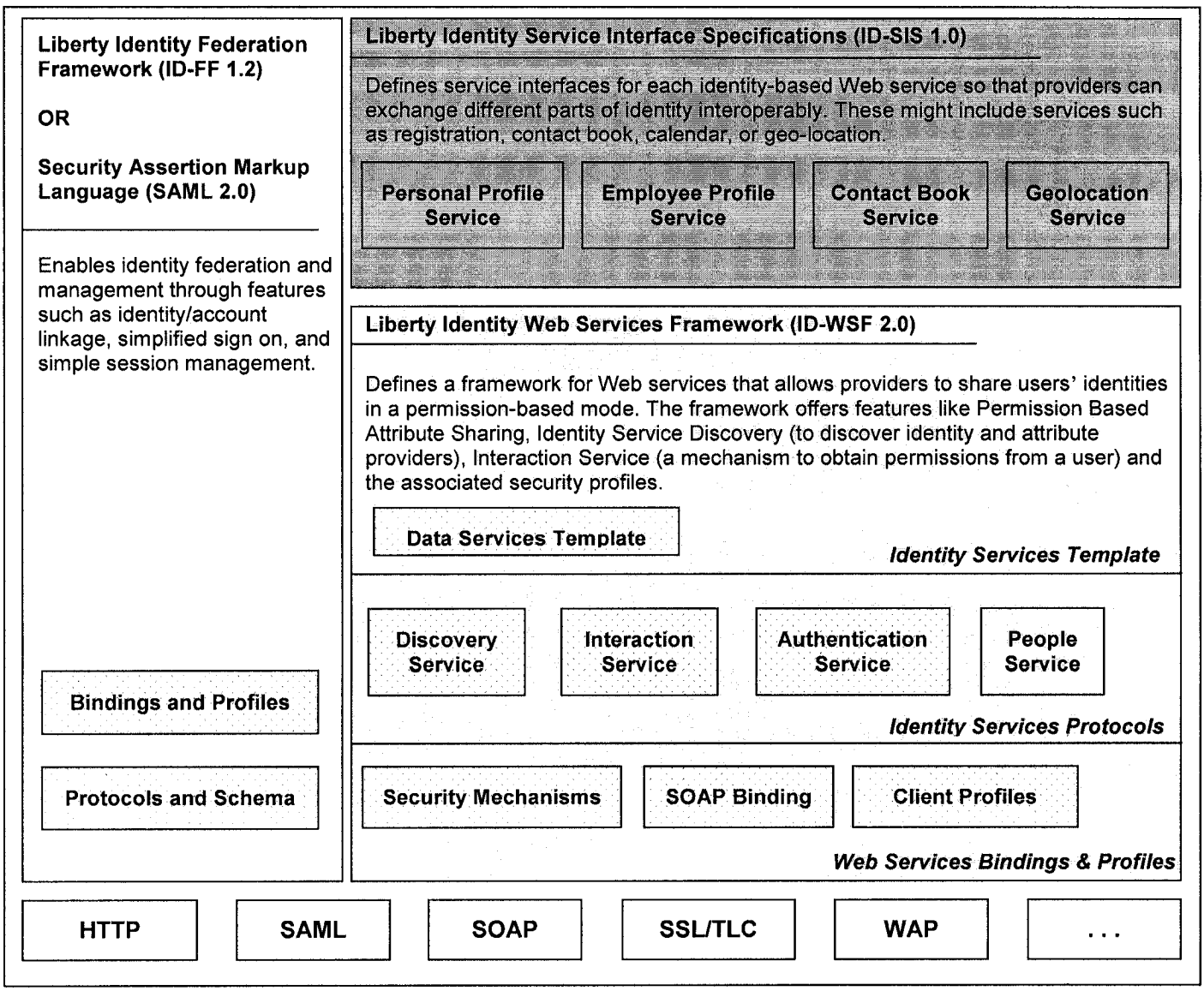

Figure 3.1: Liberty Alliance Frameworks

3.1.1 Liberty Identity Federation Framework (ID-FF)

Identity federation and Single Sign-On (SSO) are the two main features offered by the Liberty ID-FF. Identity federation enables users to link, assemble and control an identity virtually from separate accounts where a user can control sharing of his/her identity

attributes between service providers. Identity federation defines mechanisms for enterprises to share identity information between domains. These mechanisms include single sign-on (SSO), authorization, identity mapping and account linking, and directory services. The SSO mechanism reduces redundant logons whereby a user can login once with a member of a federate group and gain access to resources of multiple members among the group without signing-on again. In addition to fewer redundant logons, identity federation has the advantages of reducing the administrative costs of user profiles for service providers and keeping more accurate and up-to-date information about users [84] [93].

To best describe these features, we will go through the following use case scenario. In this scenario, a sales employee (SE) in a hardware company (compABC) goes to a business exhibition in a different province to promote compABC's new product. SE needs to book a hotel room, so she checks her favorite hotels-search Web site (hotserABC) to find a good hotel deal that is close from the exhibition location. hotserABC identifies SE after she logs in using her credentials (e.g., user name and a password). All that is required is to choose the hotel and specify the booking date where hotserABC will locate her profile and book the room. Moreover, SE gives her consent to hotserABC to introduce her to some members of the affinity group (e.g., car rental Web site). This introduction enables a service provider (SP) to discover which identity providers in the circle of trust have authenticated the user. hotserABC is an LE Web site and it is SE's Identity Provider (IdP) in this scenario. At a later time, she clicks on a car rental company (carrntABC) that is a member of the affinity group (or the circle of trust, CoT). carrntABC, which is SE's service provider, will recognize that the visitor is an LE user and that hotserABC is the visitor's IdP. SE may have a local account with carrntABC and in this case, she will typically login to carrntABC. carrntABC will offer to federate her local identity with her IdP, hotserABC, where she gives her consent to federate.

Identity federation between the two Web sites will enable the use of an SSO mechanism. If SE logs in to hotserABC and then visits carrntABC (while the Web session is still valid with hotserABC), she will not need to login to carrntABC. In fact, carrntABC will get authentication assertions from her IdP (either through browser

redirect or a back-channel) that SE has been already authenticated with the IdP. When SE logs out from hotserABC, the authentication status (logout notice) is sent by hotserABC to carrntABC and all other SPs within the CoT where SE identity federation occurs with her IdP (and which were visited within the same Web session). Furthermore, Identity federation will enable the IdP and SP to exchange SE's personal information attributes upon her permission. For example, carrntABC will get SE's geographical information and perhaps her credit card number from hotserABC in order to conduct the car rental transaction. Identity federation does not imply that IdP will expose user identity by sharing user's identifiable attributes with the SP. The user IdP always shares a unique pseudonym with the SP to identify the user. This unique pseudonym is valid just between these two providers and means nothing to any other provider in the CoT. Therefore, if the user wants to conceal her identity from the SP, her IdP can provide authentication status (and maybe authorization information as well) to the SP by using this unique pseudonym that will refer to the user and preserve her anonymity. In this case, the SP still gets some non-identifiable attributes about the user (e.g., time zone information for better customization). SE can eliminate linkage between her accounts at an identity provider and a service provider, such that the identity provider no longer provides user identity to the service provider, and the service provider no longer accepts user identity from the identity provider. This process is called defederation. Within the same CoT, the user can have multiple identities linked to one or more identity providers. The user can choose which IdP to federate with when she visits a SP. A more detailed scenario will be provided in section 3.1.1 that will show the ID-WSF key features and will reveal more underlying layers.

Web Redirection Options

In the Liberty ID-FF, there are two options for Web redirection. First is HTTP-redirect-based redirection which uses the HTTP redirection class of response of the HTTP protocol and the syntax of URIs to provide a communication channel between identity providers and service providers. The second option is to use a form-POST-based redirection. In this case, the service provider responds by returning an HTML form to the user agent containing an action parameter pointing to the identity provider and a method parameter with the value of POST. The user clicks on the Submit button where the form

and its arbitrary data contents are sent to the identity provider via the HTTP POST method [93].

Common Domain Cookie

In CoT, the service providers need a way to discover which identity providers a user is using. This framework suggests a solution to the introduction problem in which a domain common to the CoT can be used by all parties. Entries within this DNS domain will point to IP addresses specified by each affinity group member. So when a user authenticates with an identity provider, the identity provider redirects the user's browser to the identity provider's instance of a common domain service with a parameter indicating that the user is using that identity provider. The common domain service writes a cookie with that preference and redirects the user's browser back to the identity provider. When the user visits a service provider within the CoT, the service provider can redirect the user's browser to its instance of the common domain service, which reads the cookie and redirects the user's browser back to the service provider with the user's identity provider embedded in the URL and thus available to service provider systems operating within the service provider's typical DNS domain. Thus, the service provider knows with which identity provider the user has authenticated within its CoT for further transactions with the user's IdP [93].

Metadata and Schemas Component

This component includes metadata and schemas various subclasses of information and their formats exchanged between service providers and identity providers, whether via protocol or out of band [93].

3.1.2 Liberty Identity Web Services Framework (ID-WSF)

The Liberty Identity Web Services Framework (ID-WSF) defines a SOAP based invocation framework with a layered architecture. The framework does not specify any contents for the SOAP body, allowing the development of identity services within the context of the Liberty Identity Web Services Framework. The Liberty ID-WSF defines a framework for discovering, and consuming identity services. Moreover, the framework defines a conceptual model which provides relevant terminology for these identity services. As in figure 3.2, an identity service is an abstract notion of a web service that

acts upon some resource to retrieve information about an identity, update information about an identity or perform some action for the benefit of some identities [87].

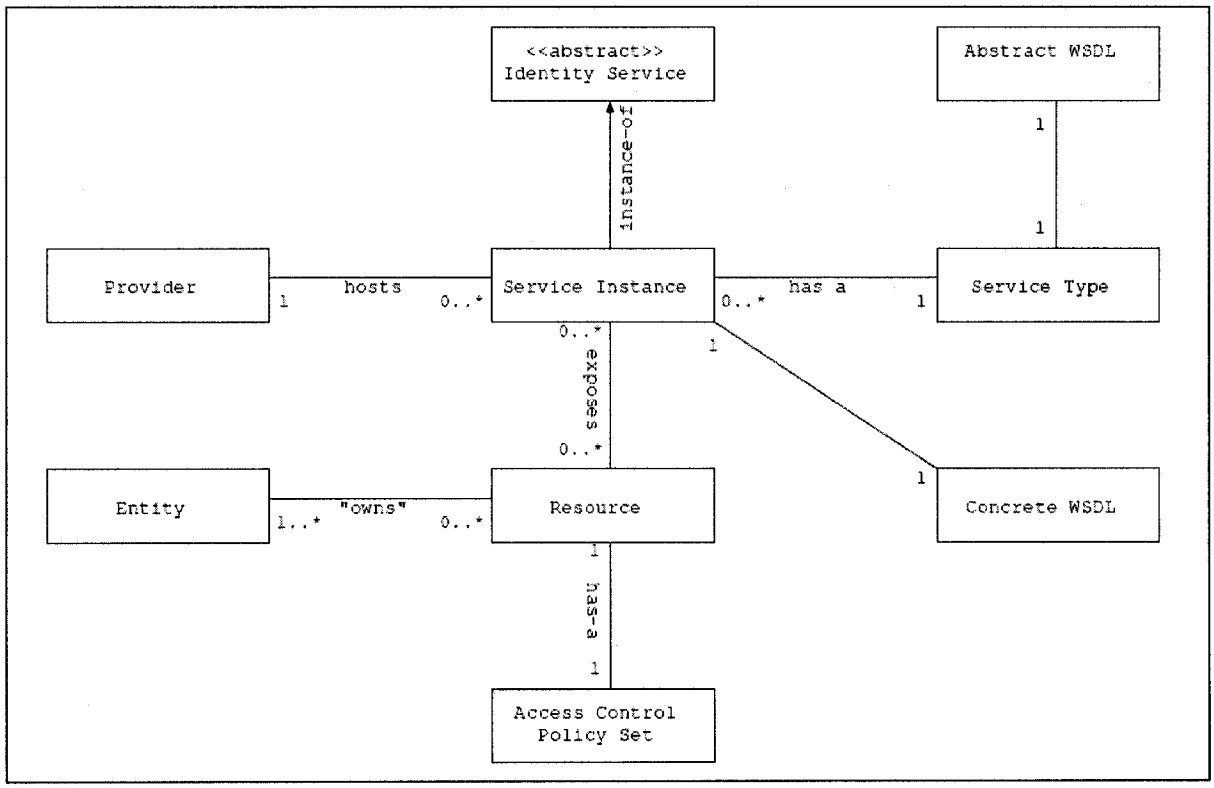

Figure 3.2: Liberty conceptual model [87]

The different types of identity services are identified by a service type identifier. The service type identifier maps to exactly one abstract WSDL definition of a service. A service instance is the instantiation of a particular type of identity service which maps to a concrete WSDL document that contains the protocol endpoint and other information (necessary to communicate with the service instance). Each service instance is hosted by some provider that is identified by a provider identifier [87].

The Liberty ID-WSF defines a SOAP based invocation framework that allows identity services to be discovered and invoked. After a service is discovered and sufficient authorization data has been received from a trusted authority, the invoking entity or the "Web Service Consumer" may invoke the service at the hosting entity or the "Web Service Provider" (figure 3.3).

The framework defines extensions such that service invocation authorization data may be generated by a trusted authority and issued to the invoking system entity (so as to convey the privilege of a system entity to access a resource). The Web Service Provider can make access control decisions based upon this authorization data based upon its business practices and the preferences of the resource owner. In most cases this trusted authority is assumed to be some Identity Provider or Discovery Service [87].

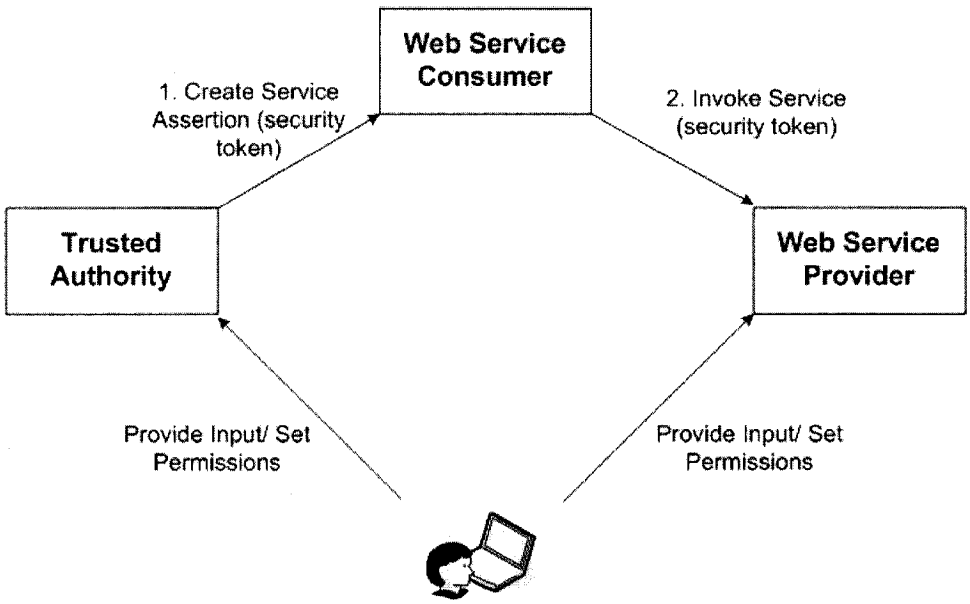

Figure 3.3: Service invocation process

The following are the important components (protocols, profiles, bindings or templates) of the Liberty ID-WSF [87]:

ID-WSF Data Services Template

The ID-WSF Data Services Template provides the building blocks when implementing a data service. The specification defines how to query, create, delete and modify data objects and their attributes stored in a data service, and provides some common attributes for data services. The specification facilitates identity-based web services to be defined using it as a template.

ID-WSF Discovery Service

The ID-WSF Discovery Service defines a core identity service that enables various entities (e.g. Service Providers) to discover a user's registered identity services. Given the criteria of service desired (e.g. service type and security mechanisms), the Discovery Service responds with ID-WSF Endpoint References for the desired identity service, provided that permissions set by the user allow the disclosure of these resources to the relevant entity.

ID-WSF Interaction Service

A provider of identity service may need to obtain permission from a user to allow them to share data with requesting providers. The ID-WSF Interaction Service details protocols and profiles for interactions with the user.

ID-WSF Authentication, Single Sign-On, and Identity Mapping Services

WSC and/or LUAD may need to authenticate with an Identity Provider by exchanging SOAP messages. However, the SOAP specification does not specify any particular security mechanisms. This specification defines an authentication protocol between entities over SOAP, based on a profile of Simple Authentication and Security Layer framework. Moreover, the specification defines Authentication Service that the Identity Provider may offer in the ID-WSF context. The Authentication Service enables WSC and/or LUAD to authenticate with Identity Providers based on the authentication protocol and to obtain ID-WSF security tokens. This specification also defines the Single Sign-On Service which enables WSC and/or LUAD to obtain SAML authentication assertions within the ID-WSF context. Moreover, the specification defines the Identity Mapping Service with which WSC obtains identity tokens for use in web service invocations and referencing principals while preserving privacy.

ID-WSF People Service

In case a user requires accessing the identity service of another user, it is necessary for the user to be able to obtain an identifier for another user from that user's Identity Provider, and to convey that identifier to this second user's identity services. Users may need to grant access rights to their browser-based resources to friends. The People

Service describes architecture for allowing secure, privacy-respecting access by one user to another identity information. This service is not covered in our privacy analysis.

ID-WSF Security Mechanisms

The ID-WSF Security Mechanisms Core describes requirements for securing the discovery and use of identity services. It includes security requirements to both protect privacy and to ensure integrity and confidentiality of messages between Service Providers. The specification defines an identity token that convey an identity between providers. The ID-WSF Security Mechanisms SAML Profile describes profile of the SAML assertion and WS-Security SAML Token Profile in conjunction with the ID-WSF Security Mechanisms.

ID-WSF SOAP Binding

The ID-WSF SOAP Binding provides a SOAP-based invocation framework for identity services. It defines use of WS-Addressing SOAP extensions as well as various SOAP Header blocks and processing rules enabling the invocation of identity services via SOAP requests and responses.

ID-WSF Profiles for Liberty-enabled User Agents/Devices

ID-WSF Profiles for Liberty-enabled User Agents/Devices defines the profiles and requirements for Liberty-enabled clients interacting with the SOAP based authentication service. A Liberty-enabled User Agent or Device (LUAD) is the user agent or device that has specific support for one or more profiles of the Liberty specifications.

3.1.3 Liberty Identity Service Interface Specifications (ID-SIS)

Liberty Identity Service Interface Specifications (ID-SIS) defines service interfaces for each identity-based Web service so that providers can exchange different parts of identity interoperably. The Liberty ID-SIS is a collection of specifications for interoperable services to be built on top of Liberty's ID-WSF. These services include personal profile, employee profile, contact book, calendar, geo-location, and presence. These services are independent and interoperable through implementing Liberty protocols for each specific service. The specifications are built so that it is easy to extend existing Liberty Identity services, or create additional services that build upon the ID-WSF framework. The

current services and future services are designed to be built on top of web services standards, meaning they are accessible via SOAP over HTTP calls, defined by WSDL descriptions, and use agreed-upon schemas [55]. In our privacy analysis, only some parts of the Liberty ID-SIS are covered (especially the parts that interact with the Liberty ID-WSF). The following is a brief description of the major services in the Liberty ID-SIS:

Personal Profile service

The Personal Profile is a Liberty identity service that supports identity information regarding principals themselves. This service defines schemas for basic profile information of a user. This usually includes name, legal identity, legal domicile, home and work addresses and can also include phone numbers, email addresses, and some demographic information, public key details, and other online contact information. The ID-SIS Personal Profile service is an instance of the Data Services Template specification. The Liberty architectural framework specifications ensure that a service properly represents the principal (or that the principal has consented) [50].

Employee Profile service

The Liberty ID-SIS Employee Profile specifies a web service. It offers principle profile information in the context of his or her employment. The Liberty Employee Profile provides basic employee information (e.g. name, work address, employee ID, internal title, department...). The Employee Profile service is an instance of the Data Services Template specification. The Employee Profile service is characterized by the ability to query and update attribute data and incorporates mechanisms from other specifications for access control and conveying data validation information and usage directives [49].

Contact Book service

The Liberty ID-SIS Contact Book specifies a web service offering contact information. This service is an instance of data-oriented identity web service. The Contact Book service allows a principal to manage contacts for private and business acquaintances, friends, family members, and even for herself. The service supports communications applications, acting as principal's phone or address book, allowing her to locate the information needed to contact people and businesses. Using this service, the principal can ship to any address in the principal's Contact Book [48].

Geolocation Service

The Liberty ID-SIS Geolocation is a Liberty identity service which enables the sharing of the geolocation information of a principal. It offers geolocation information regarding a Principal. The geolocation service uses the protocols provided by the Liberty ID-WSF Data Services Template to carry the geolocation requests and responses. The geolocation information includes the position of a principal, speed and direction related information, and information related to the quality of the position information [46].

Presence Service

The Liberty ID-SIS Presence Service implements the data service pattern by hosting information about the Principal's availability, enabling other services to query the Presence Service for that information, and enabling the principal or other authorized entities to modify the principal's presence information. The "presence" refers to a principal's availability for communications over a network (e.g. the availability to talk over a traditional or mobile telephony network, chat over an instant messaging (IM) network, or participate in a video conference) [83].

3.2 Privacy Requirements in Identity Federation

Identity federation architectures have many components that need to satisfy user privacy concerns. The Liberty Alliance project takes into consideration different fair information practices; in particular, the Organization for Economic Co-operation and Development (OECD) and the Online Privacy Alliance (OPA) guidelines. Using these guidelines, Liberty offers the following set of fair information practices [91]:

- **Notice:** Consumer-facing Liberty-Enabled Providers should provide to the user clear notice of who is collecting the information, what information they collect, how they collect it, how they provide choice, access, security, quality, relevance and timeliness to users, whether they disclose the information collected to other entities, and whether other entities are collecting information through them.
- **Choice:** Consumer-facing Liberty-Enabled Providers should offer users choices, to the extent appropriate given the circumstances, regarding what Personally Identifiable

Information (PII) is collected and how the PII is used beyond the use for which the information was provided. In addition, consumer facing Liberty-Enabled Providers should allow users to review, verify, or update consents previously given or denied.

- **User Access to PII:** Consumer-facing Liberty-Enabled Providers that maintain PII should offer, consistent with and as required by relevant law, a user reasonable access to view the non-proprietary PII that it collects from the user or maintains about him.
- **Relevance:** Liberty-Enabled Providers should use PII for the purpose for which it was collected, or the purposes about which the user has consented.
- **Complaint Resolution:** Liberty-Enabled Providers should offer a complaint resolution mechanism for users who believe their PII has been mishandled.
- **Quality:** Consumer-facing Liberty-Enabled Providers that collect and maintain PII should permit users a reasonable opportunity to provide corrections to the PII that is stored by such entities.
- **Security:** Liberty-Enabled Providers should take reasonable steps to protect and provide an adequate level of security for PII.
- **Timeliness:** Liberty-Enabled Providers should retain PII only so long as is necessary or requested and consistent with a retention policy accepted by the user.

An interesting paper from Liberty Alliance Project discussed some implications of EU Data Protection and Privacy Law for organizations implementing the Liberty identity federation and Web services frameworks and specifications, and seeking to establish robust and trusted business frameworks for federated communities (circles of trust) [29]. The paper pointed out to several points that show potential deployments of Liberty technology and demonstrate how the legal principles discussed could have an impact on the participants of the CoT. For example, if there is a transfer of personal data by one participant to another participant outside the EU, the parties will need to consider the impact of the trans-border transfer issues. If consent cannot be used as a ground for permitting the transfer, one of the other grounds will have to be sought. Another potential implication is that participants may wish to make legally binding any policies expressed in Usage Directives and to provide principals direct rights of enforcement to give participants trust and ensure that parties are bound to comply with their stated policies.

3.3 Privacy Analysis of the Liberty Alliance Frameworks

In this section, we go through our privacy analysis for these frameworks. The section begins with a survey of related work in which we summarize and provide discussion of related privacy-analysis of Liberty Alliance identity federation architecture (subsection 3.3.1). Next, we give a detailed user case scenario that integrates the usage of the Liberty frameworks (subsection 3.3.2). For each major component in the frameworks, we identify and analyze possible privacy breaches within the different transactions of the given scenario and discuss proposals for improvement (subsection 3.3.3).

