Self-service Privacy: User-Centric Privacy for Network-Centric Identity

Jose M. Alamo, Miguel A. Monjas, Juan C. Yelmo, Beatriz San Miguel, Ruben Trapero, Antonio M. Fernandez

Self-service Privacy: User-Centric Privacy for Network-Centric Identity

Jose M. del Alamo[1], Miguel A. Monjas[2], Juan C. Yelmo[1], Beatriz San Miguel[1], Ruben Trapero[1], Antonio M. Fernandez[1]

[1]Universidad Politécnica de Madrid, Ciudad Universitaria s/n, 28040 Madrid, Spain.
{jmdela, jcyelmo, smiguel, rubentb, antoniofer}@dit.upm.es
[2]Technology and Innovation Unit, Madrid R&D Center, Ericsson.
miguel-angel.monjas@ericsson.com

Abstract. User privacy has become a hot topic within the identity management arena. However, the field still lacks comprehensive frameworks even though most identity management solutions include built-in privacy features. This study explores how best to set up a single control point for users to manage privacy policies for their personal information, which may be distributed (scattered) across a set of network-centric identity management systems. Our goal is a user-centric approach to privacy management. As the number of schemas and frameworks is very high, we chose to validate our findings with a prototype based on the Liberty Alliance architecture and protocols.

Keywords: privacy, identity management, user-centric, network-centric, user control.

1 Introduction

Being online today means so much more than just being connected to the Internet and browsing simple Web sites. Users join, interact with and enjoy social Web sites and online communities such as Facebook and MySpace, photo and video hosting Web sites such as Flickr and YouTube, streaming music programs such as Spotify and Last.fm, application stores such as the Apple App Store, online banks and e-commerce businesses such as PayPal, e-government sites, and so forth.

The process is quite simple. Users create a new account on any of these feature-rich content and service providers by filling in a registration form and providing a few personal details. Additionally, by accepting the default service privacy policy, users may allow the provider to share these details with the outside world. At the end of the process, the user is allowed to access and enjoy the provider's site.

The amount of scattered personal information begins to grow as users repeat this simple process with different providers. As a result, some undesirable problems can arise: not only the most obvious of bad user experience (users have to remember different logins and passwords), and increased risk of identity theft, but,

eventually, data privacy concerns. In this context, we understand privacy as someone's right to keep their personal information and relationships secret and thereby reveal themselves selectively.

In truth, the privacy concerns might not be a problem. Legislation usually forces providers (to different extents, depending on the jurisdiction) to handle user data in accordance with the law. Thus, service providers generally offer different mechanisms and custom tools to manage user data. However**,** the resulting heterogeneity is a huge disadvantage for users who wish to actively manage and control their personal information. As concluded by a thorough analysis of 45 social networking sites [1], "*privacy in social networks is dysfunctional in that there is significant variation in sites' privacy controls, data collection requirements, and legal privacy policies*".

In summary, users currently lack a simple mechanism to verify what personal information is available on the Web, how it is used and how they can modify, update or delete it. Our vision is that in the short run privacy awareness will rise much higher as a result of the lack of comprehensive solutions. Thus governments and individuals alike will demand simple tools to govern the use and release of their personal information. In this article, we introduce a solution that aims to address some of these problems by enabling users to manage their personal data privacy in a simple and efficient way.

The remainder of the paper is organized as follows. First, in section 2 and 3 we describe the main approaches to identity management and stress the differences between them regarding privacy control. Then, section 4 details the solution that we have implemented to enable users to manage their personal data privacy in a simple and efficient way. After that, section 5 describes the validation we have conducted. Finally, section 6 analyses the state of the art of the related work and section 7 concludes the paper.

2 Network-Centric Identity Management

Identity Management commonly refers to the processes involved in the management and selective disclosure of user-related identity information, either within an institution or between several entities, while preserving and enforcing both privacy and security requirements. Identity management systems can be classified according to several different criteria. A common taxonomy distinguishes network- from user-centric approaches. With regard to the former, the first comprehensive specifications were created by the Liberty Alliance Project [2], a business alliance established in 2001, which has been recently succeeded by the Kantara Initiative [3]. Its early work was followed by others such as SAML 2.0 (Security Assertion Markup Language) [4] and WS-Federation [5].

The Liberty approach to identity federation and identity-based web services associates service providers with trusted domains (identity networks), which are supported by Liberty technology and operative agreements through which trust relationships are defined between providers. The identity network infrastructure

supports users in transacting business with associated providers in a secure and apparently seamless environment. Each company maintains its own customer accounts, including relevant identity resources. Users can federate (link) accounts at different providers using an opaque pseudonym, which enables a single sign-on between providers and allows for the secure sharing of identity information.