3.3.1 Survey of Related Work

Academic Work

Recently, some academic publications have discussed the security and privacy in the Liberty Alliance project. Pfitzmann [75] evaluated the privacy of the Liberty Alliance phase 1 specifications that concern the browser single sign-on protocol. Later, an update for this paper evaluated the privacy on the same part of the Liberty project but for phase 2 of the specifications. The majority of the privacy concerns in this paper are about the user giving clear consents for transactions that happen between the different providers. A non-technical overview in [58] showed some scenarios by which federated identity management can actually help address certain aspects of the identity theft problem. The paper pointed out that federated identity management connects together previously isolated collections of identity information, which might be perceived as contributing to the identity theft problem because it exacerbates the ramifications of any successful attack. The concern is that if one of the user identity provider accounts is compromised, then all the related service provider federated accounts will be compromised as well. The paper then suggested new mechanisms against identity theft in these scenarios, mainly in the authentication part. The authors in [13] proposed a flexible and privacy-preserving approach that allowed a user to establish a unique identifier and then proceed to establish other complex identity attributes in a federation. A solution to the problem of identity theft based on cryptographic techniques was presented.

Ahn and Lam [6] investigated the privacy issues in federated identity management focusing on the Liberty Alliance project. The authors discussed the privacy requirements

using business scenarios. They proposed a privacy preferences expression language that uses the user preferences language PREP as a basis. The paper did not give any user interface proposal that could be used to store his/her preferences in the suggested customized PREP language. Easy-to-use and clear user interface that give the user full control in specifying privacy preferences for his/her personal information seems to be a significant challenge. The authors in [7] identified information assurance requirements in federated identity management. The paper briefly discussed privacy concerns in federated identity management with the Liberty Alliance project. A security model for authentication and access control for federated systems is described in [86]. The model supports single sign-on for users, a high level of autonomy for database custodians, and low maintenance overhead. The paper is concerned with securing read-only access to sensitive data as it is transmitted and delivered as part of federated database projects. A short survey of privacy issues within current browser-based attribute-exchange protocols is given in [77]; moreover, this paper presented design decisions that are mandatory to fulfill the privacy requirements. Pfitzmann and Waidner [76] gave an overview of the security and privacy properties desirable for the zero-footprint and browser-stateless constraints. The paper proposed a new protocol for browser-based attribute-exchange with better privacy and scalability. The privacy policies for attributes exchange are discussed in detail. In [43], the author first discussed the shortcomings of the existing Attribute Release Policies (ARP). XACML was then recommended as a suitable base language for ARPs. The proposed architecture suggested the integration of XACML ARPs into SAML-based identity providers and it specified the policy evaluation workflows. Gross [40] went through a security analysis of the SAML single sign-on browser profile, revealing several security flaws in the Liberty Alliance specification of this profile. Countermeasures and solutions to these attacks are proposed.

Liberty-related Work

In addition to specification documents, Liberty Alliance has published several non-normative documents pertaining to security and privacy in their multi-level specifications. In [91], the author addressed some privacy laws, privacy and security fair information practices, and implementation guidance for organizations using the Liberty Alliance specifications. In particular, [31, 52] provided an overview of the security and

privacy issues in ID-WSF technology and briefly explained potential security and privacy ramifications of the technology used in ID-WSF. The authors in [56] investigated the topic of identity theft in Liberty Alliance Project and showed how a cross-organizational and a vendor-neutral method of approaching the problem can work where piecemeal approaches will not. Varney and Sheckler [92] gave guidelines to assist businesses deploying Liberty-enabled solutions by identifying and addressing certain privacy and security issues that arise in business-to-consumer applications. The authors in [1] provided a high-level example of how to manage privacy preferences within Liberty Alliance's ID-WSF framework. In this chapter, we look at the Liberty frameworks and identify potential privacy breaches that were not discussed at all, or in much depth in the above work.

3.3.2 Use Case Scenario using Liberty Frameworks

We will use a detailed scenario to show typical message flow between the different parties in Liberty Frameworks. The scenario will integrate the usage of both the Liberty ID-FF and ID-WSF with some functionalities of Liberty ID-SIS. In this scenario, a user (usrABC) deals with an online payment Web site, payABC, that keeps some of the user attributes (e.g., name, address, credit card information, and so on). Several e-commerce Web sites have business relationships with payABC and they are within the same CoT. usrABC wants a cell phone, so she subscribes first with a wireless service provider and then she can buy a cell phone. Therefore she will visit a phone service provider (phnABC) to sign a wireless service contract. Next, usrABC will visit an online electronics store (eleABC) to buy a cell phone that is compatible with phnABC service. The typical sequence diagram for the identity federation process between the identity provider and the service providers is depicted in Figure 3.4 below.

In this scenario, it is assumed that identity federation has occurred between phnABC and payABC, and between eleABC and payABC. Thus, there are business relationships between these Web sites and they are in the same CoT. In step 1, the user visits the phnABC Web site where she is redirected to the payABC Web site since she has not been authenticated (step 2.a and 2.b). payABC will authenticate the user by asking her to provide her credentials in case she has not yet been authenticated. Then, payABC as

usrABC's Identity Provider (IdP) redirects her back to the Service Provider 1 (SP1) Web site (phnABC) with an artifact that points to the corresponding authentication assertion. phABC will use the artifact and send a back-channel "Get" SAML assertion message (a request from SP to the IdP for the user authentication assertion) to the IdP in step 4.a. The IdP (payABC) replies with the corresponding SAML authentication assertion in step 4.b that indicates that the user is authenticated.

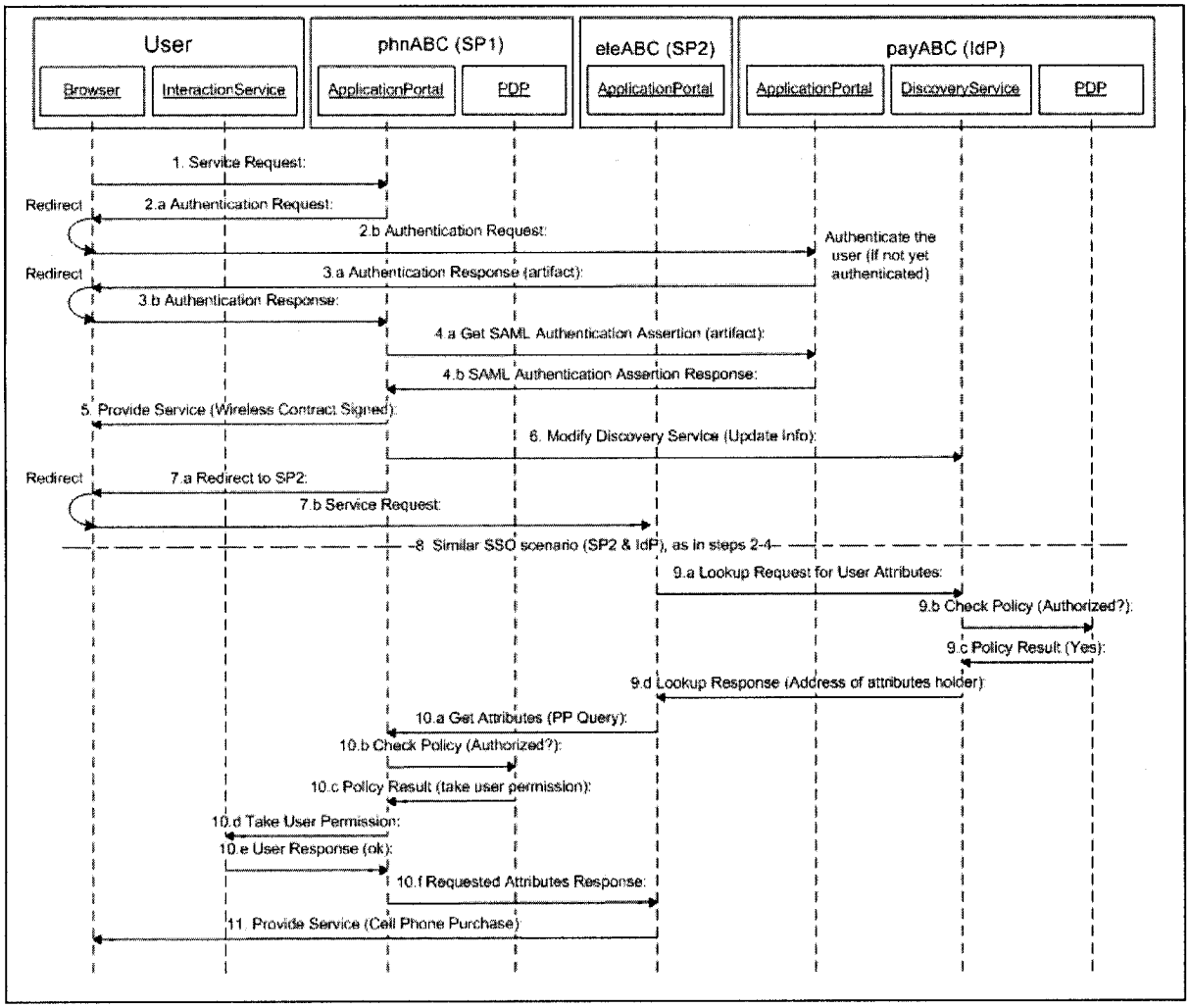

Figure 3.4: User case scenario typical sequence flow with Liberty frameworks

Next, the user will be informed of the SSO confirmation and in step 5, chooses her wireless plan. The SP1 (phnABC) needs some basic information about the user (e.g., name, address, or credit card information) in order to complete the wireless contract. In this case, phnABC either: (a) asks the user for her personal information which is then

provided (e.g., by completing a form); (b) has an old profile of the user (local account) and asks the user for any needed updates; or (c) checks with the discovery service (which is hosted by the IdP in this example) and receives the needed information from the corresponding Attribute Provider (which is the IdP). The way of obtaining user's information is both a design choice and user choice. SP1 maintains the user's attributes (e.g., wireless service type, SIM card compatibility) and is able to act as an Attribute Provider. By being requested by usrABC or by asking permission from the user (either immediately via a Web form if the user is still connected to the Web site, or via the Interaction Service), SP1 registers its resource offering to the Discovery Service (DS). This process is performed by sending an ID-WSF Discovery Service Modify message as in step 6.

Now, since the user needs a cell phone to use the wireless service, phnABC offers the user links to online electronic stores to order the needed device. These stores are expected to have business relationships with phnABC, and most likely they are within the same CoT. The user picks eleABC and she is directed to their Web site in step 7. Note that the user has the choice to visit different electronic stores and it is not necessary to be redirected from phnABC. In step 8, the user is authenticated by her IdP (payABC) using the SSO mechanism in a similar way as steps 2 to 4. After authentication confirmation, the user gets service from SP2. eleABC needs to know more about the user wireless service to offer her the sales promotions on the compatible devices (which are cell phones). SP2 does not maintain user attributes. Therefore, at the request of the user, SP2 tries to retrieve the user's attributes from other Web sites. This process is achieved by sending the ID-WSF Query message (lookup request) to DS in step 9.a. Note that SP2 uses ResourceOffering of DS, which it received from IdP with the ID-FF AuthnResponse (i.e., ResourceOffering of DS is embedded in the ID-FF AuthnResponse that was exchanged earlier). In step 9.b, the IdP acts as a Policy Enforcement Point (PEP) and sends a request to the Policy Decision Point (PDP) to find out whether SP2 is authorized to get information about the attribute provider possessing the user attributes (e.g. WSDL for the desired service). The access policy set by the user in the IdP PDP allows this in step 9.c. In this scenario, the IdP is both a PEP and a PDP. It is possible that the PDP service is hosted by another service or identity provider within the same CoT. In step 9.d,

the DS responds to SP2 with an ID-WSF Discovery Service QueryResponse in which ResourceOfferings (by SP1) that match with specified ResourceID and ServiceType are embedded.

SP2 receives SP1's ResourceOffering, and sends an ID-SIS Personal Profile Query message to SP1 in step 10.a (Get Attributes), to receive the necessary attributes of the user. This message is defined in the ID-WSF Data Service Template specification. SP1 checks its local policy by sending an authorization request to the hosted PDP service as in step 10.b (SP1 is both a PEP and a PDP). Since the user never gave permission for SP2, there is no permit or deny policy result. Therefore, SP1 requires the user permission first, if she is available online. SP1 sends a request for user permission in step 10.d to the Interaction Service (IS). IS is often a user agent that enables providers to interact with the owner of a resource to obtain his/her consent for particular resource exposure. After the user gives consent in step 10.e, SP1 responds to SP2 with an ID-SIS Personal Profile QueryResponse message in which the user's attributes (e.g., SIM type) are embedded (step 10.f). The user will then be able to see the sales promotions on compatible cell phones and purchase the one she likes. Likewise, SP2 can obtain needed user information (e.g., name, address, or credit card information) by requesting the user IdP or contacting the user SP1.

In general, service providers can register some of the user's attributes with the user DS upon his/her permission/request. For instance, browsing or shopping preferences may be kept at the SP. In the Liberty Alliance project, steps 2, 3, 4, and 8 are defined by the Liberty ID-FF specification; steps 6, 9, 10.b, 10.c, 10.d, and 10.e are defined by the Liberty ID-WSF specification; and steps 10.a and 10.f are defined by the Liberty ID-SIS specification.

3.3.3 Consumer-Privacy Breaches in the Liberty Frameworks: Identification, Analysis, and Proposals

In this section, we identify and analyze possible consumer-privacy breaches and concerns within the Liberty ID-FF and ID-WSF frameworks throughout the scenario given in 3.3.1. Furthermore, we propose several improvements to the frameworks to enhance consumer privacy. Most of the privacy issues discussed here are not identified within

Liberty specification documents or non-normative security and privacy documents (e.g. [52, 91]). In addition, we will clarify any privacy concerns that are not clearly explained.

In this privacy analysis, most of the identified possible privacy breaches are not necessarily because of direct privacy flaws in the Liberty specifications. In fact, the various design options in resolving the non-determinism of the Liberty specifications are what could cause the majority of these privacy breaches. Moreover, some of these breaches are indirect results of privacy weaknesses in Internet protocols and browsers, upon which the Liberty specifications are built. The proposed solutions to the identified privacy breaches are not intended to be complete or fully-specified solutions. Many proposed solutions are either recommendations or improvements to enhance consumer privacy in a general sense. The identified privacy breaches and concerns will persuade in finding a more privacy-aware, strict, and comprehensive Liberty specifications that will enhance consumer privacy in federated frameworks.

In the CoT, the user usually trusts some providers more than others. Identity providers are usually the most trusted parties. In fact, the user chooses the IdP because she trusts it more than other providers. The same case applies to the user attribute provider. IdPs themselves trust some SPs more than others and so they deal with them differently according to the SPs' security and privacy policies and practices. Moreover, providers interact with the Liberty services differently according to the trustworthiness of the provider hosting the service. We will assume this in the discussion below. For each subsection, we will list the possible privacy breaches together with the proposed solutions/recommendations.

A. Identity Federation

A.1. The idea of IdP introducing the user to members of the affinity group seems a simple direct concept, but this introduction could lead to a privacy contravention. It is not sufficient that IdP gets a general user consent to introduce the user. Some SPs' privacy polices may not match with the user privacy preferences. In this case, giving only one general consent may lead to a privacy breach.

- *Proposed Solution.* The IdP needs to get a user consent for every single introduction with a member in the CoT. Moreover, the user needs to have the choice of knowing the

privacy practices of every member that s/he will be introduced to and whether it matches the user's privacy preferences or not. This seems to be a lot of work for the user (since one of the Liberty objectives is to enable simplified and fast user sign-on through federated network identification). However, aside from the fact that this introduction only happens once, there are several ways to facilitate the verification of providers' privacy polices by the user. If the user can specify the privacy preferences in advance and there is an automated mechanism to compare SP privacy policy with the user privacy preferences, then this can act in place of the user and give warnings in case of discrepancies. If it is difficult to go through this process before the introduction step, then this can be done when the user visits the SP and before federation. Adding a new member to the affinity group at a later time requires obtaining another consent from the user for introduction. In this case, the IdP will either interact with the user via the Interaction Service, or wait for the user login in order to get the introduction permission for the new member. The IdP should enable a mechanism for the user to opt-out from the introduction consent for each member independently.

A.2. Some design options in the Liberty specification enable SPs (especially the ones to whom a user agrees to be introduced) to know some basic information about the user even before federating the identity with the user IdP. Examples of this information are user IdPs list, user preferred IdP (or the most recently established IdP session which is the last one in the list), and other introduction details. Moreover, SPs could exchange such information with each other.

- *Proposed Solution.* Introduction information should remain private (by both the user and his IdP) regardless of what introduction technique is used, either via the Identity Provider Introduction Profile (i.e. Common Domain Cookie) or when the user agent is a LE client or proxy (LECP). When the user gives his consent to be introduced to a SP, the SP should not know any information about this introduction until the user visits the SP website and he wants to federate. For example, if the user has more than one IdP, then the SP should not know the user's preferred IdP unless the user wants to federate his identity with the SP. The SP (where the user federated his identity) should protect the privacy of this information (user IdPs list and user preferred IdP). Furthermore, other SPs should not even know that the user has given consent to be introduced to a specific SP.

A.3. Nonrepudiable clear user consent for Identity federation between IdP and SP is another important requirement. Giving this consent to the SP side may cause some privacy impacts. This is because any SP is always trusted less than other providers and it is not easy to prove that the user has given consent for the identity federation between SP and IdP.

- *Proposed Solution.* It is always preferable that the user gives this consent to the IdP side [75]. Therefore, there should be a mechanism to enable the SP to interact with the user IdP through a back-channel and request a user consent. Then the IdP will contact the user via the user agent interaction service as defined in ID-WSF. In this case, the IdP will probably authenticate the user first and then get a nonrepudiable user consent.

A.4. Untrustworthy IdP: the current Liberty alliance specifications give the user IdP the power to trace and track most of the user actions across all related SPs in the same CoT. Therefore, the user IdP could check and cross-profile user interactions with SPs in the same CoT in real time [57]. Furthermore, the user IdP can find out when the user access the SP website (and probably can figure out that the user logs out from the SP when the user logs in to a different SP) and when the user gives permission to the SP to get some personal information about him/her (in case the discovery service is hosted by the user IdP, which is the typical case). Having this kind of capabilities with Liberty identity federation architecture, the user IdP can be a serious privacy risk if it is an untrustworthy IdP. In addition, the IdP can impersonate the user and get access to all his SPs accounts within the same CoT.

- *Proposed Solution.* The first burden is on the user himself. With the current Liberty architecture, the user should be careful in choosing trusted IdPs. Moreover, the user may have more than one IdP in one or more CoT. If the user is not sure how trustworthy is his IdP, it is preferable not to host the discovery service or the interaction service in the IdP. The existence of a privacy seal trusted third party that can certify and monitor IdP privacy policies, monitor privacy practices, and monitor cooperation, can help the user in choosing the right IdP. In addition, an audit trail party can monitor the user IdP privacy practices and warn the user in case of any privacy-violation.

A.5. Connecting CoTs: in many cases, the user is involved in two or more CoT and the user wants to connect his identities between the different CoTs. In Liberty specification,

this is usually done by connecting the user IdPs in each CoT through a new main IdP. The new IdP can cross-profile and monitor user interactions with SPs in all connected CoTs. As in A.4, this new IdP could be a main user-privacy-invasive party.
- *Proposed Solution.* As we suggested in A.4, an audit-trail and privacy-seal services can help effectively in retreating this user-privacy concern.

B. Single Sign-On

B.1. Federation domain cookie: one of the design choices in ID-FF is to use a federation common domain cookie. This cookie can be used to find out whether the user has been authenticated recently by the IdP. Note that the most recently established identity provider session is the last one in the list. This cookie is accessible by any federated SP in the CoT (when the user visits the SP Web site). This represents a privacy breach since other members in the CoT do not need to always know the user's authentication status (the most recent IdP that authenticated the user). For example, in our scenario, if the user has a local account with SP2 and wants to access the SP2 Web site using only her local credentials, then it is not necessary for SP2 to know her authentication status with her IdP.
- *Proposed Solution.* This cookie should not be used to reveal user authentication status (the most recent IdP that authenticated the user). Moreover, this cookie should not divulge the user's preferred IdP or any other information (other than a list of user IdPs). If we remove the restriction that the most recently established IdP session should be the last one in the common domain cookie IdPs list (so the list becomes random), then the visited SP would not be able to discover the last IdP the user logged into. SP should always be able to contact the user IdP through a back-channel and request user authentication status if necessary.

B.2. Browser redirect for SSO: The ID-FF specification has the option for browser-redirect messages to carry some information. This information may contain users' personal information (either identifiable or not). Since redirect message length is limited and it is not usually encrypted, this is a privacy breach (in case of eavesdropping attacks).
- *Proposed Solution.* There should be a strict rule to not include any valuable information in the redirect itself. For example, in step 3.a, the artifact that comes with the

authentication response redirect should not contain any user PII. The artifact should be always an arbitrary number that is known only to the IdP. Any PII that the IdP needs to send to the SP should be through a back-channel between them using encrypted SAML assertions.

B.3. Redirection between SPs: When the user is redirected from SP1 to SP2, SP2 will know that the user came from SP1. This is a potential user privacy breach as an indirect result of privacy weakness in Internet protocols and browsers. Let us assume that SP1 and SP2 have different access policies to the user attributes stored in her IdP, so SP1 may get a user attribute that SP2 is not authorized to get and vice versa. SP1 gets a non-identifiable attribute att1 from the user IdP according to the IdP access control policy and SP2 gets a non-identifiable attribute att2 from the user IdP according to the corresponding access control policy. SP1 is not authorized to get att2 and SP2 is not authorized to get att1 where none of them is intended to know any PII about the user. The IdP uses a unique user pseudonym to deal with each SP, so SP1 and SP2 have no way to link the user. However if SP2 knows that the user was redirected from SP1, then SP2 knows it is the same user and they can exchange att1 and att2 illegally. Moreover, knowing att1 and att2 could lead to deducing identifiable attributes and may reveal user identity. In addition, SP2 may show customized ads for the user knowing that she just visited phnABC.

- *Proposed Solution.* This situation requires more attention from the user IdP. For example, the IdP may decide not to expose att2 to SP2 when it notices successive authentication requests (SP1 followed by SP2). An audit trail mechanism at either a trusted third party or the IdP could help in discovering such cooperation so that the appropriate actions are taken.

B.4. Authentication information: federated SPs have the right to re-authenticate the user (via his IdP) whenever they want. Moreover, they can query the user IdP for the authentication method and other related detail (e.g., password length). This information helps SPs to evaluate the authentication mechanism. The SP may ask for stronger authentication or ask the user to re-authenticate before carrying out some transactions at the SP. However, this information can be a threat to the user privacy and security in the

case of a malicious SP. The authentication method information and authentication repetition can help the attacker to figure out user access credentials.

- *Proposed Solution.* To limit these consequences, the user should have control over which SPs can get this information, either through direct consent or through her privacy preferences. Furthermore, the user IdP may warn the user after a specific number of re-authentications within the same session, or apply strict security mechanisms for further re-authentications.

B.5. Pseudonyms exploitation: the SP that deals with a non-identifiable user (using only Liberty unique pseudonym that is shared and valid just between the user IdP and the SP to refer to the user) can exploit the fact that the user is non-identifiable to it to request many unnecessary non-identifiable user-attributes. In this case, the SP could not sufficiently adhere to some basic privacy requirements with the excuse that the user is not identifiable, for example, the "Relevance", "Notice" or "Choice" privacy requirements. The SP could also collect some non-identifiable personal information without obtaining the appropriate user consent, using the pseudonym mechanism in Liberty as an excuse since the SP can not link the user to his real identity.

- *Proposed Solution.* There should be extra caution when the user is anonymous to the SP. The user IdPs and attribute providers should make sure that the SP indeed took a clear consent from the user before releasing any user attributes. Moreover, the existence of a mandatory trusted third party (audit trail party) that can monitor SP privacy policies and monitor privacy practices can help in diminishing this privacy concern.

C. Discovery Service

C.1. Using the current ID-WSF specification, the SP can request resource-holder address for more than one user attribute within the same request message as in step 9.a; however the SP may use only one Usage Directive SOAP header to specify only one usage purpose and other usage information (e.g., retention time). This may lead to a privacy breach since the declared purpose may apply only to some requested attributes.

- *Proposed Solution.* There must be a separate Usage Directive SOAP header for each attribute resource-holder address where the attribute requester must send a request that

contains the purpose for the request, the retention period, and other necessary usage information. The response should contain elements for any privacy obligations that are to be imposed upon the requester. Alternatively, if the SP will request more than one address in one lookup request message with one Usage Directive header, then there should be a standard way to express more than one purpose and other usage information within the single usage field of the lookup request.

C.2. If the user policy within DS PDP allows the SP to get the address of the user attribute resource holder, the SP may locally store the address information. When the user changes the DS PDP at a later time, the same SP may no longer be allowed to get the address of the resource holder. However, the SP still has the resource address stored locally.

- *Proposed Solution.* The SP must not store the resource address after getting the needed resources. The existence of a mandatory trusted third party that can monitor SP privacy policies and monitor privacy practices can help in diminishing this privacy breach. Another option is that the DS should give the SP the resource address in an artifact with a timestamp (signed by the DS). So that the resource holder (Attribute Provider) will accept the SP attribute request only if the artifact is not expired.

C.3. Privacy expression language: Lack of standard privacy expressions could lead to inconsistent interpretation of data privacy directives.

- *Proposed Solution.* A standard fine-grained, machine-readable, and dynamic privacy expression language (e.g., XACML) will enable providers to understand the syntax and semantics of privacy elements. The standard language will be used by both the PDP at the discovery service and the PDP at the attribute provider.

D. Interaction Service

D.1. With the current ID-WSF specification, SP is able to deny its query to the user, as well as user consent (or user deny) returned. There is no suggested way that enables the user or a trusted third party to verify this exchange.

- *Proposed Solution.* SP queries to the user should be recorded by the interaction service. SPs should digitally sign each query to the user; moreover, SPs should sign a

confirmation of receipt for the user consent (or denial) together with the request itself (with timestamp). This provides an audit trail mechanism in case of any dispute.

D.2. The user SP (or may be the user Attribute Provider or IdP) can fabricate user consent. Moreover, an unencrypted channel between the user and one of her providers may enable the attacker to illegitimately post a user consent. In the ID-WSF IS specification, the <InteractionRequest> can include a "signed" attribute which indicates that recipient (e.g., IS) should attempt to obtain a signed <InteractionStatement> from the user, so the SP can ask the user to sign his consent. However, it is not clear how the user can control the integrity of either the request or the response. The user has no direct way to force the SP to sign a request for consent. Moreover, the user can not sign the response if the SP did not ask for the signature. This enables user non-repudiation (user can not deny his consent) but not SP non-repudiation.

- *Proposed Solution.* There must be a mechanism in the specification to enable users to control the integrity of consent request messages. Furthermore, it is usually hard for users to manage digital signature mechanisms. Hence, it is more appropriate that IS signs user consents on behalf of the user. This however requires that the user logs in to the IS first (e.g., userID and password) before giving his consent. In this case, the IS must be trusted by the user.

D.3. ISs hosted by other providers may have privacy impacts.

- *Proposed Solution.* If the IS is not hosted by the user agent itself then the provider hosting the user IS should be very trusted by the user. The fact that the interaction service is responsible for gathering user consents makes it a very sensitive service. User IdP is one trusted provider that could host user IS.

E. User Attribute Access Control

E.1. SP cooperation: if the user deals with two SPs, and each of them knows some identifiable attributes of a user, then there is a risk of attribute exchange between them.

- *Proposed Solution.* An audit trail mechanism can help in exposing such leakages. The fact that the user can check her personal information usage will enable her to discover any unauthorized attribute sharing. The existence of a trusted third party for the audit trail

will give more confidence to the user. In addition, the existence of a seal service (as proposed in chapter 4) will enable the user to carefully select her SPs.

E.2. Attribute deduction: if an SP collects more than one non-identifiable attribute about the same user for different needs on different sessions, this SP may be able to deduce an identifiable user attribute that leads to user identity. Furthermore, if more than one SP collects non-identifiable attributes about the user and they are able to infer that it is the same user, then they may work together to deduce an identifiable user attribute.
- *Proposed Solution.* A privacy seal trusted third party that can certify and monitor SP privacy policies, monitor privacy practices, and monitor cooperation, can help to diminish this privacy concern. More privacy precautions are also needed by attribute providers.

E.3. ID-WSF architecture enables an SP to request more than one user attribute from the attribute provider within the same request message as in step 10.a (the provided user scenario in section 3.3.1); however the attribute requester may use only one Usage Directive SOAP header to specify the usage purpose and other usage information. This may violate user privacy since the declared purpose and other directives may apply only to some requested attributes. Moreover, the attribute provider response could have only one field for attribute obligations and other usage directives.
- *Proposed Solution.* There must be a separate Usage Directive SOAP header for each attribute, and the attribute requester must send a request that contains the purpose for each attribute and any other necessary data privacy directives (preferably using a standard usage directive language). The response should contain elements for any privacy obligations that are to be imposed upon the requester. (See Discovery Service privacy concern above.)

E.4. If the attribute provider relies on the discovery service to be the PEP, then this could lead to a privacy violation. The SP can reuse or share the information about the attribute provider holding the user's attributes, so the same SP or other providers may access the user's attributes illegitimately.

- *Proposed Solution.* Both the discovery service and the attribute provider should act as a PEP. At the attribute provider, the PEP must always be designed as a back-line guard by the entity hosting or exposing the resource (as in step 10). On the other hand, the discovery service PEP acts as a first-line guard for user information access (as in step 9).

E.5. If there is a conflict between the attribute provider local access control policy and the user privacy policy at the attribute provider PDP, then unexpected decision results may occur.
- *Proposed Solution.* Each attribute provider should predefine a conflict strategy to deal with this case. For example, a deny-overrides combining algorithm (see XACML) can be used to deny the attribute request if either policy denies it.

E.6. Cooperation between the user IdP and SPs: if the user IdP and the SPs cooperate with each other, they can create a master user account that could combine in addition to the user profile with his IdP, all the user local accounts with the different SPs with the same CoT.
- *Proposed Solution.* An audit trail party as suggested in previous points can monitor the user IdP-SPs cooperation and warn the user in case of any privacy-invasive transaction.

3.3.4 Discussion of Privacy Concerns from other Perspectives

While the focus in this thesis is on consumer privacy in identity federation architectures, there are privacy concerns from other perspectives as well. We can classify the different perspectives as following:

A. Consumer Privacy Perspective

B. Service-Provider Privacy Perspective

C. Government or Authorities Privacy Perspective

We went through a comprehensive privacy analysis from the consumer perspective (point A) in the Liberty Alliance identity federation architecture in section 3.3.3 where we identified and analyzed possible consumer-privacy breaches and concerns within the Liberty frameworks and we proposed several improvements. In this section, we briefly

discuss the importance of the other two perspectives and identify some privacy concerns from these different perspectives as examples.

Service-Provider Privacy Perspective

Most of the SP's privacy concerns in the Liberty identity federation architecture are in opposition to the power of the user IdP. Brands in [15] identified and discussed some of these concerns:

- The IdP can monitor the SPs' customers interactions in the CoT in real time.
- SPs are dependent on the on-line availability (to facilitate SSO mechanism) and honest collaboration of the IdP in order to interact with their own clients.
- The user IdP can impersonate its user and access his/her account in all related SPs in the same CoT.

In addition to that, the user IdP can be selective in dealing with the user SPs in the CoT. Thus, the IdP can easily deny access to the clients of a certain SP probably in certain times (for example, the IdP can ignore authentication, authorization, or federation requests from specific SPs) which may limit SP's clients access and consequently slow down the SP online business. The Liberty identity federation architecture requires trusted business relationships between the IdP and SPs in the CoT. Such trust is not necessarily always available among all the SPs in the CoT with the IdP in today's competitive business environment. This SP-privacy concern could be exacerbated if the SP is a member of more than one CoT and there is some kind of cooperation between two or more IdPs in these CoTs. The IdPs could monitor SP interaction whether with its customers or in certain interactions with other SPs (through some hosted Liberty services, e.g. Discovery service) in two or more CoTs, creating a valuable more comprehensive profile of the SP business activities. Furthermore, many SPs want to manage direct relationships with their customers without the burden of managing other relationships with third parties (IdPs) that may not be related to their businesses.

Another SP privacy concern is that the SP security and privacy interests actually depend on the security and privacy practices of the IdP itself (SP-clients IdP). As we discussed earlier, the Liberty architecture gives the IdP an invasive power in watching and facilitating most of the user transactions in the CoT. Therefore, if the IdP security

and privacy practices are not reliable, this will directly affect the security and privacy of the affiliated SPs in the CoT. For instance, if an IdP is compromised by an attacker, the attacker can commit identity theft across the entire CoT, having access to all users federated identity accounts. Consequently, this will cause massive identity theft damage that affects both the users and their SPs.

Similar SP privacy concerns exist with other parties hosting one or more Liberty services. For instance, the Discovery service, Interaction service, and People service are all very important components of the identity federation architecture. If one of these services is hosted by unreliable or untrustworthy provider, the SP privacy will be affected. All of these SP privacy concerns induce the SP to let its clients access its website using their local accounts instead of using their federated accounts. What is more likely is that the SP will not let the user federate his local account if the SP does not trust the security or privacy practices of the user IdP. As a result, with such power given to the IdPs (and other providers hosting Liberty services) given different business interests and competitions, we may end up with many cases in which the Liberty identity federation architecture is counteracted for the sake of SPs privacy.

Government or Authorities Privacy Perspective

Governments and regulatory authorities share most of the users and service provider privacy concerns. In fact, it could be part of the authorities' job to ensure user and service provider security and privacy in identity federation architectures. The Identity Project report from LSE in the UK [57] commented on privacy concerns from this perspective:

> "While enterprise identity architectures such as the Liberty Alliance ID-FF architecture may be adequate for the corporate management of the identities of employees and suppliers who access their corporate resources, it would have highly problematic implications if used for government-to-citizen identity management. The identity provider and the service providers would have the power to electronically monitor all citizens in real time across all government services, and its insiders (as well as hackers and viruses) would have the power to commit undetectable government-wide identity theft with a single press of a central button."

In other words, there is big privacy concern from the power given to the IdPs in the Liberty architecture as we discussed in detail in the pervious section (service-provider privacy perspective). These Governments and regulatory-authorities privacy concern would be exacerbated if we connect two or more circles of trust with each other since this connection usually requires linking them through a high-level IdP that would be a valuable party in both CoT. This could lead to a tree hierarchy where there are a few dominant IdPs. Furthermore, there could be different sorts of interaction and cooperation between these dominant parties leading to a worse situation.

The IdP power (and SP power in some cases) is of a particular sensitive concern to governments if some of the users' transactions and activities in the architecture include government-to-citizen data or identity management. For example, a voting system or a health insurance system in which Liberty identity federation architecture is deployed where there are few dominant IdPs (e.g. banks or well-known IT companies). In addition, the governmental SPs will not be able to function autonomously since the IdPs can monitor most of the interactions between governmental SPs and their clients.

We can diminish this privacy problem if the government assigns governmental IdPs in any CoT that include sensitive government-to-citizen data or transactions. However, the privacy and security practices of these IdPs will be very crucial since it will be very easy for a hacker compromising a governmental IdP or an IdP fraudulent insider to commit identity theft across the entire CoT. This could cause massive identity theft damage or denial of access for probably big portion of the government population.

Indeed, a comprehensive privacy analysis is required for privacy breaches/concerns from this perspective. An identity federation architecture that does not carefully address government privacy and confidentiality issues and reflect this in the design and the specification reduces the chance of adoption by governmental agencies. In fact, The Identity Project report from LSE concluded that Liberty Alliance identity federation architecture may be perfectly legitimate for a company to deal with the access rights of its own employees or small group of companies with coherence business relationships, but they found that the architecture has several unacceptable privacy implications when adopted for government-to-citizen interactions.

Chapter 4

Proposals for New Services in the Liberty Identity Federation Architecture

In this chapter, we propose three new services in the Liberty identity federation architecture to enhance consumer privacy. The Liberty privacy seal service is discussed in section 4.1. This service is similar to Web privacy seals but in an identity federation context. Thus, a relatively brief discussion is sufficient to describe. In section 4.2, the Liberty audit trail service is discussed. Likewise, several Web audit trail mechanisms exist and it is not necessary to describe this service in too much detail. We then focus on the Liberty user privacy preferences service which is new concept in the identity federation and not been proposed before. We discuss this service in detail in section 4.3.1. In section 4.3.2, we list our criteria for selecting a proper privacy preferences language for the new Liberty service. In this section, we also extend P3P terminology and we illustrate our choice of the XPref language.

4.1 Liberty Privacy Seal Service

As we noticed in our privacy analysis in chapter 3, it is always difficult to ensure that a service provider will adhere to its stated privacy policy and its declared purposes and other attribute usage directives. First, the user needs to know the privacy practices of the service provider (SP) to which the user identity provider (IdP) wants to introduce the user and the user needs to know to what extent this SP meets the identity federation privacy

requirements. Having this information will help the user to decide whether to be introduced to the SP or not. Moreover, this information will also help the user in deciding whether to federate his/her identity with the SP or not. Second, both the discovery service and the attribute provider (AP) will benefit from this information in deciding whether to reveal the user attributes (or the user attribute holder in case of the discovery service) to the SP requesting the user attribute.

If no technical mechanism exists, both the user and the AP will need to rely on their trust of the attribute requester. In this section, we propose a Liberty privacy seal service by a trusted third party that can certify and monitor identity and service providers' privacy policies, monitor privacy practices, and resolve any user disputes. This service should be hosted by a trusted third party that can be part of more than one CoT. Unlike the regular web privacy seal programs [11], the proposed seal program will assure the user and other providers that the SP privacy policies comply with the Liberty identity federation privacy requirements to some extent (please refer to section 3.2 for more information about the privacy requirements in identity federation). We suggest here four different levels of compliance with the Liberty identity federation privacy requirements:

1. "Very Compliant" Certificate: means the SP privacy policies, data collection, data usage processes and other identity federation privacy practices fully comply with the identity federation privacy requirements.
2. "Compliant" Certificate: means the SP privacy policies, data collection, data usage processes and other identity federation privacy practices comply with the identity federation privacy requirements with some minor exceptions. The exceptions are listed in the certificate itself.
3. "Some Compliance" Certificate: means the SP privacy policies, data collection, data usage processes and other identity federation privacy practices comply with the identity federation privacy requirements with some major exceptions. The exceptions are listed in the certificate itself.
4. "Not certified": the SP is not certified.

The privacy seal issuer is supposed to perform a thorough website audit and review of each domain to ensure these are consistent with the identity federation privacy

requirements before issuing the corresponding certificate. Moreover, after the SP is granted a privacy seal certificate, the issuer has to monitor the SP privacy practices and change the certificate to the corresponding level in case of any major change. The privacy seal issuer is also responsible to mediate disputes relating to the use of the user personally identifiable information within the identity federation architecture.

The user and some providers in the CoT will need to interact with this service in many cases. The following is a description of the most important cases:

A. Federation Introduction Consent

At the time of federation introduction, this service will assure the user to what extent an SP privacy policy and actual privacy practice match the Liberty identity federation privacy requirements. This will be via the user IdP that will first check the CoT Liberty privacy seal service to find out the SP privacy seal level. The user IdP will then notify the user of the possibility of federating his local identity with the SP (which is part of the affinity group) and will solicit permission from the user after giving the user the SP privacy seal level to facilitate the introduction process. Figure 4.1 gives the sequence diagram of the typical flow among the different parties.

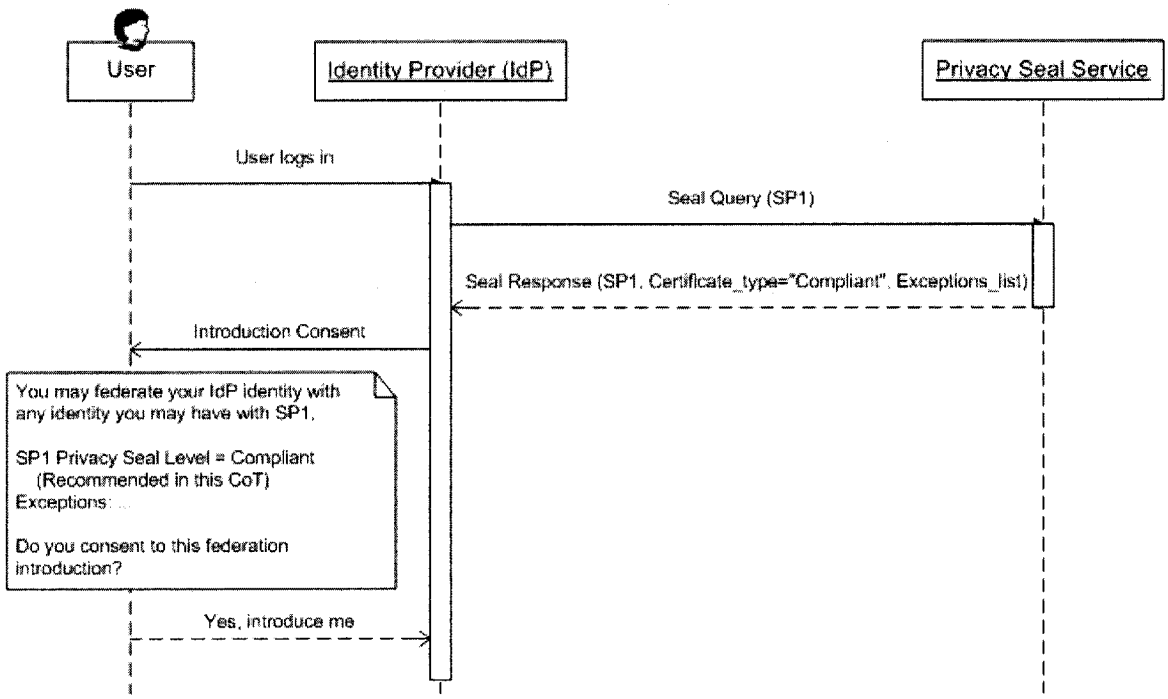

Figure 4.1: Liberty privacy seal service sequence diagram in the federation introduction

In fact, this service can also be used by the user to help in choosing the appropriate IdP to begin with.

B. Identity Federation Consent

The user will need to interact with the Liberty privacy seal service directly prior to the identity federation consent with the SP. So at a later time after the federation introduction, the user may visit SP1 that is a member of the affinity group. SP1 will recognize that the visitor is an Liberty-enabled user and will know which IdP in the CoT is the visitor's IdP. SP1 will offer to federate the user's local identity (in the SP1 website) with the user IdP identity. SP1 may also show the Liberty Privacy certificate to the user so the user will be more confident in giving consent. Before the user gives his consent, the user can verify that the seal is valid by checking with the Liberty privacy seal service. Figure 4.2 gives the sequence diagram of the typical flow in this case.

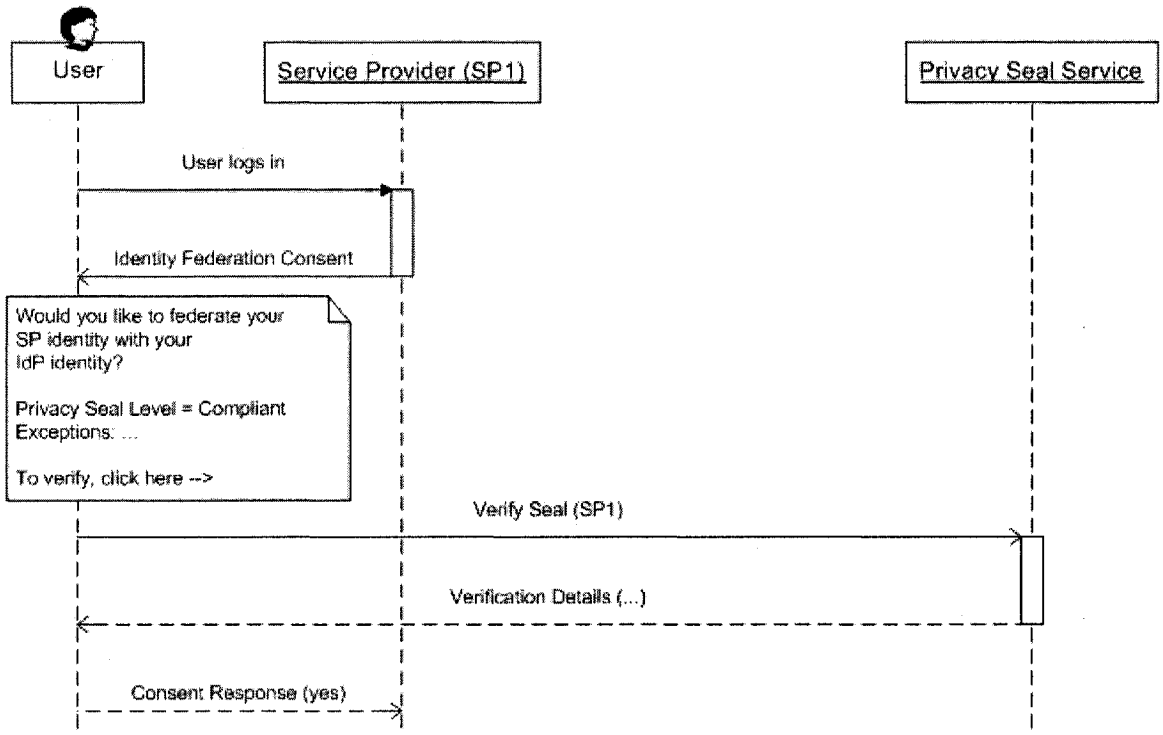

Figure 4.2: Liberty privacy seal service sequence diagram in the identity federation consent

C. Attribute Provider Request

This service can be consulted by the attribute provider before revealing any information. According to the attribute requester seal level, the attribute provider could permit or deny the access to the user attribute. Figure 4.3 gives the sequence diagram of the typical flow in this case.

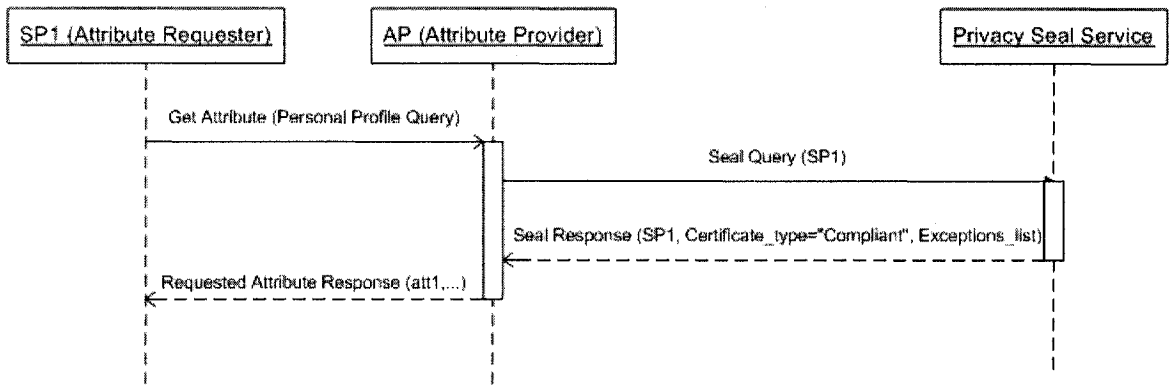

Figure 4.3: Liberty privacy seal service sequence diagram in the attribute provider request

Typically, a trusted third party is the best place to host the service. The privacy seal service can be part of the ID-WSF specification. Detailed message (requests/responses) specification is needed in which compatible interaction with ID-WSF protocols and services can be conducted. This service can help in increasing user trust in using Liberty-enabled services. It is important to note, however, that Web privacy seal organizations do not always revoke a seal certification from a business even after privacy violations have occurred (please refer to section 2.3.3.4 for more discussion about Web privacy seal). This indicates the need for strict rules (e.g., an automatic revocation mechanism) when deploying a Liberty seal service.

Taking the shortcomings of privacy seals into consideration, it is hard to make this new service a compulsory service in ID-WSF. Therefore, this service best fits in Liberty identity federation as an optional service that service providers and identity providers can use to increase users' trust in using their electronic businesses and Internet services. Furthermore, we will see in section 4.3 that the proposed user privacy preferences service can effectively substitute the seal services in some cases. To the best of our knowledge, there is no previous work related to a Liberty privacy seal and we believe there is still a need to further specify this new service in greater detail. However, this service is not the

main focus of our work in this thesis but it is definitely required to complement building a privacy-aware Liberty identity federation architecture. In fact, we saw in chapter 3 that several privacy breaches (section 3.3.3, A.4, A.5, E.1, and E.2) in Liberty architecture can be diminished when this Liberty privacy seal service becomes a part of Liberty identity federation architecture.

4.2 Liberty Audit Trail Service

Using the existing Liberty architecture, a user may have many privacy concerns about the different providers exchanging his personal information without his permission. In addition, many other transactions need to be recorded (e.g. transactions in steps 6, 9, and 10 in our use case scenario in section 3.3.2). The user needs to know if any privacy violation has been committed by any SP allowing him to take the appropriate action for future transactions and permitting him to update his privacy preferences accordingly.