Some entities may focus on managing these federations, as well as providing ancillary services:

- Identity Providers (IdP) know all users and service providers within the identity network and their affiliations. They also know how to authenticate users; thus, they can certify a user's identity to any provider. Whenever users want to access a service provider, they will be redirected to the IdP, where they will be authenticated. Once a user has successfully logged in, the IdP will send a statement back to the service provider containing information related to the identity of the user (pseudonym), information about other entities within the identity network and the credentials needed to access them.
- Discovery Services (DS) record where the users' identity resources are stored within the identity network. When a service provider wishes to find any user's information it sends a query to the DS, which returns the endpoint reference for that resource and a valid credential for one-time access.

In the Liberty context, a service provider is a Web service that acts with a certain identity resource to retrieve information about an identity, update information about an identity or perform an action for the benefit of some identity. A service provider may play the roles of both an identity-based Web Service Consumer (WSC) or an identity-based Web Service Provider (WSP). WSPs usually aggregate several identity resources into an identity profile (e.g., a personal profile or address book). The Liberty protocol that allows WSCs to access WSPs to query/update/delete personal data on behalf of a user is called the Data Service Template (DST) [6].

Within a given Liberty-based identity network, privacy is supposed to be handled by the user at each WSP (and not at the IdP/DS level). From that point on, the WSP is responsible for managing user privacy settings and preferences. The method for choosing and recording privacy settings is outside the scope of the Liberty specifications.

3 User-Centric Identity Management

Network-centric identity management approaches, such as the one proposed by the Liberty Alliance, have failed to reach the interest (and trust) of users, becoming constrained to the enterprise or governmental domains. For years, companies (especially telecommunication operators) have controlled user attributes: they know who you are, what is done with some of your identity-related information, who is using it, how and for what. Fortunately now, multiple service providers populate the Internet without being associated to any operator thus bringing the ex-

isting identity management systems towards a new scope around the concept of *user centricity*.

The term user-centric was introduced in the identity management arena in about 2006. A user-centric identity management system "*needs to support user control and considers user-centric architectural and usability aspects*" [7]. Two main principles support user-centric identity management:

- *The user is in the middle of a data transaction.* Any information that flies between any entity pass through the user as a must condition, who has the power of consenting the transaction.
- *Huge scale advantages,* since the identity provider does not need to know about every service provider, the user directly deals with them. The identity provider can simply be used to validate the information that will be sent to the service provider.

User-centric identity management principles can be implemented by different means but the most popular ones can be divided into two categories, namely: URL- and card-based systems.

- *Simple URL-based systems* are decentralized and use URLs as users' identifiers. Compatible service providers accept identifiers created by a trusted entity (the identity provider); users are free to choose any identity provider to create an identity. The security of the transaction depends on how much the user relies on her identity provider.

  This is precisely one of the drawbacks of this approach. Malicious identity providers may give a false trust feeling, allowing unaware users to gain access to malicious service providers. Once inside the service provider, the victims can be easily cheated taking advantage of a false security feeling. Some examples of URL-based systems are OpenID [8] and LID (Light-Weight Identity) [9].
- *Identity card-based systems* are built using the information card metaphor [10]. Although there are different definitions of this concept, the simplest could be that expressed by the Information Card Foundation [11]: "*Information Cards are the digital version of the cards you carry in your purse or wallet today. You use them with a new kind of digital wallet called a selector.*" Selectors are pieces of software that complement Web browsers and allow users to choose the information card that will be supplied to service providers upon access. Depending on the user identity information that the service provider requires, the user chooses a certain information card. In these systems users decide, using a single interface (the card selector), which pieces of their identity they share with a given service provider. This approach gives users excellent control over their identity resources: they can assume at least partial control over their privacy, while enjoying enhanced usability.

  Additionally, card-based systems provide user with a consistent user experience. No matter what service provider is being used or what type of identity information is being enclosed towards them. In general, users interact with the same type of identity managers for every identity transaction they do. This is an important difference with respect to the network centric identity management scenario, where each service provider provides its own user interface

which means the user is learning a new interface, sometime simply for using it only one time (for instance, at registration time). Examples of identity card based systems are Higgins [12] and Windows Cardspace [13].