An audit trail service as part of the Liberty identity federation architecture can achieve this task. The user (and probably his IdP) can access this service to review the transaction records. The provider hosting this service may need to notify the user in case of potentially dangerous violations. The user IdP is the appropriate provider to host this service since it is more trusted by the user. However, having service providers' privacy in mind and to limit the IdPs power, it is wiser that an independent trusted third party within the CoT host the service. This becomes a necessity if the audit trail service contains detailed information about the different activities between service providers and their customers or between service providers themselves.

For the audit trail service to be effective, identity and service providers need to be cooperative in supporting the audit trail service with the related logs of the different transactions. Table 4.1 gives a summary of the recorded information in the audit trail service from the different parties in the Liberty Alliance identity federation architecture. For each logged transaction, the following elements can also be recorded:

- Transaction Time
- Transaction Date
- CoT ID

- User Identity ID
- Provider Digital Signature
- Audit Trail Service Digital Signature

Provider/Service	Log of the following transactions
Discovery Service	• Revealing resource holder address to a requesting service provider • Updating information about user identity by adding or deleting an attribute provider (when the attribute provider registers its resource offering)
Interaction Service	• Obtaining permission to share the data with a Web services consumer (WSC) • Obtaining permission to federate/defederate with the service provider
Attribute Provider	• Sending user attribute to a service provider upon service provider request • Offering some user attributes by updating the discovery service
Service Provider	• Requesting user attribute or user attribute holder address from an attribute provider or a discovery service • Sharing user attribute directly with another provider either in the CoT or not
Identity Provider	• Introducing the user to other providers in the CoT • Federating/defederating user identity • User SSO login/logout

Table 4.1: A summary of the recorded information in the audit trail service

Peyton and Nozin in [67, 74] proposed Information Transfer Registry (ITR), as a mechanism to track compliance in a business to business network that is complementary to existing legislation and technical standards. The authors integrate the ITR as an audit trail component in the Liberty Alliance architecture to register all actions and to create a detailed report for the user. Moreover, the authors suggested a customer service gateway that enables the user to audit his personal information usage in one place by finding all the ITRs holding user information usage.

As in section 4.1, this service is not the main focus of our work in this thesis but it is definitely required to complement building a privacy-aware Liberty identity federation architecture. The existence of an audit trail service will help in diminishing some privacy breaches in Liberty architecture (section 3.3.3, A.4, B.3, B.5, D.1, E.1, and E.6). This service has to be a compulsory part of the Liberty ID-WSF in order to enhance consumer privacy in the Liberty identity federation frameworks. A smart audit trail service that can discover privacy violations in service providers' interactions and that is capable of presenting easy-to-understand reports and summaries to the user are among the major challenges. Furthermore, there should be an enforcement mechanism to gather the logs of the different transactions happening in the federation architecture (the transactions in table 4.1).

4.3 Liberty User Privacy Preferences Service

The main focus of this chapter is on the Liberty user privacy preferences service described in detail in section 4.3.1. In section 4.3.2, we explain our criteria for selecting a proper privacy preferences language for the new Liberty service and we also discuss our choice of the XPref language. In section 4.3.3, a brief summary of the key implementation issues is given.

4.3.1 Liberty user privacy preferences service: Motivation, Design, and Interaction with Other Services

One of the main objectives of identity federation is to enable simplified and fast user sign-on while browsing to different service providers. Thus, when the user visits an SP Web site, he/she will be automatically signed on and some of his information may be transferred from his IdP to the visited SP (upon a previous user permission) for better customization and service. Nevertheless, as we noted in chapter 3 scenario and privacy analysis, to enhance user privacy, we need to make the user aware of the excessive transactions occurring and to request his permission in many cases. Consequently, the user may be overwhelmed by many access permission requests and so identity federation will no longer be a fast, easy-to-use mechanism.

We propose a user privacy preferences service that can be part of the ID-WSF specification. The new service will enable the user to enter his default privacy preferences. As a design option, the user can have several preferences categorized by a generic classification of SPs according to different levels of privacy practices. The best place to host this service appears to be the user IdP. A Liberty-enabled user agent can host this service too; however, the user may then not be able to use his privacy preferences in case of using a different machine. The key advantage of the proposed service is that some access permission requests can be directed first to the Liberty user privacy preferences service to find out whether the user's default preferences allow the requested access (e.g., step 10.d in the use case scenario in section 3.3.2). The proposed Liberty user privacy preferences service will also help in diminishing some privacy breaches that we identified in our privacy analysis of Liberty identity federation architecture in section 3.3.3. In particular, the following privacy breaches/concerns are diminished: A.1, A.5, B.4, B.5, B.3, and E.4.

In our perspective, a suitable new service to the Liberty identity federation architecture should satisfy three important requirements. The first requirement is to enhance consumer privacy. The second is to fit compatibly in the current Liberty architecture. The third requirement is to increase user convenience or at least preserve the current user convenience of using the Liberty identity federation architecture. We will consider and meet all three requirements in our design of the proposed Liberty user privacy preferences service.

In our high-level design of this new service, the emphasis will be on the different cases and scenarios in which the Liberty user privacy preferences service can be utilized to enhance consumer privacy while preserving easy-to-use identity federation architecture. Thus, the design will not specify the actual Liberty message content for the different possible requests and responses for the functions offered by this service. Consequently, the new service interfaces for service metadata management, schema that accompanies the Liberty ID-WSF SOAP binding specification, and related SAML messages will not be specified. We believe that once we describe the conceptual model of this service and illustrate the functionality of the different operations offered by the service together with any accompanied processing rules, it would be best to leave

specifying the actual SOAP messages and other details to the Liberty Alliance team to agree on and define the detailed specification.

In the following, we will start our conceptual model by giving a detailed description of each function of the Liberty user privacy preferences service and a use case scenario in which the new service can be used for that particular purpose. In each use case scenario, we will go through the necessary operations and processing rules.

A. Expressing User Privacy Preferences

The first core role of the Liberty user privacy Preferences Service (PS) is to enable the user to specify her privacy preferences in the CoT. The provider hosting the PS will facilitate a mechanism through which the user can easily and clearly express her privacy preferences. This could be via an easy to use Web application where the user can login first and update her privacy preferences. There are two main requirements to efficiently enable this functionality. The first requirement is to use a fine-grained, machine-readable, dynamic, interoperable, and expressive privacy preference language. We will discuss this requirement and our choice of the privacy preference language in section 4.3.2. The second requirement is to provide clear, expressive, easy-to-use, and accurate mechanisms for the user to specify her privacy preferences. It is hard to find a privacy preference language that has the features mentioned in the first requirement and at the same time easy-use by the user. Therefore, we need a Web user interface that meets the second requirement and can be used by the user to specify her privacy preferences. Using this user interface, the Web application should be able to encode user privacy preferences into the privacy preferences language that we suggested for the first requirement. Designing the appropriate user interface that meets the second requirement is out of the scope of this thesis.

In diagram 4.4, we give a simple use case scenario in which the user interacts with the PS to set her privacy preferences. In this scenario, the user IdP is the provider hosting the PS in the CoT. The user needs to sign in first (using the SSO mechanism) in order to access the PS. The Web server in the PS will then give the user three options: add, delete or modify a privacy preference document. The user can create more than one privacy preference document in one or more PSs in the same CoT. Having more than one privacy preference document in the same PS enables the user to change the default preferences

document (it could be for example, to change to a more strict privacy preferences document or a more casual one). Having more than one PS for the same user in the same CoT enables the user to assign a different PS for each IdP, in case the user has more than one IdP in the same CoT. Moreover, it gives the user the flexibility to switch between her PSs in the same CoT (for example in case one PS is down).

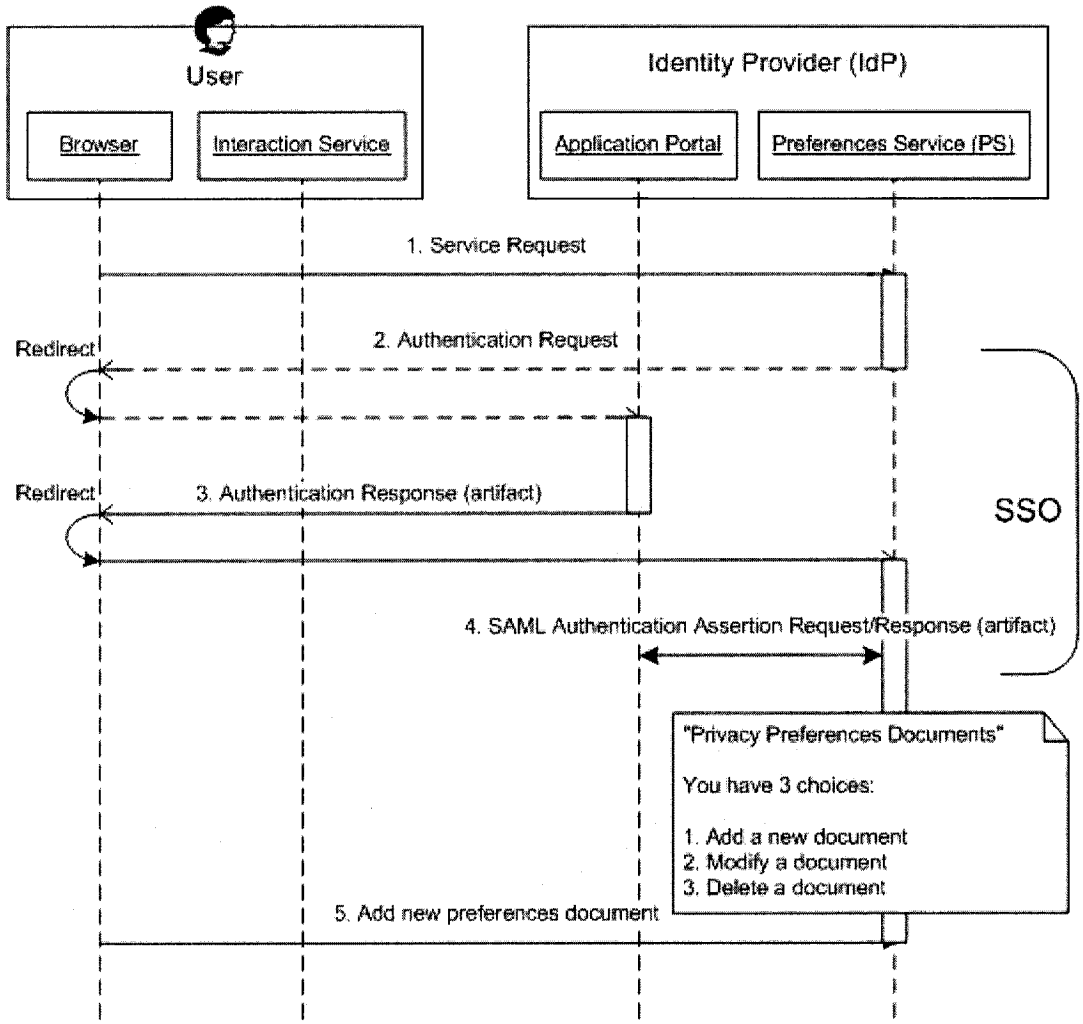

Figure 4.4: Setting user privacy preferences sequence diagram

There is actually no SOAP messages in this case scenario (other than authentication request/response messages for the SSO mechanism). The user in step 5 or any other operation to set her privacy preferences uses the PS Website directly. In case of adding a new preference document, the user will have the choice of starting the document either

from scratch or from predefined template. There should be several privacy preferences templates with different strictness to help the user start setting her new preferences document. However, it is always a privacy concern that the user may seldom change the template, especially if the user interface for setting the privacy preferences is not easy-to-use, and so these templates could be the default privacy preferences for many users in the architecture. This emphasizes again the importance of an easy-to-use web user interface that could encourage the user to change the template and give the user appropriate warnings and tips throughout the template changing process. After the user sets her privacy preferences document, the PS will convert the Web entries by the user to the corresponding machine-readable privacy preferences language (in the following section, 4.3.2, we will describe in detail our choice of the privacy preferences language which is XPref). For security and privacy reasons, the PS could send a signed copy of the user privacy preferences (in XPref format) to the user Interaction Service (IS) to sign it. The IS will sign and send the preferences document back to the PS. In fact, the last signature step can be one of the user's privacy preferences where the user is expected to turn this step on in case the PS host is untrustworthy.

B. Attribute Request at the Attribute Provider

The Liberty PS decides on behalf of the user when the Attribute Provider (AP) needs to release a specific user attribute to a specific SP in the CoT. The sequence diagram in figure 4.5 describes how the different parties collaborate with the preferences service. After the SP sends a Get Attribute request to the AP in step 1, the AP checks its privacy policy in steps 2 and 3 to find out whether the SP is authorized to access the requested attributes (by sending an authorization request to the hosted PDP service; AP is both a PEP and a PDP). Since the user never gave permission for the SP, there is no permit or deny policy result. Therefore, AP requires the user permission first. Instead of sending a request immediately to the user IS, the AP in this scenario will send a request to the user PS in step 4. The AP will first get the privacy policy address of the SP requesting the user attribute. The AP will send an access permission request to the user PS that includes the following:

- User attribute requested

- Usage directives (e.g. purpose, retention time, recipients …)
- SP identifiers (e.g. name or ID)
- SP privacy policy address

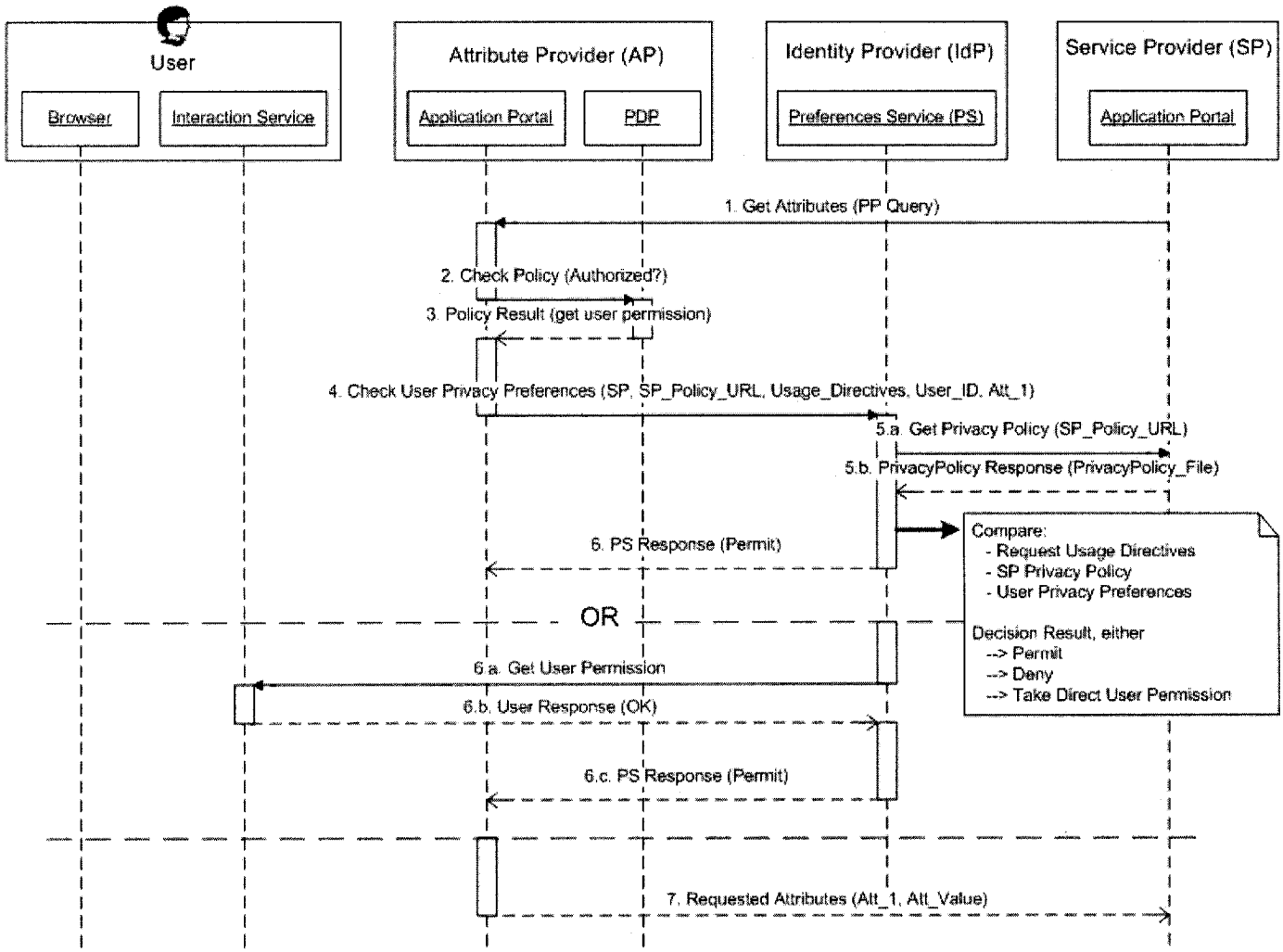

Figure 4.5: Attribute request at the attribute provider sequence diagram

The user PS will automate decision-making (whether to permit the SP to access the user attribute) based on comparing the corresponding user privacy preference set with the request message usage directives information. In case the attached usage directives (i.e. SP privacy practices) are not sufficient to make the comparison, the user PS may need to get the corresponding SP privacy policy (e.g. a P3P policy file). The decision output made by the PS is one of the following:

i. Permit: to let the AP release or send the user attribute to the requesting SP.

ii. Deny: to prevent the AP from releasing the user attribute to the requesting SP.

iii. Ask: to ask the user (via the interaction service) whether to release the user attribute to the requesting SP or not.

The PS will yield output (i) after the compliance check if the SP usage directives match the user privacy preferences (the usage directives are similar or stricter than user privacy preferences). Output (ii) will be as a result of the lack of compliance (discrepancies) with respect to user privacy preferences. Output (iii) indicates that the PS is not able to decide on behalf of the user given the information above and thus user assistance is needed. In this case, PS will ask for user permission by sending a request to the user interaction service (IS). The IS will get the user decision (when the user is online) and send a permit or deny response back to the PS.

As a result, the user PS in this scenario will either send a "Permit" as in step 6 or will need to get the user permission in case the privacy policy/privacy preferences matching designates a need to contact the user (as in steps 6.a and 6.b). If the user AP gets a "Permit" from the user PS (as in step 6 or 6.c), then the AP will send the requested attribute(s) to the SP as in step 7. Notice that the PS may need to load the SP privacy policy as in step 5 (it could be for example a P3P Web document) in order to compare it with the user privacy preferences.

C. Attribute Request from the User

The user privacy preferences service can substitute the need for a user-agent that requests and gets the SP website privacy policy file as in P3P (section 2.3.3.1). This user-agent compares the received policy file to the user's own set of privacy preferences (e.g. written in APPEL language) and then gives a warning in case of discrepancies (the user is asked if he wishes to consider other alternatives, consent to the exceptional practice, or to go elsewhere). The user-agent tasks can be achieved efficiently by the user PS in the Liberty identity federation architecture. This is shown in the interaction diagram in figure 4.6. As such, when the user visits the SP Website and before the user gives personal information to the SP, the SP will redirect (HTTP redirect) the user browser to the user PS as in the following steps:

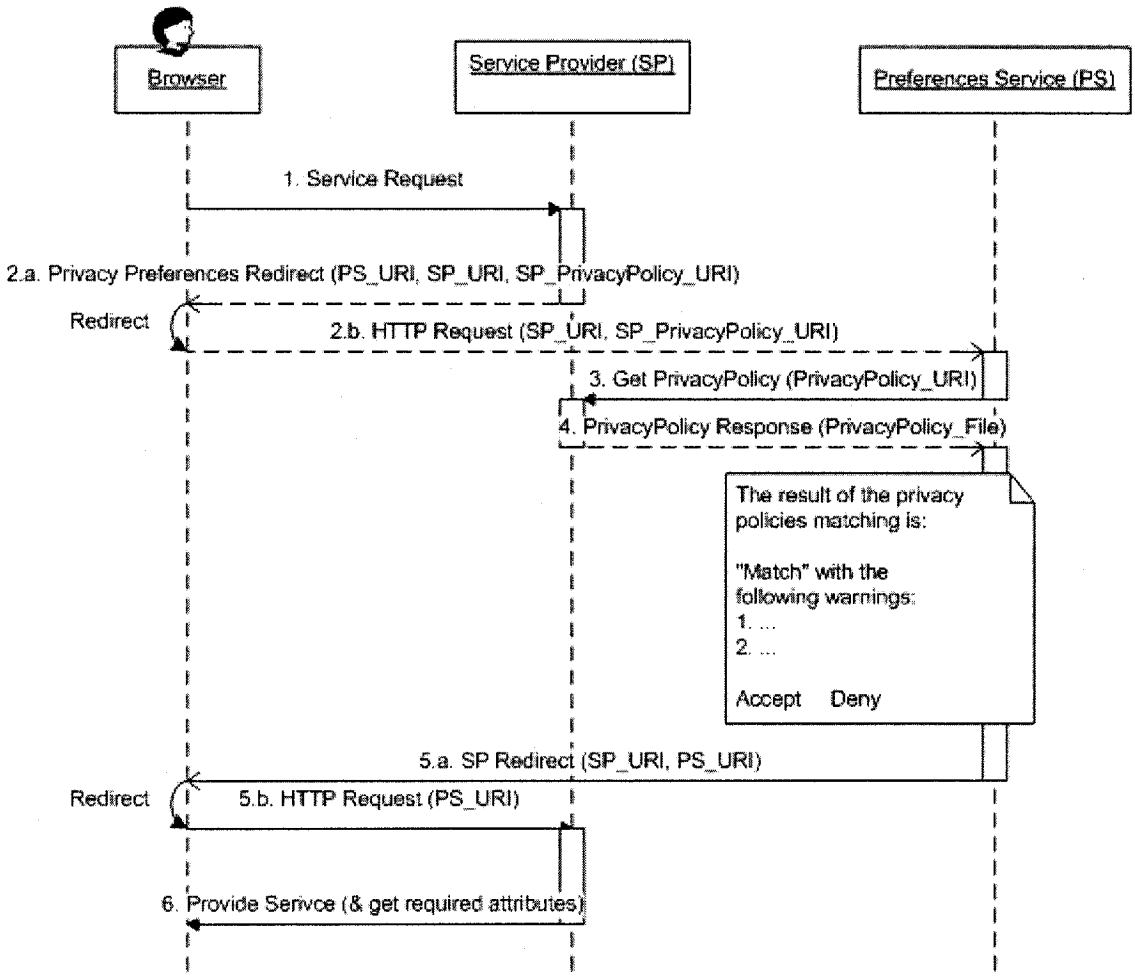

Figure 4.6 Direct user attribute request sequence diagram

1. The user browser sends an HTTP request to the service provider (typically a GET). In this step the user has typically clicked on a link in the Webpage presently displayed in the user browser.
2. (a) The service provider responds with an HTTP response with a status code of 302 (i.e. a redirect) and an alternate URI in the location header field. In this scenario, the Location URI will point to the user PS and will also contain two embedded URIs, one pointing to the SP privacy policy address and one pointing back to the SP.
2. (b) The user browser sends an HTTP request to the PS (typically a GET), specifying the complete URI taken from the location field of the response

returned in Step 2 as the argument of the GET. The URI contains also the two embedded URI.

3. The PS will send a SOAP privacy policy request message to the SP; the message contains the SP privacy policy address.
4. The SP sends a SOAP privacy policy response message containing the corresponding privacy policy file.
5. (a) The PS can then respond in kind with a redirect whose location header field contains the URI pointing to the service provider (extracted from the GET argument URI supplied in Step 2.b) and optionally contains an embedded, second URI pointing back to itself.
5. (b) The user browser sends an HTTP request to the service provider (typically a GET), specifying the complete URI taken from the Location field of the response returned in Step 5.a as the argument of the GET.
6. The SP provides the requested service which may require getting some personal information from the user.

D. Service Providers Introduction

The user privacy preferences service can also be used when the user IdP likes to introduce the user to other service providers in the affinity group. This use case is of particular value in case the Liberty seal service is not deployed in the identity federation architecture. Figure 4.7 depicts the interaction between the PS, IdP, and the SPs in the same CoT before the user gives his consent to be introduced.

In step 1, the user logs in first and requests a service. The user IdP will notify the user of the possibility of federating her local identity with an SP of the affinity group and will solicit permission to facilitate the introduction of the user to the member of the affinity group. However, the IdP needs first to find out if the SP privacy policy matches the user privacy preferences. In step 2, the IdP sends a "Check ServiceProvider PrivacyPolicy" query message to the user PS with the following parameters: user_ID, SP_ID, and SP_Policy_URL. In step 3, the user PS gets the SP privacy policy. Next, the PS compares user privacy preferences with the SP privacy policy and sends the result (whether SP privacy policy matches the user preferences and if there are any warnings to the user) back to the user IdP as in step 4. Finally, the IdP notifies the user of the possibility of

federating her local identity with the SP and shows to the user how the SP privacy policy matches the user privacy preferences. The user then either permits the federation introduction or denies it (as in step 5).