However, several security and privacy issues arise in information card based systems [14]. Firstly, the identity providers are aware of the service providers to which the user attempts to log in, so malicious identity provider can learn about the behavior of the users on the Web. Secondly, and even more important, is the problem of the reliance on the user's judgment of the trustworthiness of the service provider. This means that if a service provider is not trustworthy, it could gather information about users and potentially use this in unauthorized ways.

In both URL- and card-based approaches, the user identity information sharing happens in the foreground, since the user must be online and the service provider must actively request this information. This might be seen as a major handicap since users must be online and connected with the consumer service provider. Additionally, users cannot configure default privacy preferences to infer automated decisions regarding the release of their identity information. Instead, users must always give explicit permission for the release of information on a per-request basis.

Furthermore, it is unrealistic to expect that all user-related data are reachable through a user agent. Consider, for instance, information such as cellular network-based location information. Moreover, consider identity-based services such as sending an SMS or billing the user on behalf of a third party. These are identity-based services that only a specialized identity-based service provider can offer, preferably within a trusted domain.

4 Self-service Privacy in Practice

We feel that network-centric identity management solutions are essential to manage some identity information and identity-based transactions. However, a user-centric approach to privacy enabling consistent user-experience is still required in order to provide users with adequate control of their information. Therefore we introduce a solution that merges both approaches, providing a user-centric privacy approach to a network-centric identity management solution to put the concept of Self-Service Privacy into practice.

Fig. 1 introduces a high-level view of a network-centric identity management architecture consistent with the Liberty Alliance specifications. A new entity has been added to this architecture, namely *Privacy Controller* (PC). The PC joins the identity network so that its users can federate (link) their accounts with those they have in other providers (using pseudonyms).

Using this new entity users are able to:

1. Retrieve a global view (snapshot) of their identity resources across different nodes of the identity network to understand which identity resources are stored,

where they exist and their specific values. Users are also allowed to modify and delete any data.
2. See the history of how their identity resources have been used, i.e., what entities have requested them, when, the outcome of each request, etc.
3. Govern the future use and release of all identity resources by setting privacy preferences.

A generic requirement for all scenarios is that the PC must be authenticated, on behalf of the user, against the identity network in order to gain access to participating entities. The authentication is carried out against the IdP following standard Liberty protocols (Fig. 1, flow 0) using the pseudonyms obtained during the federation process. At the end of the authentication process the PC obtains an endpoint reference and credentials to access the DS.

Apart from these interactions, Fig. 1 also shows some other information flows. There is a communication flow between the user and the PC through a Graphical User Interface (GUI). The PC shows information to users and, in response, users take certain actions and decisions, which are sent back to the PC in order to drive the aforementioned flows.

Flows labelled A, B and C correspond to standard protocols as defined by the Liberty specifications: (A) a WSP informs the DS that it is storing an identity resource for a certain user; (B) a WSC queries the DS to discover which WSP stores selected identity resources about a user and also asks the DS for credentials to access them; and (C) a WSC uses the credentials retrieved in (B) to attempt to access an identity resource stored in a WSP.

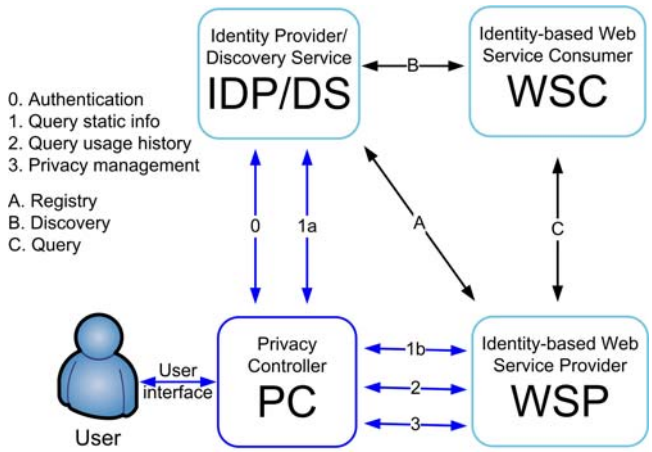

Fig. 1. Overview of the Privacy Controller.

The following sections further elaborate on each particular scenario in order to describe the functionality that allows user-centric privacy management of distributed personal information.

4.1 Providing a Snapshot of Distributed User Identity Resources

Our first sample scenario is a combination of two major flows. Firstly, users see an overall picture of the distribution of their identity resources (Fig. 2, steps 0 to 4). Through this flow, the user is able to determine which parties are hosting user resources and what kind of resources are being stored. Once the users have seen the overall set of identity resources, they can choose to manage (modify or delete) any or all of the stored data (Fig. 2, steps 5 to 11).