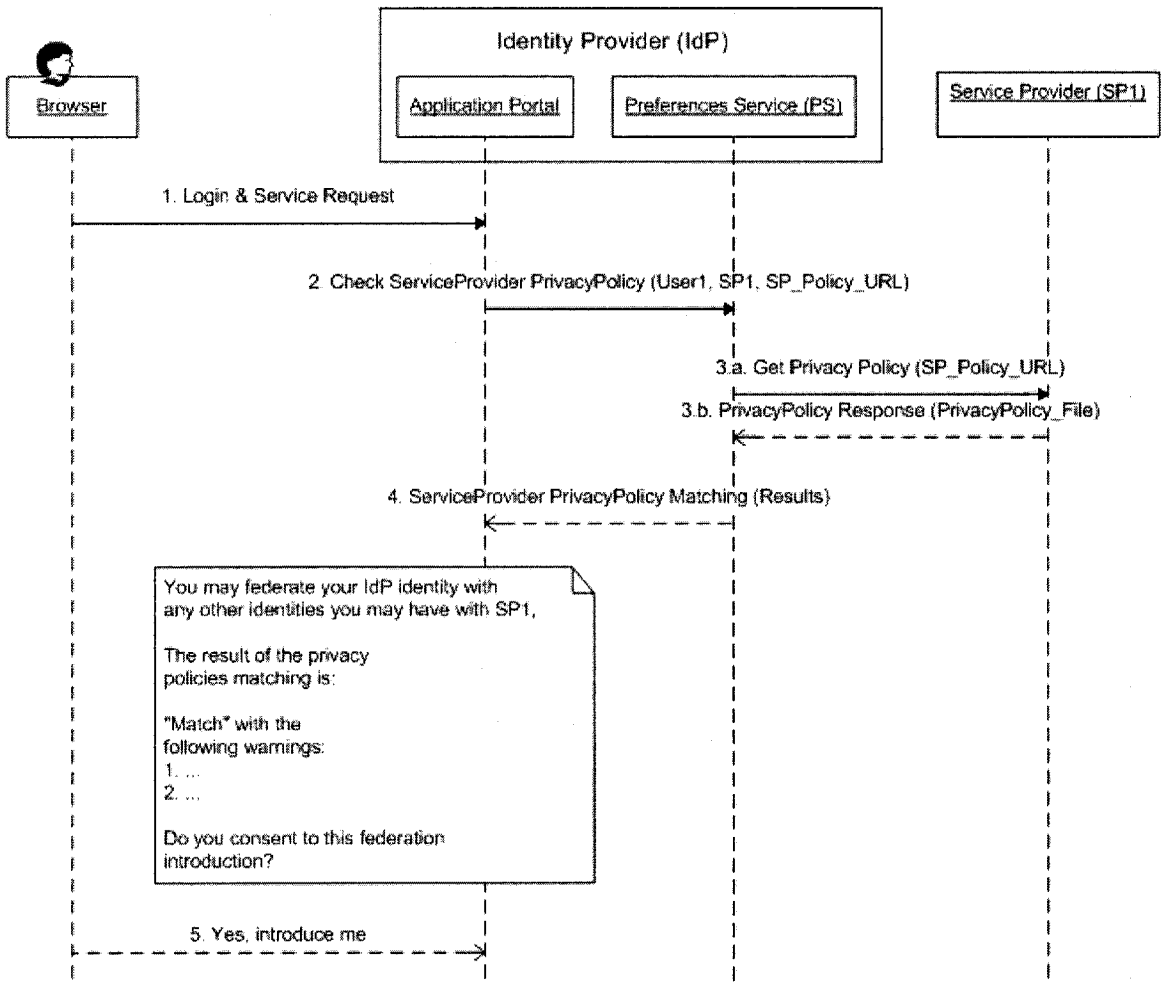

Figure 4.7 Service provider introduction sequence diagram

The idea of involving the user PS in the federation introduction process facilitates getting user permission to introduce more than one SP at the same time. This is due to the assurance of SPs' privacy policies matching offered first by the user PS in an automated mechanism. What's more, the user could specify in her privacy preferences that she is willing to be introduced to other members in the affinity group without a consent (sent by the user IdP) in case the members (SPs) privacy policies match the user privacy preferences.

Other Usages of the User Privacy Preference Service

There are a few other possible utilizations of the user privacy preferences service in the Liberty identity federation architecture. Similar to point B, the user PS can be used by the discovery service PDP to help decide whether to release the requested attributes holder address to the requesting SP. The discovery service will consult the user PS in case the PDP has no previous policy rule for releasing a specific user attribute to a specific SP. The PS will support the discovery service with the necessary information indicating if the SP privacy policy matches the user privacy preferences. Using this information, the discovery service PDP can decide either to release the attribute holder address or not. Furthermore, it adds the corresponding policy rule for further queries.

Another usage of the user privacy preferences service is in choosing the appropriate IdP. Thus, the user checks how the IdP privacy policies matches her privacy preferences and upon that and taking into consideration other factors, the user decides the suitable IdP.

Challenges of the User Privacy Preference Service

Throughout our discussion of the different usage scenarios of the user PS, one may wonder about the following: instead of matching SP privacy policy against the user preferences, the AP can more easily verify whether or not its policies subsume the SP privacy policy. Subsumption reasoning is used on policies defined over equal or similar class of data in order to determine if they conflict. Therefore, in case the AP policy subsumes the SP policy, this means the AP can send the requested user attribute to the SP and so there is no need to interact with the user PS. On the other hand, if the AP policy does not subsume the SP policy, then the AP needs to interact with the user interaction service to get immediate user consent. However, this subsumption mechanism is not quite ideal for two main reasons:

1. It implies that the user trusts the AP (which is actually a SP in the CoT) in making a proper fair subsumption of the privacy policies.
2. It implies that the user has no control in changing the way her personal information will be shared between service providers after she releases this information. In other words, the user can not change her privacy policy of

certain data after she releases this data and so the service providers can share this data using only the original privacy policy.

Using our PS, the user can change her privacy preferences to a stricter or relaxed one after releasing some of her information to a SP (which becomes an AP). At a later time when this AP needs to share this information with another SP and since the AP needs to consult the user PS first, the releasing of this information will depend on the strictness of the new user privacy preferences. Such a characteristic is hard with the subsumption mechanism.

This new service will raise some new privacy concerns. The following are the main concerns:

- Service providers should not know the user's privacy preferences unless required.
- The preferences service specification should oblige referring to this new service in several usage scenarios, in particular, the AP has to interact with the PS whenever there is a new attribute request that is not covered by the AP access policy.
- The AP could release the user attributes even if the PS shows no match between the SP privacy policy and the user privacy preferences; an audit trail of PS requests and results could help in limiting this problem.
- The user should be informed of the PS results in some cases; failing to do so might makes the user give her consent relying on incomplete information.
- An ambiguous user interface for editing privacy preferences may lead to improper usage or sharing of the user personal information.
- The PS should be hosted by a trusted party in the CoT; hosting this service by the user IdP will raise the power of the IdP in the Liberty architecture (one may refer to section 3.3.4).

It is also important to define the threat model for this new service. In fact, this service shares many characteristics of other existing services in the Liberty identity federation architecture. In particular, the way the different messages are sent to/from the user privacy preferences service. In [91], the author addressed privacy and security fair

information practices, and implementation guidance for organizations using the Liberty Alliance specifications. In particular, [31, 52] provided an overview of the security and privacy issues in ID-WSF technology and briefly explained potential security and privacy ramifications of the technology used in ID-WSF. The threat model discussed in these documents can be used here together with the different suggestions and recommendations to improve the security of our proposal.

4.3.2 Criteria for Selecting an Appropriate Privacy Preferences Language for the Liberty User Privacy Preferences Service

In this section we will go through our criteria for selecting a proper privacy preferences language for the new Liberty service and we will also discuss our choice of the XPref language. To do so, we need first to create a typical user privacy preferences document in the Liberty identity federation architecture. Thus, we need to specify what attributes, transactions, or objects the user needs to determine her privacy preferences about. We will also extend P3P to accommodate necessary identity federation terminology.

Typical User Privacy Preferences Document in Identity Federation

Generally speaking, Web privacy policies include multiples of the following elements:

- **Data:** a specific user data item that will be collected
- **Purpose:** describes purposes for which the user data item will be collected.
- **Recipient:** describes the intended users (recipients) of the data item that will be collected.
- **Retention:** defines the duration for which the data item will be kept.

The user privacy preferences are usually a set of similar rules stating for every collected data item what possible recipient, purpose, and retention time the user prefers the collector (service provider) privacy policy to include. Accordingly, the user can specify an action to be taken for each rule when the rule fires. The action can be simply one of three options:

- "accept" the service provider privacy policy
- "deny" the service provider privacy policy
- "ask" the user for permission after supporting the user with sufficient warnings.

In fact, the comparison between the service provider privacy policy and the user privacy preferences can be achieved by matching each rule from the user privacy preferences against the provider privacy policy set of rules. Once a match triggers with a specific action, all other preferences rules with the same action are evaluated in order so that any corresponding warning is shown to the user.

The only standard that enables service providers to express their privacy practices in a standard format that can be retrieved automatically and interpreted easily by user agents is P3P (described in section 2.3.3.1). Therefore, we will adopt using P3P to express service providers' and identity providers' privacy policies. However, the Liberty identity federation architecture has special privacy requirements that cannot be satisfied using P3P as it is. P3P does not accurately address "notice", "choice", relevance", and "timeliness" that we discussed in section 3.2. For example, the language needs to be more specific about the recipients of the data (e.g. within the CoT or outside CoT), the purpose for which user attributes were collected (e.g. for federation purpose), or how long these attributes will be retained (e.g. retained for the user federation period). In this section, we propose an extension to P3P to accommodate identity federation privacy requirements. This extension consists mainly of adding new elements to the P3P terminology.

In this sense, we propose a service provider privacy policy in the Liberty identity federation architecture that consists of two parts:

A. A typical Web privacy policy: enables the service provider to specify what personal information will be collected about the user under what conditions and for what purposes to which recipients and for how long. Typically speaking, this is a regular Web privacy policy (P3P policy) that a Web site might publish in a machine-readable format so that it can be automatically fetched, evaluated and viewed by user-agent.

B. Liberty identity federation privacy policy: enables the Liberty-enabled provider to specify what identity federation decision can be taken about the user under what conditions, for what purposes, to which recipients, and for how long.

Each provider is expected to have one or more privacy policies for each circle of trust (CoT). Likewise, we propose a user privacy preferences in the Liberty identity federation architecture that consists of two parts:

A. Typical Web privacy preferences: enables the user to specify what personal information can be collected about her under what conditions and for what purposes to which recipients and for how long. This is a regular Web privacy preferences that a Web user might write so that before visiting any Web site, the user agent (e.g. the browser) can compare these prewritten user preferences with the visited Web site privacy policy.
B. Liberty identity federation privacy preferences: enables the user to specify what identity federation decision can be taken about her by the Liberty provider, under what conditions, for what purposes, to which recipients, and for how long.

The user could have one or more privacy preference for each CoT. Our focus in this section is the point B (either from the service provider policy or the user preferences) since it has some specific characteristics to facilitate expressing identity federation privacy preferences. In our work, we adopt the terminology of the P3P standard. Under this part, the user can express the Liberty identity federation privacy preferences by using some of the statements in table 4.2 in a P3P Preferences Exchange Language (APPEL) format. This table provides some important identity federation privacy preferences to illustrate the concept of point B and is not intended to be complete. Any element in the table that starts with # means it is a new one that we add to the list of possible elements in the corresponding field in the P3P 1.1 specification [94]. In the following tables we explain each new element that we suggest adding to the P3P standard.

It is possible to use a different format to express both the Liberty identity federation provider policy and the user preferences. For example, they could be expressed in a simple XML format. However, using P3P terminology has the advantages of using a standard Web privacy policy format and it is consistent with part A. In fact, the data items in the statements in table 4.3 are not actual data items that the Liberty provider may collect about the user. Rather, they are user options pertaining to identity federation that could affect the user privacy. The Liberty providers will publish similar privacy policies

so that the user PS can evaluate the provider identity federation privacy policy against user identity federation privacy preferences. This evaluation might result in some warnings to the user that could be sent either directly from the user PS or via another provider (e.g. the IdP). The warnings vary according to the different identity federation transactions so that the PS advices the user and gives her different actions to choose.

#	Rule	Statement
1	Federation Introduction	Data-Item: #"federation-introduction" Purpose: #"IdP-introduction" | "other-purpose" Recipient:"same" | "other-recipient" | "unrelated" Retention: "stated-purpose" | "indefinitely"
2	Identity Federation Consent	Data-Item: #"federation-consent" Purpose: #"federation" | "other-purpose" Recipient: "ours" | "unrelated" Retention: "stated-purpose" | "indefinitely"
3	Identity Federation Consent Signature	Data-Item: #"user-signature" Purpose: #"federation" | "other-purpose" Recipient: "ours" | "unrelated" Retention: "stated-purpose" | "indefinitely"
4	SSO	Data-Item: #"login-information" Purpose: "current" | "other-purpose" Recipient: "ours" | "same" | "other-recipient" | "unrelated" Retention: "stated-purpose" | "indefinitely"
5	SSO Redirection	Data-Item: #"web-redirection" Purpose: #"http" | #"post" Recipient: "ours" | "unrelated" Retention: "stated-purpose" | "indefinitely"
6	User IdP Discovery	Data-Item: #"common-domain-cookie" Purpose: #"IdP-discover" | "other-purpose" Recipient: "ours" | "same" | "other-recipient" | "unrelated" Retention: "stated-purpose" | "indefinitely"
7	Discovery Service Location	Data-Item: #"discovery-service" Purpose: #"hosting" | "other-purpose" Recipient: "ours" | "same" | "other-recipient" | "unrelated" Retention: "stated-purpose" | "indefinitely"

Table 4.2: Some Liberty identity federation privacy preferences options

For the Liberty identity federation privacy preferences, the user can make a condition that the service provider has a Liberty privacy seal before dealing with the service provider Web site. Assuming there is a Liberty seal organization (www.liberty-privacy-seal.com), the following example shows how to specify this condition using APPEL:

```
<appel:RULE behavior="limited" prompt="yes"
              description="the visited service provider has no Liberty
              privacy seal ">
              promptmsg="Warning! This Service Provider has no Liberty
              privacy seal, Do you want to continue (using limited
              access)?">
    <p3p:POLICY>
      <p3p:DISPUTES-GROUP appel:connective="non-or">
        <p3p:DISPUTES resolution-type="independent"
                    service="http://www.liberty-privacy-seal.org/*"/>
      </p3p:DISPUTES-GROUP>
    </p3p:POLICY>
</appel:RULE>
```

New Elements to P3P

Our extension to P3P (to accommodate identity federation privacy requirements) consists of adding new elements to the P3P terminology. For the data collection, there are currently 17 predefined data categories in P3P. We add the new data category called "<identity-federation>" which all the new data elements are under. The new data elements that we suggest are explained in table 4.3. It is important to notice that we can add further new data elements under this new category (i.e. "identity-federation" category). All data items should be classified into one of 12 predefined purpose classes in P3P. We suggest adding several more classes that we describe in table 4.4. Furthermore, there are sex options to describe who has access to the data. We propose adding 5 more options as in table 4.5. There are five options to specify how long the data is being retained. In table 4.6, we define two more options. Finally, P3P supports six different access methods which the user can use to access her personal information (e.g. to change or delete the data if it is incorrect or illegally stored). In table 4.7, we add two more methods to serve the identity federation user data access.

Please refer to P3P for the definitions of existing classes : data elements, purposes, access, and retention time [94].

Element	Description
"federation-introduction"	The federation introduction held by the user IdP to introduce the user to other members in the affinity group. The recipient is the most important part of the statement about this data item
"federation-consent"	The existence of this data element in the service provider privacy policy means the service provider will get user consent before federating user identity
"user-signature"	The existence of this data element in the service provider privacy policy means the service provider will enable the user to sign the federation consent during federating user identity
"login-information"	This element represents the authentication information that the user IdP sends to the service provider; for example the authentication method and other related detail (e.g., password length) to help the service provider to evaluate the authentication mechanism
"web-redirection"	This represents the Web method for redirecting the user to her IdP for SSO (it could be either Post or HTTP redirect)
"common-domain-cookie"	The existence of this data element in the identity provider privacy policy means the identity provider will use a common domain cookie (to enable SPs to discover the IdP)
"discovery-service"	The existence of this data element in the identity provider privacy policy means the identity provider will host the user discovery service as well

Table 4.3: New data elements to P3P under "identity-federation" category

P3P Purpose Classes	New Purpose Classes
<current/> <admin/> <develop/> <tailoring/> <pseudo-analysis/> <pseudo-decision/> <individual-analysis/> <individual-decision/> <contact/> <historical/> <telemarketing/> <other-purpose>	<IdP-introduction/>: Information may be collected for the purpose of enabling user introduction to the member of the affinity group
	<federation/>: Information may be used for the purpose of enabling user identity federation
	<IdP-discover/>: Information may be used for the common-domain-cookie for the purpose of discovering the user IdP
	<hosting/>: Information may be used for the purpose of hosting the service
	<SSO/>: Information may be used to enable SSO mechanism
	<http/>: HTTP web redirection
	<post/>: POST web redirection

Table 4.4: New purpose classes in P3P

P3P Recipient Classes	New Recipient Classes
<ours> <delivery> <same> <other-recipient> <unrelated> <public>	<SP-same-CoT>: Legal service providers within the same CoT
	<SP-other-CoT>: Legal service providers in other CoT
	<IdP-same-CoT>: Legal identity providers within the same CoT
	<IdP-other-CoT>: Legal identity providers in other CoT
	<DS>: Legal discovery service in the same CoT

Table 4.5: New recipient classes in P3P

P3P Retention Options	New Retention Options
<no-retention/> <stated-purpose/> <legal-requirement/> <business-practices/> <indefinitely/>	<sso>: Information is retained for the period from user single sign on until the user single logout
	<federation>: Information is retained for the period from user federation until user defederation

Table 4.6: New retention options in P3P

P3P User Access Methods	New User Access Methods
<nonident/> <all/> <contact-and-other/> <ident-contact/> <other-ident/> <none/>	<federation-and-other>: Identity federation information and other identified data: access is given to any identity federation information as well as to certain other identified data.
	<federation-and-all>: Identity federation information and all identified data: access is given to any identity federation information as well as to all identified data.

Table 4.7: New user access methods in P3P

XPref versus APPEL

As we indicated in section 2.3.3.2, the authors in [5] showed the limited expressiveness of APPEL. APPEL can specify only what is unacceptable and not what is acceptable for a user. In some cases, the APPEL constructs interact with the P3P policy language in unintended ways, making it non-trivial to get even simple preferences defined correctly. So it is easy to write a preference that appears correct and find that it does not accomplish the intended goal. Moreover, writing APPEL preferences is error prone and some simple preferences are difficult to express. The proposed solution, XPref in [5], is an XPath based language for user preferences that solves some of the APPEL problems. XPref uses

a strict subset of XPath 1.0 [25] for all of its functionality except the use of the quantified expression "every" from the XPath 2.0 Working Draft.

We summarize the shortcomings of APPEL in expressing user privacy preferences in the following:

1. APPEL specification does not have logical operators for combining multiple rules in a ruleset, so they are evaluated strictly in order. Therefore, the first shortcoming in APPEL is that there is no logical connective for expressing preferences that specifies directly what is acceptable in the provider privacy policy, only what is unacceptable. Thus the user has to convert his specification of what is acceptable into what is unacceptable in order in express it in APPEL. This makes the current APPEL specification convoluted, verbose, and confusing for the user in several cases.
2. The second shortcoming is that an APPEL rule can be viewed as a tree rooted at the RULE node, so APPEL connectives can only be placed at the nodes. The children of a node can only be combined using the logical connective specified at the node which makes it impossible to combine them in any other logical way. Thus we have to break them in multiple rules which makes the preferences complex to write and understand.

For each shortcoming, we will go through an example in which the user will try to specify her identity federation privacy preferences in the Liberty user privacy preferences service using APPEL. For each example, we will show that it is difficult for the user to express her privacy preferences in APPEL and we will show that the corresponding XPref privacy preferences is easy to write and understand.

Example 1

In this example, the user wants to write the following privacy preferences (in the Liberty user privacy preferences service) in a certain identity federation circle of trust that she is a member of:

Privacy preferences I:

"Only identity providers or service providers in the same circle of trust (CoT) can be the recipients (in the provider privacy policy) of her personal information"

So the user will block the Web site of any provider that will exchange her personal information with another identity provider or service provider outside the CoT. Therefore, the only acceptable recipients in the user privacy preferences are:

- "ours" or
- "SP-same-CoT" or
- "IdP-same-CoT"

When the user tries to write this simple privacy preference in APPEL language, she may end up writing the following:

```
<appel:RULESET>
   <appel:RULE behavior="request">
      <POLICY>
         <STATEMENT>
            <RECIPIENT appel:connective="or-exact">
               <ours/> <IdP-same-CoT/> <SP-same-CoT/>
            </RECIPIENT>
         </STATEMENT>
      </POLICY>
   </appel:RULE>
<appel:RULE behavior="block"/>
   <appel:OTHERWISE/>
</appel:RULE>
</appel:RULESET>
```

The semantics of "or-exact" is that if a statement (in the provider privacy policy) lists "Recipient" other than "ours", "SP-same-CoT" or "IdP-same-CoT", the rule will not match. However, this is will not achieve the user preference target since the policy language allows a policy to contain multiple statements and the rule fires if any of the statements satisfies the rule. So, if for example, the provider privacy policy contains two statements, where one only lists "ours", "SP-same-CoT" or "IdP-same-CoT" as the recipients, but the second statement includes "SP-other-CoT", the above APPEL preference will match the provider privacy policy because of the first statement although

the second statement clearly states this Web site will share the user personal information with providers outside the CoT as in the second statement. Even if we replace the "or-exact" connective with "and-exact", this will not solve the problem. This is because the policy element in a P3P policy usually contains other subelements such as entity and access, apart from statement. Therefore, placing an "and-exact" operator in the policy will cause the rule to fail for many acceptable privacy policies whose policy element included other subelements, even if all the statements conformed to the rule.

The only way to write this preference in APPEL is to switch it to what is unacceptable. So we convert the following:

∀ recipient (recipient = "ours" ∨ recipient = "SP-same-CoT" ∨ recipient = "IdP-same-CoT") → "request"

to:

∃ recipient (recipient ≠ "ours" ∧ recipient ≠ "SP-same-CoT" ∧ recipient ≠ "IdP-same-CoT") → "block"

which becomes the following in APPEL:

```
<appel:RULESET>
   <appel:RULE behavior="block">
      <POLICY>
         <STATEMENT>
            <RECIPIENT appel:connective="or">
               <SP-other-CoT/> <IdP-other-CoT/>
               <delivery/> <same/>
               <other-recipient/> <unrelated/>
               <public/>
            </RECIPIENT>
         </STATEMENT>
      </POLICY>
   </appel:RULE>
<appel:RULE behavior="request"/>
   <appel:OTHERWISE/>
</appel:RULE>
</appel:RULESET>
```

This new preference is more difficult to understand and makes APPEL complex and error prone. Using XPref, we can write the same preference in a very easy and straightforward way:

```
<RULESET>
   <RULE behavior="request"
      condition="/POLICY [
         every $pname in
         STATEMENT/RECIPIENT/* satisfies
            (name($pname) = "ours" or
            name($pname)  = "IdP-same-CoT"
            name($pname)  = "SP-same-CoT"  ) ]" />
   <RULE behavior="block" condition="true"/>
</RULESET>
```

In XPref, a rule fires if the XPath expression returns a non-empty result. The two outermost XML elements in APPEL (RULESET and RULE) are retained. The only difference is that instead of using the subelements of RULE to specify acceptable or unacceptable combinations of P3P elements, XPref uses an XPath expression that specified in a new condition attribute of the rule. Notice that we here wrote what is acceptable directly as the preference indicates which makes XPref easier to write and reduces possible user mistakes.

Example 2

In this example, the user wants to moderates his privacy preferences so it is less strict than the preference-I (in example 1) as follows:

Privacy preferences II:

"Only identity providers or service providers in the same circle of trust (CoT) can be the recipients (in the provider privacy policy) of her personal information; if the recipient is a service provider outside the CoT then it must be only for the purpose of pseudonymous analysis"

So the user will block the Web site of any provider that will exchange her personal information with other identity provider or service provider outside the CoT, except if the purpose of collecting the user information is "pseudo-analysis". In this case, the provider

can exchange this information with other providers outside the CoT. Therefore, the only acceptable recipients in the user privacy preferences are:

- "ours"
- "SP-same-CoT"
- "IdP-same-CoT"

or

- "SP-other-CoT"

if the purpose is "pseudo-analysis".

Since we learned from example one the need to convert the "acceptable" rules to the "unacceptable" rules, the user will try the converting the following:

∀ recipient, purpose (
(recipient = "ours" ∨ recipient = "SP-same-CoT" ∨ recipient = "IdP-same-CoT")
∨
(recipient = "SP-other-CoT" ∧ purpose = "pseudo-analysis"))
→ "request"

to:

∃ recipient, purpose (
(recipient ≠ "ours" ∧ recipient ≠ "SP-same-CoT" ∧ recipient ≠ "IdP-same-CoT")
∧
(recipient ≠ "SP-other-CoT" ∨ purpose ≠ "pseudo-analysis"))
→ "block"

However, according to the second shortcoming of APPEL, we can only specify those logical expressions whose parse trees are subtrees of the XML tree corresponding to the P3P schema. Therefore, this expression cannot be directly expressed in APPEL and we need to divide the expression into two rules as follows:

1. ∃ recipient (recipient ≠ "ours" ∧ recipient ≠ "SP-same-CoT" ∧ recipient ≠ "IdP-same-CoT" ∧ recipient ≠ "SP-other-CoT") → "block"
2. ∃ recipient, purpose (recipient = "SP-other-CoT" ∧ purpose ≠ "pseudo-analysis") → "block"

So the APPEL preferences will be as follows:

```
<appel:RULESET>
   <appel:RULE behavior="block">
      <POLICY>
         <STATEMENT>
            <RECIPIENT appel:connective="or">
               <IdP-other-CoT/> <delivery/>
               <same/> <other-recipient/>
               <unrelated/> <public/>
            </RECIPIENT>
         </STATEMENT>
      </POLICY>
   </appel:RULE>
   <appel:RULE behavior="block">
      <POLICY>
         <STATEMENT>
            <RECIPIENT appel:connective="or">
               <SP-other-CoT/>
            </RECIPIENT>
            <PURPOSE appel:connective="or">
               <current/> <admin/>
               <develop/><tailoring/>
               <pseudo-decision/> <individual-decision/>
               <individual-analysis/> <contact/>
               <historical/> <telemarketing/>
               <other-purpose/> <IdP-introduction/>
               <federation/> <IdP-discover/>
               <SSO/> <extension/>
            </PURPOSE>
         </STATEMENT>
      </POLICY>
   </appel:RULE>

   <appel:RULE behavior="request"/>
      <appel:OTHERWISE/>
   </appel:RULE>
</appel:RULESET>
```

Clearly, this APPEL preference is hard to construct and understand. In fact it is tricky to write this preference to begin with. This preference can be easily written using XPref as follows:

```
<RULESET>
   <RULE behavior="request" condition="/POLICY [
         every $stmt in STATEMENT satisfies (
            every $recipient in $stmt/RECIPIENT/*,
            every $purpose   in $stmt/PURPOSE/* satisfies (
               name($recipient) = " ours" or
               name($recipient) = " SP-same-CoT"
               name($recipient) = " IdP-same-CoT" or (
                  name($recipient) = "SP-other-CoT" and
                  name($purpose) = "pseudo-analysis")      )
               ]" />
<RULE behavior="block" condition="true"/>
</RULESET>
```

Here, the user specified what is acceptable the way the textual preference indicates. It is also straightforward and easy to understand by the user as well.

As we noticed from the previous examples, XPref is more appropriate as a user privacy preferences language in our proposed Liberty user privacy preferences service. Using XPref, we can easily specify preferences that are hard to specify using APPEL. This will be easier whether it is the user who will directly specify the privacy preferences in XPref or it is the Web application that will convert the user preferences from a dedicated Web user interface.

Chapter 5

Discussion of Proposed Services

In this chapter, we demonstrate the benefits and the effectiveness of the proposed services by presenting two use case scenarios before and after applying our proposals in section 5.1. The first use case scenario is the same one we discussed in chapter 3. The second scenario is a new one to show some other aspects that we have solved.

In section 5.2, we discuss how our proposed services would conform and integrate with Liberty identity federation architecture. We also show that Liberty retains all the advantages the architecture already has after integrating with the proposed services. In addition, we describe the applicability of the proposed services and solutions to other identity federation architectures in section 5.3. In particular, we consider Shibboleth and Microsoft InfoCard architectures.

5.1 Use Case Scenarios

In this section, we show how our new services support key scenarios that the current Liberty identity federation architecture does not. The first use case scenario is already discussed in section 3.3.2 and we show here the same scenario but after having the Liberty user privacy preferences service in place. The second use case scenario is a new one in which we show some other aspects that we have solved. For each scenario, the interaction between Liberty services and the user involvement is evaluated before and after injecting our proposed Liberty services.

5.1.1 Use Case Scenario I

In section 3.3.2, we have discussed this use case scenario in detail. This scenario shows the typical message flow between the different parties in Liberty Frameworks. The scenario integrates the usage of both the Liberty ID-FF and ID-WSF with some functionalities of Liberty ID-SIS. Though, the focus of this scenario is on the ID-WSF framework.

Figure 5.1 shows the cooperation and interactions between different Liberty components after having our new services in place. Once again, the focus in this chapter is on the Liberty user privacy preferences service. In this diagram, we disregard the steps from 1 to 8 in figure 3.4 since they would not have major changes.

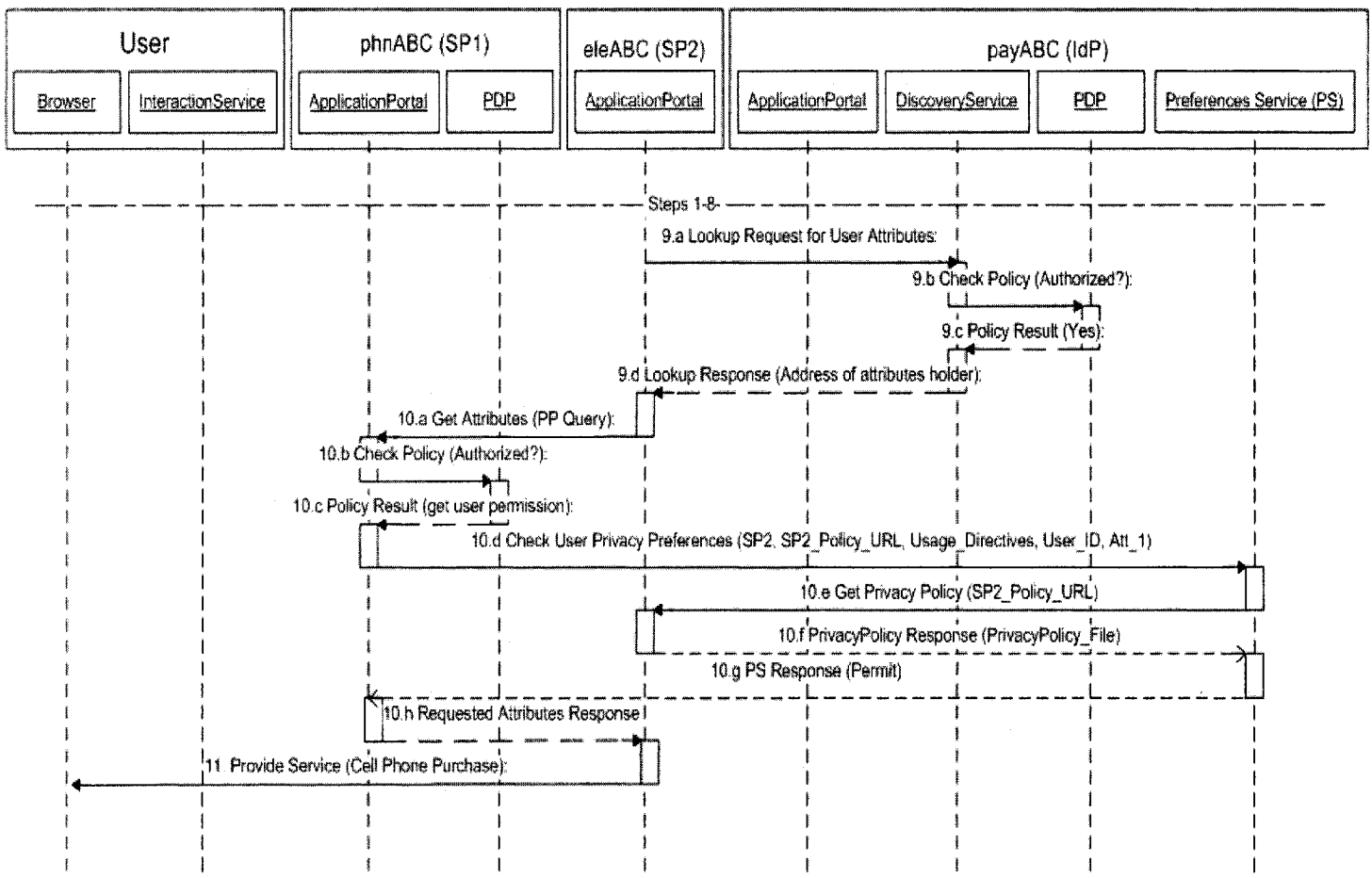

Figure 5.1: Scenario-I interaction diagram

This section in the diagram represents the request from SP2 for the some user attributes. SP2 needs these attributes before it can offer the service to the user. Up to step 10.c is similar to the use case scenario in figure 3.4. In step 9.a, SP2 sends an ID-WSF Query message (i.e. lookup request) to the discovery service. In step 9.b, the IdP acts as a PEP and sends a request to the PDP to find out whether SP2 is authorized to get information about the attribute provider possessing the user attributes. The access policy in the IdP PDP allows this in step 9.c. In step 9.d, the DS responds to SP2 with an ID-WSF Discovery Service QueryResponse in which ResourceOfferings (by SP1) that match with specified ResourceID and ServiceType are embedded. SP2 receives SP1's ResourceOffering, and sends an ID-SIS Personal Profile Query message to SP1 in step 10.a (Get Attributes), to receive the necessary attributes of the user. SP1 checks its local policy by sending an authorization request to the hosted PDP service as in step 10.b (SP1 is both a PEP and a PDP). Since the user never gave permission for SP2, there is no permit or deny policy result as in step 10.c. Therefore, SP1 requires the user permission first. In step 10.d instead of asking the user through the interaction service, SP1 sends a request "CheckUserPrivacyPreferences" to the user PS that is hosted by the user IdP in this scenario. The request includes the following parameters: the SP2 identifier, SP2 privacy policy URL, SP2 request usage directives, user ID and the required attributes. The user set her privacy preferences in this CoT at earlier stage (after the identity federation introduction) in the PS as follows:

"Any type of personal information can be exchanged for the purpose of completing and supporting an activity for which data was provided as long as the recipient is in the same identity federation circle of trust and the information is not retained after defederation. The user information may be shared with legal entities following different practices but after providing appropriate warning prompts and obtaining user permission"

This will be as follows in XPref:

```
<RULESET>
   <RULE behavior="request" condition="/POLICY [
         every $stmt in STATEMENT satisfies (
            every $recipient in $stmt/RECIPIENT/*,
            every $purpose   in $stmt/PURPOSE/*
```

```
            every $retention in $stmt/RETENTION/* satisfies
            (  name($purpose)    = "current" and
               (  name($recipient) = "ours" or
                  name($recipient) = "SP-same-CoT"     ) and
               (  name($retention) = "federation" or
                  name($retention) = "no-retention"    )
            )
   ) ]" />
   <RULE behavior="limited" prompt="yes"
      promptmsg="Warning! Data may be shared with legal entities
      following different practices. Do you want to continue (using
      limited access)?"
      condition="/POLICY [
      (  every $stmt in STATEMENT satisfies (
               every $recipient in $stmt/RECIPIENT/*,
               every $purpose   in $stmt/PURPOSE/*
               every $retention in $stmt/RETENTION/* satisfies
               (  name($purpose)    = "current" and
                  (  name($recipient) = "ours" or
                     name($recipient) = "SP-same-CoT" or
                     name($recipient) = "other-recipient"    ) and
                  (  name($retention) = "federation" or
                     name($retention) = "no-retention"    )
               )
         )
         and   {-- to make sure that there is at least one retention =
                  "other-recipient" --}
         (  /STATEMENT/RECIPIENT/*
               [name(.) = "other-recipient"]
         )
   ]" />
<RULE behavior="block" condition="true"/>
</RULESET>
```

In step 10.e, the user PS requests the SP2 privacy policy for this CoT and gets the policy document back (in P3P format) in step 10.f. The PS evaluates the SP2 privacy policy against user privacy preferences. If the SP2 privacy policy matches user privacy

preferences, the result of the evaluation will be "request". In this case, the PS sends a "PS Response (Permit)" back to the AP which is SP1 as in step 10.g. The AP now responds to SP2 with an ID-SIS Personal Profile QueryResponse message in which the user's attributes (e.g., SIM type) are embedded (step 10.h). The SP2 privacy policy that matches the user preferences may look simply as follows:

```
<POLICY xmlns="http://www.w3.org/2000/12/P3Pv1"
        discuri="http://www.scenario-1.com/privacypolicy-CoT-3.html">
   <ENTITY>
    <DATA-GROUP>
      <DATA ref="#business.name">eleABC</DATA>
    </DATA-GROUP>
  </ENTITY>
  <ACCESS>
    <all/>
  </ACCESS>
  <STATEMENT>
    <PURPOSE>
      <current/>
    </PURPOSE>
    <RECIPIENT>
      <ours/>
      <SP-same-CoT/>
    </RECIPIENT>
    <RETENTION>
      <federation/>
    </RETENTION>
    <DATA-GROUP>
      <DATA>
      ...
      </DATA>
    </DATA-GROUP>
  </STATEMENT>
</POLICY>
```

In case the `<RECIPIENT>` tag in the SP privacy policy also includes `<other-recipient/>`, this means the user information may be shared with legal entities following different practices. In this case, the user second preference (second rule) fires with a "limited" behavior as a result. The PS requires the user permission first, if she is available online. So, PS sends a request for user permission to the interaction service. In

this request, the PS provides the user with appropriate warning prompts. After the user gives consent, PS responds to the query from SP1.

The user PS is also supposed to send a log of these different transactions to the user audit trail service that is hosted by a trusted third party. Moreover, the user AP sends the following log: sending user attribute to a service provider upon service provider request and offering some user attributes by updating the discovery service. At a later stage, the audit trail service analyzes these logs and warns the user in case of any possible security or privacy violation. In addition, the user can access this service and audit the various logs. Consequently, the user may change her privacy preferences upon noticing some security or privacy violations or user data abuse.

This scenario demonstrates the following improvements:

- There is now a more accurate and automatic mechanism to evaluate the privacy practices of the service provider requesting the user attribute. This implies an enhancement of user privacy in this architecture.
- In case the user privacy preferences require his/her consent before releasing the data, there is now a privacy-aware way to request user consent via PS. The PS provides the user with all the necessary warning and advice before the user consents.
- The attribute provider does not need to request direct user consent for the user's attribute release each time there is a new attribute request that is not registered in the attribute provider PDP.
- The user does not need to give his/her consent for attribute release each time there is a new attribute request sent to the user attribute provider. Moreover, the user usually needs to set his/her privacy preferences only once. This means the Liberty identity federation architecture is more convenient to the user.
- Having the privacy practices evaluation process at the user PS that is hosted by a trusted party (e.g. user IdP) implies enhancement to the user security and privacy.

5.1.2 Use Case Scenario II

In this use case scenario, we examine the user involvement in the federation. This begins when the user first chooses her identity provider in the circle of trust. Then the identity

provider will notify its eligible users of the possibility of federating their local identities among the members of the affinity group and will solicit permission to facilitate the introduction of the user to the members of the affinity group. The user is expected to visit some of the service providers, federate her identity with some of them, and request some services. Diagram 5.2 shows the cooperation and interactions between different Liberty components after having our new services in place.

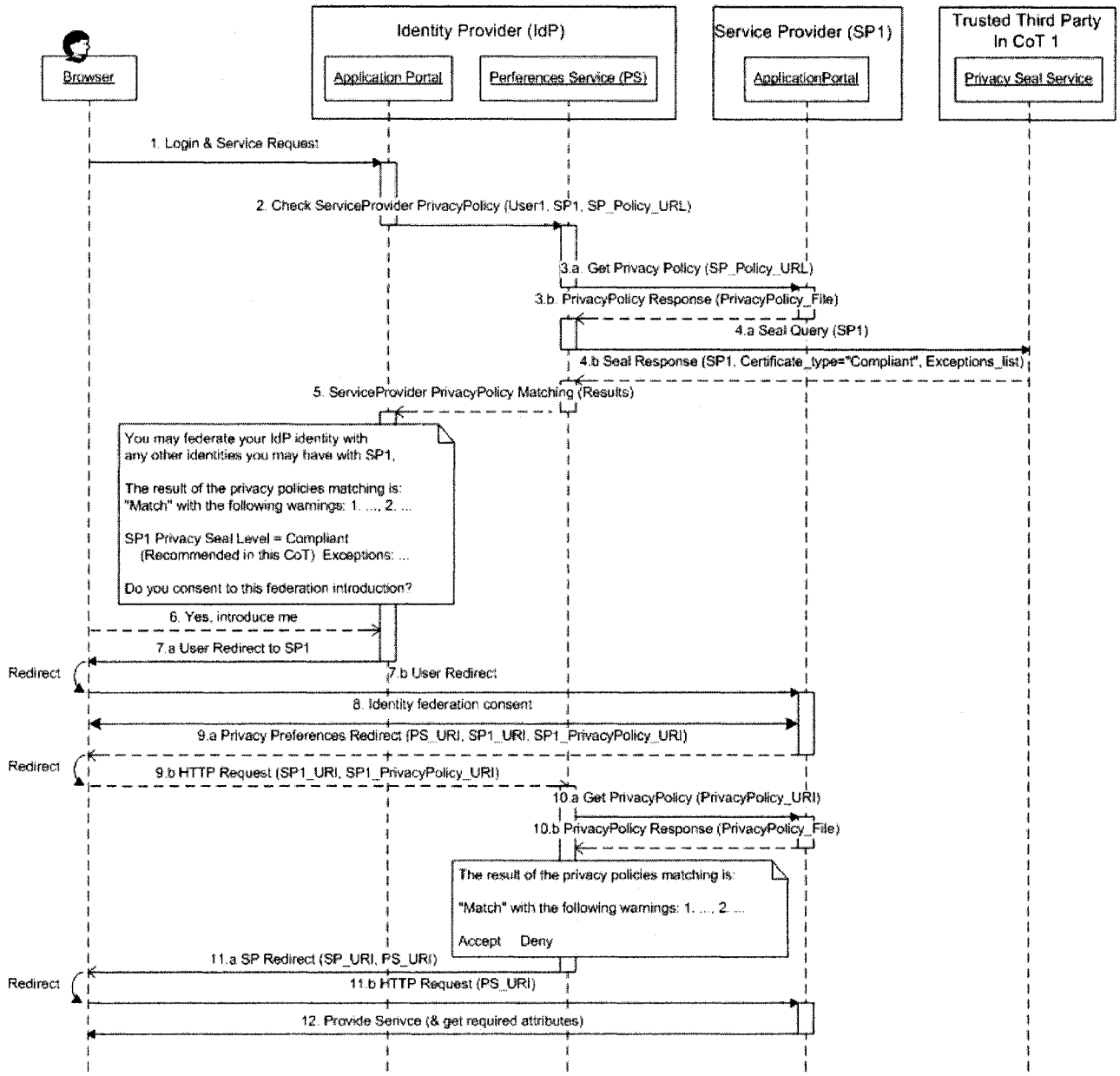

Figure 5.2: Scenario-II interaction diagram

At first, the user may need to consult the Liberty seal service before choosing an IdP. The user IdP in this scenario is Bank_ABC and the service provider (SP1) is BookStore_ABC. The user has an account with this bank and would like to utilize the online banking in ordering a book from the bookstore (SP1) which is in the same CoT.

After the user logs in to her IdP in step 1, the IdP would like to introduce the user to other members in the CoT. Instead of getting the user permission immediately, the IdP sends "Check Service Provider Privacy Policy" request to the user PS in step 2. The PS gets the SP1 privacy policy in steps 3.a and 3.b. The PS sends a "Seal Query" to the Liberty seal service to find out whether this SP is certified or not and what kind of seal it has as in steps 4.a and 4.b. After gathering this information about SP1, the PS compares the Liberty seal level stated in SP1 privacy policy with the one received from the Liberty seal service to verify similarity. Next, the PS matches SP privacy policy against user privacy preferences.

As we described in section 4.3.2, both the user preferences and SP privacy policy consist of two parts: a typical Web privacy preferences/policy and Liberty identity federation privacy preferences/policy. The second part can be used in this scenario in the following cases (not shown in the interaction diagram):

- Before the user selects an IdP, the user PS matches the IdP identity federation privacy policy against user identity federation privacy preferences to find if there is any inconsistency in the different identity federation choices. For example, the possible recipients of the federation introduction information or the Web method for redirecting the user using SSO mechanism. If the user is given appropriate warnings from the PS about any inconsistency, the user may choose a different IdP.

- Before the user consents to federate her identity with the SP, the user PS matches the IdP identity federation privacy policy against user identity federation privacy preferences to find out if this SP will enable the user to sign the federation consent during federating user identity.

Simple flexible user Web privacy preferences may contain the following:

"Any type of personal information can be exchanged with providers that have a Liberty seal "Very Compliant" or "Compliant" as long as this data will not be used for contacting visitors for marketing and the recipient is not a service provider in another CoT"

This will be as follows in XPref:

```
<RULESET>
   <RULE behavior="request" condition="/POLICY [
      (  /DISPUTES-GROUP/*
         [name(.) = "http://www.LibertyPrivacySeal.org/VeryCompliant" or
          name(.) = "http://www.LibertyPrivacySeal.org/Compliant"]
      )
      and
      (  every $stmt in STATEMENT satisfies (
               every $recipient in $stmt/RECIPIENT/*,
               every $purpose   in $stmt/PURPOSE/*  satisfies
               (   name($purpose)   != "telemarketing" and
                   name($recipient) != "SP-other-CoT"
               )
      )
   ]" />
<RULE behavior="block" condition="true"/>
</RULESET>
```

On the other hand, a simple SP1 Web privacy policy may look as follows:

```
<POLICY xmlns="http://www.w3.org/2000/12/P3Pv1"
        discuri="http://www.scenario-2.com/privacypolicy-CoT-1.html">
   <ENTITY>
     <DATA-GROUP>
       <DATA ref="#business.name">BookStore_ABC</DATA>
     </DATA-GROUP>
  </ENTITY>
  <ACCESS>
    <all/>
  </ACCESS>
  <DISPUTES-GROUP>
    <DISPUTES resolution-type="independent"
              service="http://www.LibertyPrivacySeal.org/Compliant"
              short-description="Compliant Liberty Privacy Seal">
```

```
            <IMG src="http://www.LibertyPrivacySeal.org/Logo.gif"
               alt="Privacy Seal Logo"/>
        <REMEDIES>
          <correct/>
        </REMEDIES>
      </DISPUTES>
    </DISPUTES-GROUP>
    <STATEMENT>
      <PURPOSE>
        <current/> <admin/> <develop/> <tailoring/>
      </PURPOSE>
      <RECIPIENT>
        <ours/>
        <IdP-same-CoT/> <IdP-other-CoT/> <SP-same-CoT/>
      </RECIPIENT>
      <RETENTION>
        <stated-purpose/>
      </RETENTION>
      <DATA-GROUP>
        <DATA>
        ...
        </DATA>
      </DATA-GROUP>
    </STATEMENT>
</POLICY>
```

Since the SP1 privacy policy matches the user preferences and SP1 has obtained a "compliant" liberty seal, PS returns a "match" result to the user IdP. The IdP offers to introduce the user to SP1. In the introduction consent request, the IdP shows to the user the policy match results together with the corresponding Liberty seal level of SP1. Upon user consent in step 6, the user chooses to be redirected to SP1 (to order the required book) in step 7. In step 8, the user agrees to federate her identity with SP1. Since SP1 will need specific personal data from the user to process the order and since either the SP1 privacy policy or user privacy preferences may change with time (the user may not be redirected immediately and may visit SP1 at a later time), there is a need for further policies matching. Thus, the PS gets the SP1 privacy policy and matches it against the user privacy preferences in step 10. If the two policies match for the required data (or may be after the user consent in case of "limited" result), the PS redirects the user back to SP1 in step 11. SP1 can now get the required personal information from the user to order the book.