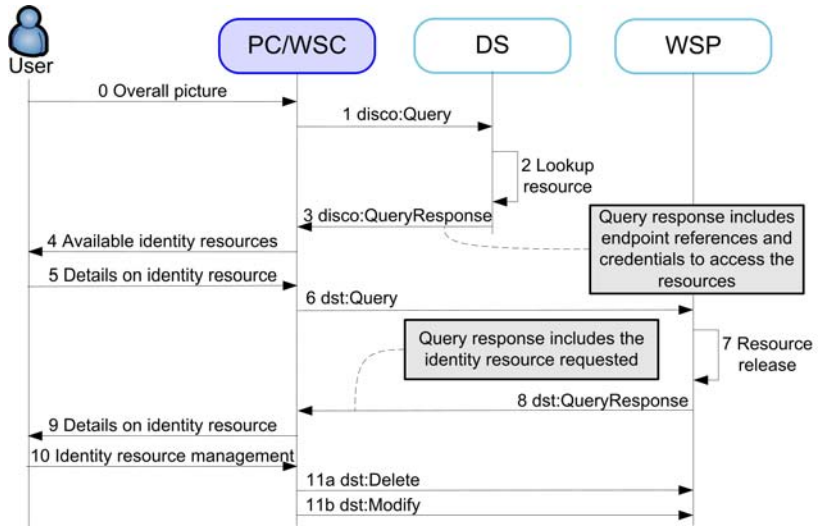

Fig. 2. Sequence diagram for the retrieval and management of identity resources.

The PC performs as a Liberty WSC to query or update a WSP using the DST protocol. The PC receives the end-point reference and the credentials needed to access the WSP each time information about identity resources is retrieved (Fig. 2, step 3).

Fig. 3 shows a screenshot of the user interface that we have implemented for the PC. It is an example of the personal data that a user has stored in two different WSPs (iProfile and myPaymentBroker). Specifically, it indicates that myPaymentBroker stores two personal resources (name and credit card) that the user can edit and modify.

Fig. 3. Screenshot of the PC providing a snapshot of distributed personal data.

4.2 Retrieving the Usage History for Identity Resources

In this scenario a user wants to know the usage history of one of their previously obtained identity resources. The PC shows the usage history for that resource with details about the resource type and value, timestamps of access and all WSCs that accessed the resource. Further information can be presented if available, such as privacy protection commitments made by the requestor, or conditions on the release of the information.

Fig. 4 shows a screenshot of the user interface that we have implemented. It shows the services that have requested access to an identity resource (credit card details stored at myPaymentBroker), for what purpose, the timestamp of the request and the outcome of the process (shown with self-explaining icons).

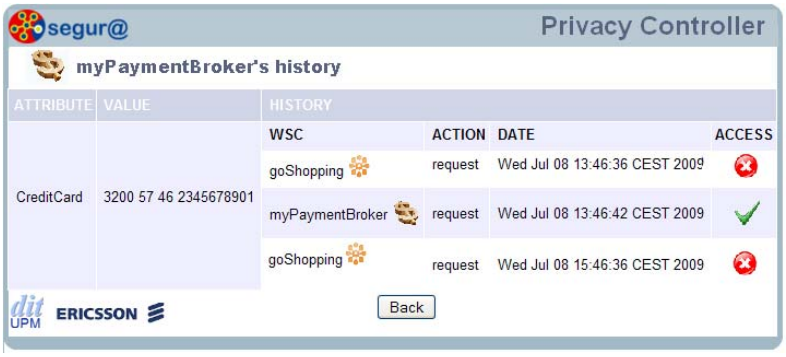

Fig. 4. Screenshot of the PC showing a usage history for a personal data

There are several approaches to gathering usage history. An approach based on proactive notifications from the WSP that stores the identity resource to the PC is not viable because it might cause scalability and performance problems in the event of large bursts of notifications. These problems can be reduced using thre-

sholds and timers to smooth out the data flow. Nevertheless, this mechanism is always WSP-initiated (push mode), preventing the PC from requesting information at will.

A better approach is to extend the Liberty identity profiles with information that registers the use of the identity resources. These records can be considered an extension of the information in a profile since they add relevant information about the retrieval of identity resources. As in the previous scenario, the PC uses the DST protocol to retrieve this information. The main benefit of this approach is that it can be PC-initiated (pull mode). This means that the information is retrieved whenever the PC requires it and is limited to PC requests. Thus, we do not foresee any scalability or performance issues. Additionally, the Liberty specifications for identity services consider extensions of the identity resources, and thus our solution can be considered Liberty-compliant.