This scenario demonstrates the following improvements to Liberty identity federation architecture:

- The user gives her federation introduction consent after matching SPs privacy policies with her privacy preferences. Thus, the user is given warning in case of discrepancies before giving consent which will enhance user privacy.
- The user PS will verify the SP Liberty seal and will warn the user in case of a mismatch which will enhance user privacy.
- The SP privacy policy is checked against user privacy preferences by user PS each time the SP requests personal information from the user in a new Web session.
- The policies are matched in an automated mechanism and in most of the cases will not need any user interaction or involvement.

5.2 Compatibility with Liberty Architecture

In this section, we assess the compatibility with the existing Liberty frameworks. A discussion of how our proposed services would conform and integrate with Liberty identity federation architecture is given for the following aspects: usability and deployment, reliability, and scalability of the proposed solutions and services. Right from the beginning, throughout the design of these services, we made sure of the compatibility with other services and components in Liberty architecture. In fact, our proposals comply with Liberty specification and they are built on top of existing Web protocols and standards. In reality, all communicated messages between our new services and other components in the Liberty identity federation architecture are sent using standard protocols (SAML protocol messages within SOAP messages over HTTP). This is similar to the other services in the current Liberty specifications. Moreover, we intended not to specify new service interfaces for service metadata management, schema that accompanies the Liberty ID-WSF SOAP binding specification, or other related SAML messages. We believe that once we describe the conceptual model of these services and illustrate the functionality of the different operations offered by the services, it would be best to leave specifying the actual SOAP messages and other details to the Liberty Alliance team to agree on and define the detailed specification.

5.2.1 Usability and Deployment

In this section, we discuss usability and deployment issues related to our three new Liberty services. Usability is a great concern for any Internet service that is supposed to be used by general users. Usability is also one of the identity federation architectures' main goals. The following are some important usability aspects in our proposals:

- In our privacy analysis in section 3.3.3, we suggested solutions for the privacy breaches that we have identified. These solutions need more investigation about their usability and deployment. For example, the timestamp signed by the discovery service (as expiry date and time) to solve the problem that the SP may store locally the address of the attribute resource holder.
- In the user privacy preferences service (PS), the usability of the user interface that enables the user to enter her Web and identity federation privacy preferences is important. This user interface can be as simple as an XPref editor for advanced users to edit their preferences directly in the XPref language. A graphical easy-to-use user interface is absolutely necessary for this service to function properly. Although this user interface design is out of the scope of this thesis, we emphasize the importance of an appropriate design. Poor user interface in this service will affect significantly the usability of this service.
- For security and privacy reasons, the user IdP is recommended to host the user PS in the CoT. We believe this will improve the usability of PS service for two main reasons. First, the user needs to login first to the service (since this service is hosted by the IdP, the user probably already logged in). Second, in the identity federation architecture, CoTs are connected through IdPs, this makes it easier to export user privacy preferences between CoTs.
- By having the PS in Liberty architecture, we decreased dramatically the number of times the user consent is required. Thus, the user is no longer overwhelmed by many access permission requests which improves the usability of the overall architecture.
- In case the PS decision result indicates requesting user permission (in case the user privacy preferences designate that), we still face the problem of user

unavailability online. In this case the request to the PS is on hold until the user is online (or it could timeout to "Deny" result).

- As we showed in section 4.3.1 (C), the PS removes the need for privacy preferences plug-in in the user agent. The SP redirects the user to the PS to compare the SP privacy policy against the user privacy preferences before the SP requests any personal information from the user. We believe this enhances the usability of the architecture since the user does not need to setup or configure any browser-based privacy engines.
- The user privacy preferences service can also be used when the user IdP wishes to introduce the user to other service providers in the affinity group 4.3.1 (C). In this case, the PS checks and compares the SP privacy policy on behalf of the user. This also increases the usability of the architecture.
- The Liberty identity federation privacy preferences document enables the user to specify what identity federation decision can be taken about her by the Liberty provider (under predefined circumstances). Using the user PS, the IdP and the SP could now have many identity federation transactions in place without the need to interrupt the user.
- The user PS with the help of the Liberty seal service verifies automatically the service provider seal level without user involvement. The verification can be done in both the federation introduction and the identity federation consent.
- Using XPref, we can specify preferences that are hard to specify using APPEL. This will be easier whether it is the user who will directly specify the privacy preferences in XPref or the Web application that will convert the user preferences from a dedicated Web user interface.

5.2.2 Reliability

As we mentioned, all communicated messages between our new services and other components in the Liberty identity federation architecture are sent using Liberty message format. They are encapsulated in either SAML assertions (over SOAP) or directly in SOAP messages (over HTTP) and that is compatible with Liberty specifications. Second, in the Liberty preferences service, the most costly function is the policy matching which

is matching user preferences in XPref with service provider privacy policy in P3P. XPref uses a strict subset of XPath 1.0 for writing rule conditions (except the quantified expression "every" from XPath 2.0) and there are several reliable XPath libraries. In addition, having a small subset implies smaller footprint implementations of XPref. Moreover, even though XPref is strictly based on XPath, it does not use many of the expensive features of XPath. Although there are many off-the-shelf XPath implementations, it is possible to build a specialized implementation with optimized processing speed and smaller memory footprint. In addition, there is no use case scenario with intensive or frequent calls for this function. Therefore, we are not expecting a reliability issue in the PS functionality. We believe the reliability of this service depends essentially on the reliability of the provider hosting and running the service. We suggested the user IdP to host this service which is supposed to be very reliable in the CoT.

Likewise, both the Liberty seal service and the audit trail service have no complicated functions that may require intensive processing time or memory space. However, the audit trail service is called frequently throughout the architecture to record many transactions and different logs. The availability of this service is important.

5.2.3 Scalability

The scalability of the three services relies on their ability to handle many requests at the same time. As we discussed in section 5.2.3, the limited processing and memory resources required for these services keep them from being the bottleneck in the Liberty identity federation architecture. However, the volume of data received by both the Liberty privacy preferences service and audit trail service is big compared with other services in the architecture. The emphasis here is the need for high capacity communication channels to these services.

5.3 The Applicability of the Proposed Services to Other Identity Federation Architectures

In this section, we discuss the applicability of the proposed services and solutions to other identity federation architectures. In particular, we consider the Shibboleth and Microsoft InfoCard architectures.

Shibboleth

We described briefly the Shibboleth identity federation architecture in the background chapter (section 2.5.2.3). Shibboleth supports both SSO and identity federation mechanisms. It also has a framework for attribute exchange. The key function of this architecture is typically to determine if a person using a web browser has the permissions to access a resource at a target resource based on information such as being a member of an institution or a particular class. For example, students and faculty access to a higher education institute resources or internet users access to digital content providers resources. However, Shibboleth focuses only on this scenario, that is, the SP requests some user attributes from her IdP in order to give the user access to an SP resource. For SSO and identity federation mechanisms, Shibboleth is very similar to the Liberty architecture. In fact, Shibboleth is considered a subset of the Liberty identity federation architecture.

Figure 5.4 shows the typical flow for this scenario. In step 1, the user requests a target resource at the SP. The SP redirects the user to the single sign-on (SSO) service at the IdP in step 2. The SSO service processes the authentication request and performs a security check. If the user does not have a valid security context, the IdP identifies the user (step 4). Once the user has been identified, the SSO service obtains an authentication statement from the authentication authority (step 5). In step 6, The SSO service redirects the user to the assertion consumer service at the SP (the assertion consumer service is the service provider endpoint of the SSO exchange. It processes the authentication assertion returned by the SSO service, initiates an optional attribute request, establishes a security context at the SP, and redirects the client to the desired target resource). The assertion consumer service verifies user authentication and passes control to the attribute requester (step 7). The attribute requester POSTs a SAML SOAP message to the attribute authority at the

user IdP (step 8). The attribute authority at the IdP checks the request and responds with an attribute assertion (step 9) subject to attribute release policies in the internal PDP. The SP site uses attributes for access control and other application-level decisions (step 10).

As observed in the previous scenario, there is no discovery service or interaction service as in the Liberty architecture. Moreover, there is no SP-SP attribute exchange profile as in Liberty. The user controls access to her attributes through setting the attribute release policies which is part of the attribute authority in the user IdP. In these policies, the user specifies in detail the name of SPs, what attributes they have access to, and under what conditions they may be accessed. Notice that SPs always request user attributes from the IdP (e.g. when the student accesses digital library resources from off-campus, accesses a co-taught class web site at another university, and so on).

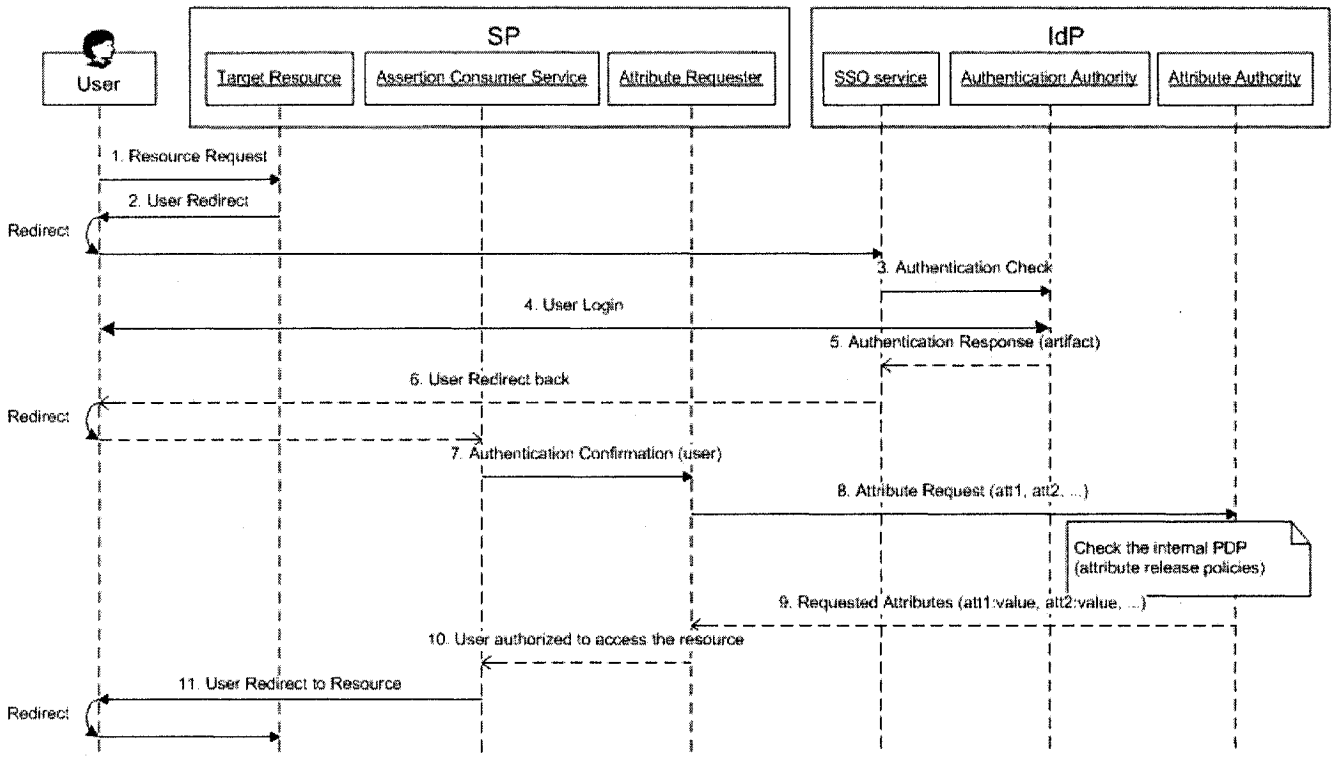

Figure 5.3: sequence diagram for the typical Shibboleth scenario

We can apply our privacy preferences service to the Shibboleth architecture by adding this service as part of the user IdP. Figure 5.5 depicts the expected flow after adding the service. Steps 8.a~8.d are similar to scenario B in section 4.3.1.

Notice that the user privacy preferences document in the PS is not specific for each SP. Thus, there is no need for user involvement each time a new provider requests user attributes. The IdP PDP is now an access control policy for the IdP itself as in Liberty (not the user as in Shibboleth), so we will not need the attribute release policies any more. Furthermore, the user PS compares the SP privacy policy against user preferences and could also interact with the seal service to find out the SP seal level. This design can also be applied if Shibboleth is extended to include SP-SP attribute exchange or in case of merging more than CoT. For the other two services, the seal and the audit trail services, the necessary changes seem similar to the changes we made to the Liberty architecture. Nevertheless, a more comprehensive study is required for this architecture.

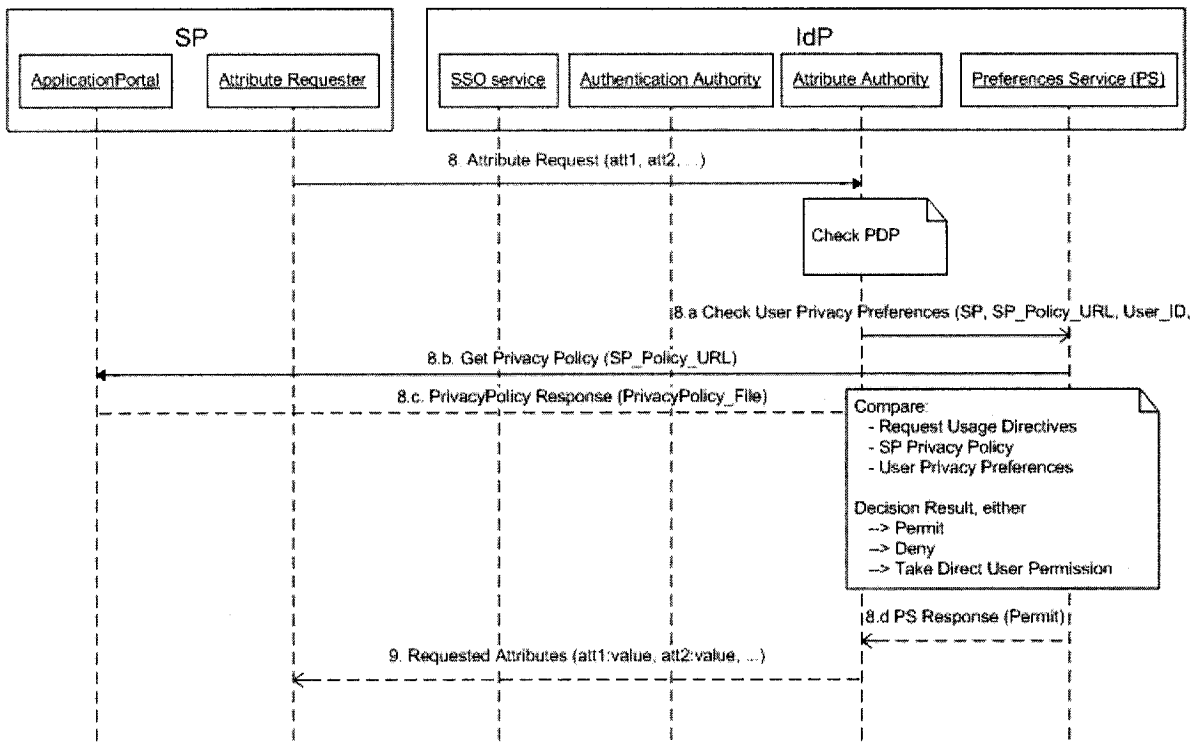

Figure 5.4: Shibboleth scenario sequence diagram after having user PS in place

Microsoft InfoCard

We gave a simple InfoCard use case scenario interaction diagram and we described InfoCard in section 2.5.2.2. The InfoCard model gives more functionality to the user

agent. In InfoCard, user IdP and SP do not interact directly with each other, instead, they exchange security tokens via the user agent. Thus, we found the best place for our privacy preferences service is the user agent itself. Figure 5.5 depicts the expected flow after adding the service.

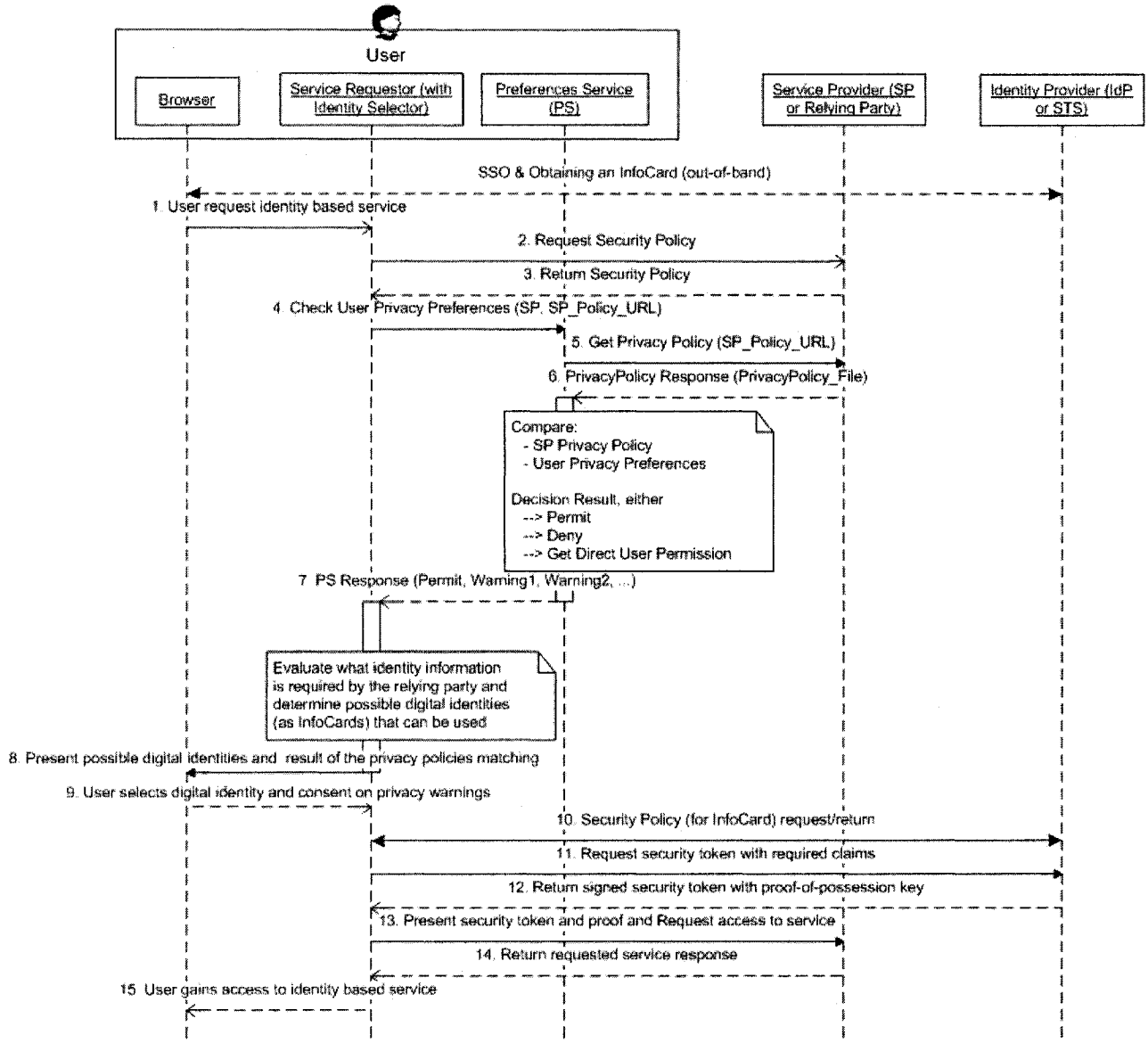

Figure 5.5: InfoCard scenario sequence diagram after having user PS in place

The service requester application obtains the security policy of the target service using metadata exchange (steps 2 and 3). The target service policy requires that the

requester obtains and presents a security token issued by the delegation Authentication/Authorization security token service (STS which is part of the IdP). Now, the service requester application sends a request to the user PS in step 4. The AP will first get the privacy policy address of the SP requesting the user attribute (step 5 and 6). The user PS compares the SP privacy policy against user preferences (and could also interact with the seal service to find out the SP seal level). The matching results are sent back to the service requestor in step 7.

The service requester application requests the identity selector component to produce a security token that can satisfy the target service policy. The identity selector displays the matching InfoCards which can satisfy the target service policy, and the user selects and approves one for use (step 8). Moreover, the user gives consent with respect to the SP privacy warnings (results from PS). Before the identity selector on the service requester can request a security token from the identity provider STS for the selected InfoCard, it first obtains the security policy of the identity provider STS using metadata exchange to determine the security binding to use for the message channel (step 10). The InfoCard specifies the required authentication mechanism to request security tokens. The identity selector authenticates to the identity provider STS using an acceptable user credential as specified by a credential selector in the InfoCard. It requests a security token of the appropriate type with the required claims as specified by the target service and receives the token (step 11 and 12). The identity selector hands the requested security token to the service requester application. The service requester application presents the token obtained and provides proof-of-possession to the target service to gain access (step 13). Notice that as a design option, we could have the PS hosted by the IdP; however, this will include direct interaction between the IdP and the SP.

The privacy seal service can be hosted by a trusted third party. In this case, the user PS in the user agent requests the seal service before comparing privacy policies. The audit trail service fits to be part of the user agent. The service requestor and the privacy preferences service interact with the audit service to log the important transactions. Our new services appear to fit well within the InfoCard architecture; however we believe further analysis is definitely required for the InfoCard model especially since its specification is not released yet.

Chapter 6

Conclusions and Future Work

In this chapter, we summarize our contributions in this thesis. For each contribution, we list our conclusions. Possible improvements of our proposals, which are subject to future work, are also discussed.

6.1 Contributions and Conclusions

Our objective in this thesis is to enhance consumer privacy in the Liberty identity federation architecture such that consumer trust in using Liberty-enabled providers' services will be increased and hence will lead to greater adoption of Liberty identity federation standards.

In our first contribution of this thesis, we documented previously unknown privacy breaches and concerns in Liberty Alliance identity federation frameworks and proposed solutions and recommendations to the identified privacy breaches to enhance consumer privacy. Knowing the advantages of identity federation and the need of a set of technical and business standards to regulate and facilitate the different federation mechanisms, the Liberty Alliance identity federation architecture is an attractive choice that is evolving as a widely-accepted open standard for federated network identity. However, any successful identity federation architecture has to satisfy sufficiently user confidentiality and privacy in particular in addition to online service providers' privacy. Although one of the key objectives of the Liberty Alliance project is to enable consumers to protect the privacy and security of their network identity information, we believe that this objective is biased towards business goals of the Liberty architecture rather than giving priority to consumer privacy. In fact, our privacy analysis identifies previously unknown privacy breaches and

concerns in the current Liberty specifications. We found that some of the identified breaches are indirect results of privacy weaknesses in Internet protocols and browsers, upon which the Liberty specifications are built. On the other hand, some other breaches are a result of non-determinism and leniency in the Liberty specification. The proposed solutions to the identified privacy breaches are not intended to be complete or fully-specified solutions; many proposed solutions are either recommendations or improvements to enhance consumer privacy in a general sense. Although we studied and described some of these solutions in detail (in chapter 4), many other solutions need further investigation and evaluation. Furthermore, there are several other important privacy concerns from other perspectives: service-provider privacy and government privacy. We have identified and analyzed some of these concerns; however, a further detailed study is required. We believe that the identified privacy breaches and concerns should motivate the development of more privacy-aware, strict, and comprehensive Liberty specifications that will enhance consumer privacy in federated frameworks. We also showed how some of the identified privacy breaches also exist in other identity federation architectures. This is because identity federation architectures share many concepts and mechanisms (e.g., SSO and federated identity).

In the second contribution, we introduced new services to Liberty frameworks: User Privacy Preferences Service, Privacy Seal Service, and Audit Trail Service. The user privacy preferences service enhances user privacy in many cases. In particular, attribute request at the attribute provider, SP attribute request from the user, and service providers' introduction at the IdP. The key advantage of the proposed privacy preferences service is that some access permission requests can be directed first to this service to find out whether the user's default preferences allow the requested access. This reduces the need of the user agent (or simply the user) being online to respond to the different user attribute access requests. Moreover, the user privacy preferences document is accessible to the user whether from the user machine or a different machine since the privacy preferences are not stored in the user machine. Furthermore, the user can change his privacy preferences for certain data after he releases this data to a SP and so this SP may share this data according to different user privacy preferences over time. This is important in case the user changes his privacy preferences upon realizing some privacy

violations by some SPs (with the help of the audit trail service). In an identity federation environment, the privacy preferences service fills the need for a separate user-agent privacy engine which compares SP privacy policy against user preferences. While the privacy preferences service enhances user privacy, it also offers a simplified and fast user experience within the identity federation architecture. The second proposed service, the Liberty seal service, informs the user and other providers how the SP privacy policies comply with the Liberty identity federation privacy requirements. The Liberty seal has four different levels of compliance: "Very Compliant", "Compliant", "Some Compliance", and "Not certified". The Liberty seal service can be used in the following cases: federation introduction consent, identity federation consent, and attribute provider request. The third service, the audit trail service, helps the user to know if any privacy violation has been committed by any SP allowing him to take the appropriate action for future transactions and permitting him to update his privacy preferences accordingly.

All three proposed services fit best within the Liberty identity Web services framework (ID-WSF) specification. It is important that any privacy-enhancing service in the identity federation is hosted by a trusted provider in the circle of trust. We also showed that all the proposed services can be used to enhance consumer privacy in other identity federation architectures with minor changes. In particular, we have shown this with the Shibboleth and Microsoft InfoCard architectures.

In the third contribution, we extend P3P to accommodate identity federation privacy requirements by adding new elements to the P3P terminology. First, we suggested a typical content of user privacy preferences in the Liberty identity federation architecture that consists of two parts: typical Web privacy preferences and Liberty identity federation privacy preferences. Typical Web privacy preferences enables the user to specify what personal information can be collected about her under what conditions and for what purposes to which recipients and for how long. Liberty identity federation privacy preferences enables the user to specify what identity federation decision can be taken about her by the Liberty provider, under what conditions, for what purposes, to which recipients, and for how long. We also proposed using the XPref language for Liberty User privacy preferences service and identified some APPEL issues in this context. The XPref privacy preferences language is both more efficient and easier-to-use than APPEL

(APPEL is complex and error prone in many cases). We have shown several examples in which it is more appropriate to use XPref as the privacy preferences language in the Liberty user privacy preferences service.

6.2 Future Work

Future work is summarized as follows:

- Privacy analysis for new components in the Liberty architecture (e.g. ID-WSF People Service).
- Privacy analysis of other identity federation architectures (in particular, the InfoCard model for which release of the first specifications is expected in the first quarter of 2007). In addition, a detailed study of how to apply our proposals to the InfoCard model to enhance consumer privacy would be available.
- Further investigation and evaluation for some other solutions that we suggested for the identified privacy breaches.
- Designing an easy-to-use UI for the user to enter his identity federation privacy preferences which will be encoded in XPref.
- Detailed usability evaluation of the proposed services.
- Building on top of ID-WSF V2.0 (if and when finalized) to implement our proposals and conducting a more accurate evaluation.

6.3 Concluding Remarks

Any Internet identity federation architecture is expected to have the features of SSO, federated identity, and provider-to-provider user personal information exchange. These features are usable only if they are efficient, easy-to-use, and "privacy-aware" for both consumers and businesses. All of those features include exchanging some user attributes, either between the user and the provider or between different providers. It is not sufficient to build a Web identity federation system without taking into consideration consumer privacy. Instead, it is very important to have consumer privacy in mind as a priority objective in designing the identity federation standard right from the beginning

since it is difficult and inefficient to add "consumer-privacy" to an existing standard. Enhancing consumer privacy in identity federation architectures will increase consumer trust which is essential and significant in using identity federation services. We believe that the most successful identity federation standard will be the one that takes consumer privacy strongly into consideration.

References

1. Aarts, R., Björksten, M., Deadman, S., Duserick, B., Karhuluoma, N., et al., *"Liberty architecture framework for supporting Privacy Preference Expression Languages (PPELs)"*. Nov 2003, Version 1.0, Liberty Alliance Project. Available from: http://www.projectliberty.org/about/whitepapers.php.

2. Adams, C. and Barbieri, K., "Privacy Enforcement in E-Services Environments": in *Privacy Protection for E-Services*, G. Yee, Ed., Idea Group Publishing, 2006, pp. 172-202.

3. Agrawal, R., Evfimievski, A., and Srikant, R. "Information sharing across private databases". In *Proc. of the 2003 ACM SIGMOD Int. Conf. on Management of Data*, ACM Press, 2003.

4. Agrawal, R., Kiernan, J., Ramakrishnan, and Xu, S. Y. "Hippocratic Databases". In *Proceedings of the 28th VLDB Conference*, Hong Kong, China, 2002.

5. Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. "An Xpath-based preference language for P3P". In *Proceedings of the twelfth international conference on World Wide Web, pages 629-639*, 2003. ACM Press.

6. Ahn, G.-J. and Lam, J. "Managing privacy preferences for federated identity management". In *Proceedings of the 2005 workshop on Digital identity management*, Fairfax, VA, USA, November 2005. ACM Press.

7. Ahn, G.-J., Shin, D., and Hong, S.-P. "Information Assurance in Federated Identity Management: Experimentations and Issues". In *Proceedings of the 5th International Conference on Web Information Systems Engineering: Web Information Systems – WISE 2004*, Brisbane, Australia, November 2004. LNCS, Volume 3306, Jan 2004, Pages 78 - 89.

8. Anderson, A. "A Comparison of Two Privacy Policy Languages: EPAL and XACML". Sun Microsystems Laboratories, September 2005. Available from: http://research.sun.com/techrep/2005/smli_tr-2005-147/TRCompareEPALandXACML.html [Accessed: October 2005].

9. Ashley, P., Hada, S., Karjoth, G., Powers, C., and Schunter, M. "Enterprise Privacy Authorization Language (EPAL 1.2)". W3C Member Submission, November 2003. Available from: http://www.w3.org/Submission/EPAL/ [Accessed: November 2005.

10. Asonov, D. "Private Information Retrieval: an overview and current trends". In *Proceedings of the ECDPvA Workshop, Informatik 2001*, Vienna, Austria, September 2001.

11. Benassi, P., "TRUSTe: an online privacy seal program". Communications of the ACM, February 1999. **42**(2): p. 56 - 59.

12. Beres, Y., Bramhall, P., Mont, M. C., Gittler, M., and Pearson, S., "Accountability and Enforceability of Enterprise Privacy Policies". HPL-2003-119, Trusted Systems Laboratory (TSL), Hewlett-Packard Laboratories, 2003.

13. Bhargav-Spantzel, A., Squicciarini, A. C., and Bertino, E. "Establishing and protecting digital identity in federation systems". In *Proceedings of the 2005 workshop on Digital identity management*, Fairfax, VA, USA, November 2005. ACM Press.

14. Blarkom, G. W. v., Borking, J. J., and Olk, J. G. E., "Handbook of Privacy and Privacy-Enhancing Technologies - The case of Intelligent Software Agents". 2003: College Bescherming Persoonsgegevens.

15. Brands, S. "Analysis of pseudonyms in SAML 2.0 & Liberty Alliance". 2/28/2005. The Identity Corner. Available from: http://www.idcorner.org/index.php?p=40" [Accessed: June 2006].

16. Brown, K. "A First Look at InfoCard". April 2006. MSDN Magazine, Security Briefs. Available from: http://msdn.microsoft.com/winfx/reference/InfoCard/default.aspx?pull=/msdnmag/issues/06/04/securitybriefs/default.aspx [Accessed: May 2006].

17. Brown, K. "Security Briefs: Step-by-Step Guide to InfoCard". April 2006. MSDN Magazine. Available from: http://msdn.microsoft.com/winfx/reference/InfoCard/default.aspx?pull=/msdnmag/issues/06/05/securitybriefs/default.aspx [Accessed: May 2006].

18. Buecker, A., Filip, W., Hinton, H., Hippenstiel, H. P., Hollin, M., et al., "Federated Identity Management and Web Services Security". Second ed. October 2005: IBM Redbooks.

19. BusinessWeek online. "Business Week/Harris Poll: A Growing Threat". March, 2000. Available from: http://businessweek.com/2000/00_12/b3673010.htm [Accessed: January 16, 2006].

20. Byun, J.-W., Bertino, E., and Li, N. "Purpose based access control of complex data for privacy protection". In *Proceedings of the tenth ACM symposium on Access control models and technologies*, Stockholm, Sweden, 2005. ACM press.

21. Canadian Standards Association. "Privacy Model Code (Code)". Available from: http://www.csa.ca/standards/privacy/Default.asp?language=English [Accessed: March 10, 2006].

22. Chaum, D., "Untraceable electronic mail, return addresses, and digital pseudonyms". Communications of the ACM, 1981. **24**(2): p. 84-90.

23. Cheskin Research. "Trust in the wired Americas". July, 2000. Available from: http://www.cheskin.com/assets/report-CheskinTrustIIrpt2000.pdf [Accessed: November 2005].

24. Chiu, D. K. W. and Hung, P. C. K. "Privacy and Access Control Issues in Financial Enterprise Content Management". In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05)*, Hawaii, USA, 2005. IEEE Computer Society.

25. Clark, J. and DeRose, S. "XML Path Language (XPath) Version 1.0". November 1999. W3C Recommendation. Available from: http://www.w3.org/TR/xpath [Accessed: March 2006].

26. Common Criteria for Information Technology - Security Evaluation -. "Part 2: Security functional requirements". August 1999, Annotated in 2003. Available from: http://www.commoncriteriaportal.org/public/files/part2.2003-12-31.pdf [Accessed: October 2005].

27. Cover, R. "Web Services Federation Language Provides Federated Identity Mapping Mechanisms". March 2004. Cover Stories. Available from: http://xml.coverpages.org/ni2003-07-08-b.html [Accessed: March 2006].

28. Daniel. "Differences Between XACML Version 1.0 and 2.0". January 7, 2005. Available from: http://blog.parthenoncomputing.com/xacml/archives/2005/01/the_differences.html [Accessed: October 2005].

29. Deadman, S. "Circles of Trust: The Implications of EU Data Protection and Privacy Law for Establishing a Legal Framework for Identity Federation". February 23, 2005. Liberty Alliance Project. Available from: http://www.projectliberty.org/specs/Circles_of_Trust_Legal_Framework_White_Paper_322200522576.pdf [Accessed: December 2005.

30. Ellis, S. and Oppenheim, C., "Legal Issues for Information Professionals, Part III: Data protection and the Media – Background to the Data Protection Act 1984 and the EC Draft Directive on Data Protection". Journal of Information Science, 1993.

31. Ellison, G. and Madsen, P., *"Liberty ID-WSF Security Mechanisms"*, version 2.0-03, Liberty Alliance Project. Available from: http://www.projectliberty.org/resources/specifications.php.

32. EPIC. "Electronic Privacy Information Center". Available from: http://www.epic.org/ [Accessed: May 2006].

33. European Commission. "Data Protection". Available from: http://europa.eu.int/comm/justice_home/fsj/privacy/index_en.htm [Accessed: February 10, 2006].

34. European Commission. "FAQs on the Commission's adequacy finding on the Canadian Personal Information Protection and Electronic Documents Act". Available from: http://europa.eu.int/comm/justice_home/fsj/privacy/thridcountries/adequacy-faq_en.htm#2 [Accessed: March 10, 2006].

35. FIPA. "BC Freedom of Information and Privacy Association". Available from: http://fipa.bc.ca/home/ [Accessed: May 2006].

36. Fox, S. "Trust and Privacy Online: Why Americans Want to Rewrite the Rules". August 2000. Pew Internet & American Life Project. Available from: http://www.pewinternet.org/pdfs/PIP_Trust_Privacy_Report.pdf [Accessed: January 7, 2006].

37. Gartner Group. "Industry watchdog Gartner Group". 2003. Available from: http://www.gartner.com [Accessed: March 23, 2006].

38. Gates, C. and Slonim, J. "Owner-controlled information". In *Proceedings of the 2003 workshop on new security paradigms*, Ascona, Switzerland, 2003. ACM Press.

39. Gonsalves, A. "Microsoft Nearly Finished With Windows Live ID Deployment". Apr 27, 2006. Information Week. Available from: http://www.informationweek.com/windows/showArticle.jhtml?articleID=187000723 [Accessed: April 2006].

40. Groß, T. "Security analysis of the SAML Single Sign-on Browser/Artifact profile". In *Proceedings of the 19th Annual Computer Security Applications Conference*, Las Vegas, NV, USA, Dec 2003. IEEE.

41. GUTH, R. A. "Microsoft Tests Software to Fight Identity Theft on Web". March 28, 2005. THE WALL STREET JOURNAL. Available from: http://online.wsj.com/article/SB111196644239490480.html?emailf=yes [Accessed: Novmber 2005].

42. He, Q., "Privacy Enforcement with an Extended Role-Based Access Control Model". NCSU Computer Science Technical Report, TR-2003-09, April 2003.

43. Hommel, W. "Using XACML for Privacy Control in SAML-Based Identity Federations". In *Proceedings of the TC-11International Conference, CMS 2005*, Salzburg, Austria, September 2005. Lecture Notes in Computer Science, Volume 3677, Sep 2005, Pages 160 - 169.

44. Internet2. "Shibboleth Introduction". Available from: http://shibboleth.internet2.edu/shib-intro.html [Accessed: January 2006].

45. Johnston, S. J. "Pondering Passport: Do You Trust Microsoft With Your Data?" September, 2001. PCWorld.com. Available from: http://pcworld.about.com/news/Sep242001id63244.htm [Accessed: January 10, 2006].

46. Kainulainen, J. "Liberty ID-SIS Geolocation Service Specification". 2005. Version: 1.0-12, Liberty Alliance Project. Available from: http://www.projectliberty.org/specs/draft-liberty-id-sis-gl-v1.0-12.pdf [Accessed: December 2005].

47. Karjoth, G. and Schunter, M. "A Privacy Policy Model for Enterprises". In *Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW'02)*, Cape Breton, Nova Scotia, Canada, 2002. IEEE Computer Society.

48. Kellomäki, S. "Liberty ID-SIS Contact Book Service Specification". 2005. Version 1.0, Liberty Alliance Project. Available from: http://www.projectliberty.org/specs/draft-liberty-id-sis-cb-v1.0-11.pdf [Accessed: December 2005].

49. Kellomäki, S. and Lockhart, R. "Liberty ID-SIS Employee Profile Service Specification". 2003. Version 1.1, Liberty Alliance Project. Available from:

http://www.projectliberty.org/specs/liberty-idsis-ep-v1.1.pdf [Accessed: December 2005].

50. Kellomäki, S. and Lockhart, R. "Liberty ID-SIS Personal Profile Service Specification". 2003. Version 1.1, Liberty Alliance Project. Available from: http://www.projectliberty.org/specs/liberty-idsis-pp-v1.1.pdf [Accessed: February 2006].

51. Koorn, R., Gils, H. v., Hart, J. t., Overbeek, P., and Tellegen, R., *"Privacy-Enhancing Technologies, White Paper for Decision-Makers"*. December 2004, Dutch Ministry of the Interior and Kingdom Relations. Available from: http://www.dutchdpa.nl/downloads_overig/PET_whitebook.pdf.

52. Landau, S., *"Liberty ID-WSF Security & Privacy Overview"*. 2003, Version 1.0, Liberty Alliance Project. Available from: http://www.projectliberty.org/resources/specifications.php.

53. Levine, S. A. and Kerr, J. A. "Canadian Privacy Laws: Countdown to January 1, 2004". Fasken Martineau DuMoulin LLP. Available from: http://www.fasken.com/web/fmdwebsite.nsf/AllDoc/FC3F9B5B1680725085256DC900591820/$File/ABACANADIANLAWNEWSLETTER.PDF!OpenElement [Accessed: March 20, 2006.

54. Liberty Alliance Project. Available from: http://www.projectliberty.org/ [Accessed: October 2005].

55. Liberty Alliance Project. "Introduction to the Liberty Alliance Identity Architecture". March 2003. Revision 1.0. Available from: http://www.projectliberty.org/resources/whitepapers/LAP Identity Architecture Whitepaper Final.pdf [Accessed: February 2006].

56. Liberty Alliance Project. "Liberty Alliance Whitepaper: Identity Theft Primer". December 2005. Available from: http://www.projectliberty.org/resources/id_Theft_Primer_Final.pdf [Accessed: January 2006].

57. The Department of Information Systems; The London School of Economics and Political Science (LSE). "The Identity Project" an assessment of the UK Identity Cards Bill and its implications. P 276-286. Available from: http://is2.lse.ac.uk/idcard/identityreport.pdf [Accessed: May 2006].

58. Madsen, P. and Takahashi, Y. K. K. "Federated identity management for protecting users from ID theft". In *Proceedings of the 2005 workshop on Digital identity management*, Fairfax, VA, USA, November 2005. ACM Press.

59. McClure, S. and Scambray, J. "Platform for Privacy Preferences strives for user control, but its sanctions lack teeth". October 1999. Available from: http://www.infoworld.com/cgi-bin/displayArchive.pl?/99/40/o03-40.58.htm [Accessed: January 12, 2006].

60. McCullagh, D. "Is TRUSTe Trustworthy?" Novmber 1999. Available from: http://www.wired.com/news/politics/0,1283,32329,00.html [Accessed: January 2006].

61. Menn, J. "Microsoft's Passport fails to travel far as Web strategy". December 31, 2004. The Seattle Times, Business and Technology. Available from: http://seattletimes.nwsource.com/html/businesstechnology/2002136272_passport31.html [Accessed: December 2005].

62. Microsoft. "Microsoft Passport Network". Microsoft Corporation, 2005. Available from: https://accountservices.passport.net/ppnetworkhome.srf?lc=1033 [Accessed: October 2005].

63. Microsoft Corporation. "Windows Live ID Service". April 2006. Microsoft MSDN. Available from: http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnlive/html/WinLiveIDServ.asp [Accessed: March 2006].

64. Microsoft Corporation and Ping Identity Corporation. "A Guide to Integrating with InfoCard v1.0". August 2005. Available from: http://download.microsoft.com/download/6/c/3/6c3c2ba2-e5f0-4fe3-be7f-c5dcb86af6de/infocard-guide-beta2-published.pdf [Accessed: February 2006].

65. Moores, T. T. and Dhillon, G., "Virtual extension: Do privacy seals in e-commerce really work?" Communications of the ACM, ACM Press December 2003. **46**(12).

66. Newman, A., "Protecting Privacy in Europe: Transgovernmental policy entrepreneurs and administrative feedbacks". Princeton University, September 16, 2005. http://www.princeton.edu/~smeunier/Newman.pdf.

67. Nozin, M., ed. *"A Privacy Framework To Provide Users With Control, Accuracy And Audit"*. 2005, Master Thesis, SITE, University of Ottawa.

68. OASIS. "eXtensible Access Control Markup Language (XACML) Version 2.0". Feb 2005. OASIS Standard. Available from: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf [Accessed: October 2005].

69. OASIS Security Services (SAML) TC. "Security Assertion Markup Language (SAML)". OASIS Standards. Available from: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security [Accessed: December 2005.

70. Office for Civil Rights. "HIPPA - Medical Privacy - National Standards to Protect the Privacy of Personal Health Information". United States Department of Health Human Services. Available from: http://www.hhs.gov/ocr/hipaa/ [Accessed: February 13, 2006].

71. Office of the Privacy Commissioner of Canada. "Privacy Legislation in Canada (Fact Sheet)". Available from: http://www.privcom.gc.ca/fs-fi/02_05_d_15_e.asp [Accessed: March 4, 2006].

72. Olsen, S. "Privacy terms revised for Microsoft Passport". April 4, 2001. CNET News.com. Available from: http://news.com.com/Privacy+terms+revised+for+Microsoft+Passport/2100-1023_3-255310.html?tag=bplst [Accessed: October 2005].

73. Pashalidis, A. and Mitchell, C. J. "A Taxonomy of Single Sign-On Systems". In *Proceedings of the Information Security and Privacy: 8th Australasian Conference, ACISP 2003*, Wollongong, Australia, July 9-11, 2003. Lecture Notes in Computer Science, Springer Berlin / Heidelberg.

74. Peyton, L. and Nozin, M. "Tracking privacy compliance in B2B networks". In *Proceedings of the 6th international Conference on Electronic Commerce*, Delft, The Netherlands, October 25 - 27, 2004. ACM Press.

75. Pfitzmann, B. "Privacy in Enterprise Identity Federation - Policies for Liberty Single Signon -". In *Proceedings of the 3rd Workshop on Privacy Enhancing Technologies (PET)*, Dresden, Germany, March 2003. Lecture Notes in Computer Science, Volume 2760, Dec 2003, Pages 189 - 204.

76. Pfitzmann, B. and Waidner, M. "Federated Identity-Management Protocols — Where User Authentication Protocols May Go". In *Proceedings of the 11th Cambridge Workshop on Security Protocols*, Cambridge, UK, April 2003. Lecture Notes in Computer Science, Volume 3364, Pages 153 - 174.

77. Pfitzmann, B. and Waidner, M. "Privacy in browser-based attribute exchange". In *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, Washington, DC, USA, November 2002. ACM Press.

78. PI. "Privacy International". Available from: http://www.privacyinternational.org/ [Accessed: May 2006].

79. PRC. "Privacy Rights Clearinghouse". Available from: http://www.privacyrights.org/ [Accessed: May 2006].

80. Reed, M., Syverson, P., and Goldschlag, D., "Anonymous Connections and Onion Routing". IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection, 1998.

81. Regan, P., "Legislating Privacy: Technology, Social Values and Public Policy". University of North Carolina Press, 1995.

82. Reiter, M. and Rubin, A., "Crowds: Anonymity for Web Transactions". ACM Transactions on Information and System Security, November 1998. **vol. 1, no. 1**: p. 66-92.

83. Saint-Andre, P. "Liberty ID-SIS Presence Service Specification". 2005. Version: 1.0-10, Liberty Alliance Project. Available from: http://www.projectliberty.org/specs/draft-liberty-id-sis-presence-v1.0-10.pdf [Accessed: December 2005].

84. SourceID. "Digital Identity Basics". Ping Identity Corporation. Available from: http://www.sourceid.org/content/primer [Accessed: December 2005].

85. Stufflebeam, W. H., Antón, A. I., He, Q., and Jain, N. "Specifying privacy policies with P3P and EPAL: lessons learned". In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*, Washington DC, USA, 2004. ACM.

86. Taylor, K. and Murty, J. "Implementing role based access control for federated information systems on the web". In *Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003 - Volume 21*, Adelaide, Australia, 2003. ACM Press.

87. Tourzan, J. and Koga, Y. "Liberty ID-WSF Web Services Framework Overview". 2006. Version: 2.0-06, Liberty Alliance Project. Available from: http://www.projectliberty.org/specs/draft-liberty-idwsf-overview-v2.0-06.pdf [Accessed: June 2006].

88. Turow, J., Feldman, L., and Meltzer, K. "Open to Exploitation: American Shoppers Online and Offline". Annenberg Public Policy Center of the University of Pennsylvania June 2005. Available from: http://www.annenbergpublicpolicycenter.org/04_info_society/Turow_APPC_Report_WEB_FINAL.pdf [Accessed: Novmber 12, 2005].

89. U.S. Department of Commerce. "Introduction to the Safe Harbor". Available from: http://www.export.gov/safeharbor/ [Accessed: March 1, 2006].

90. U.S. Federal Trade Commission. "Privacy Online: Fair Information Practices In the Electronic Marketplace". May 2000. Available from: http://www.ftc.gov/os/2000/05/testimonyprivacy.htm [Accessed: January 24, 2006].

91. Varney, C. and Hartson, H., *"Privacy and Security Best Practices"*. November 2003, Version 2.0, Liberty Alliance Project. Available from: http://www.projectliberty.org/resources/specifications.php.

92. Varney, C. and Sheckler, V., *"Deployment Guidelines for Policy Decision Makers"*. September 2005, Version 2.9, Liberty Alliance Project. Available from: http://www.projectliberty.org/about/whitepapers.php.

93. Wason, T. "Liberty ID-FF Architecture Overview". 2004. Version: 1.2-errata-v1.0, Liberty Alliance Project. Available from: http://www.projectliberty.org/resources/specifications.php [Accessed: September 2005].

94. W3C Recommendation WWW Consortium. "The Platform for Privacy Preferences 1.1 (P3P1.1) Specification". Available from: http://www.w3.org/TR/2006/WD-P3P11-20060210/Overview.html [Accessed: February 20, 2006].

95. P3P Preference Interchange Language Working Group WWW Consortium. "A P3P Preference Exchange Language 1.0 (APPEL1.0)". W3C Working Draft 15 April 2002. Available from: http://www.w3.org/TR/P3P-preferences/#P3Ppolicies [Accessed: January 25, 2006].

96. Yee, G. and Korba, L. "Semi-Automated Derivation of Personal Privacy Policies". In *Proceedings of the 2004 Information Resources Management Association International Conference*, New Orleans, Louisiana, USA, May 2004. NRC 46539.