We assume at this point that every WSP registers the request and release of the identity resources it stores. This information could be used, for example, for future audits (in fact, this may be compulsory under the laws in certain countries). The minimum information to be recorded at the WSP every time a WSC tries to access an identity resource includes:

- UserID – to identify whose information is being accessed. This is the user alias in the WSP.
- IdentityResourceType – to identify what information has been accessed. This is a subset of the information included within the identity profile.
- WSC – to identify who requested the information.
- Other information – for example, the time of access. Other relevant information that could be useful for future requirements may be the intended use of the information retrieved, whether the WSC will share it with third parties, and the promises the WSC makes about any future use of the requested information.

4.3 Enabling User-Centric Privacy Management

This scenario elaborates on the mechanisms that allow users to centrally manage their privacy preferences. The scenario is composed of four major steps:

1. The user selects a specific identity resource stored in a WSP and reviews its privacy preferences.
2. The PC retrieves and displays the privacy preferences associated with that identity resource.
3. The user modifies the privacy preferences and the PC updates them in the WSP.
4. The WSP enforces the privacy preferences whenever a requestor tries to access the identity resource.

Initial preferences might have been set by the user at the service provider's site or by the service provider itself using default values, or they can even be undefined. Privacy preferences follow a default-deny pattern: An empty set of prefe-

rences implies that all requests should be denied (other than those issued by the PC itself). Each preference added to the set grants a specific type of permission.

The PC allows users to change the policy applied to an identity resource by selecting one of a set of pre-defined policies, each of which is described in natural language. This natural language description is mapped to a specific policy described in machine-oriented privacy policy expression language. Policies are hierarchical so that it is easier for users to compare them and choose the one that best meets their needs. This approach benefits from simplicity and usability because users do not have to deal with technical policy details.

Additionally, the PC allows users to define specific options for the use and release of each data element from their profile (Fig. 5). Although this approach provides greater flexibility, it also poses some usability risks, since only advanced users understand (and probably want to know) the detailed meaning of each available policy. Therefore, this is offered only as an advanced option.

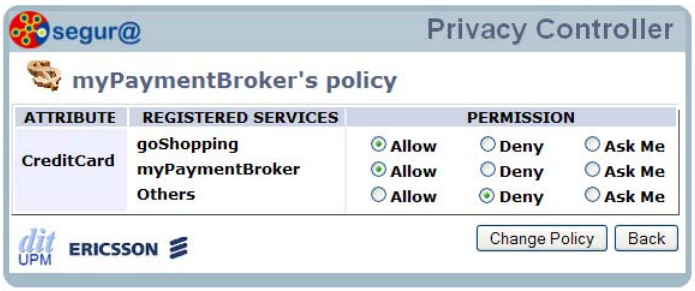

Fig. 5. Screenshot of the PC to control privacy preferences

The variables that define a custom privacy preference include the identity resource, the requestor of the identity resource, the operation requested, the permission level chosen by the user and the resource owner identifier. The identity resource values are constrained to those defined within the identity profile provided by the WSP. The requestor can be any WSC from within the identity network. The operation values might be those defined in a DST protocol (i.e., query, create, delete, modify and subscribe). The permission can be set to grant, deny or askMe, when the user prefers to decide on a per–request basis.

Since the number of rule combinations will increase exponentially (number of requestors multiplied by number of identity resources), we allow for simpler options such as *allow anyone to query this specific identity resource* or *allow just this WSC to update any identity resource*. Therefore, options such as *all* or *just one* are supported.

The solution presented in this article does not impose any specific privacy preference language. However, we have evaluated several different alternatives in terms of the following requirements. On one hand, we must allow users to easily describe their preferences and on the other hand, we have to translate these preferences into policies that a Web service provider can enforce. For our prototype we have chosen the Extensible Access Control Markup Language (XACML) [15]

since it allows for easy governance of information once the policy has been defined. However, it poses certain usability difficulties in terms of limiting how a user can express this information. Previous paragraphs have described the measures that we have taken to address these problems.

User-generated privacy preferences (and their expression in the form of privacy policies) govern user identity resources. Thus, privacy policies must be associated with the identity of a user or, more specifically, with the identity resources that they govern. Our solution realizes this association by extending the generic identity profile as defined by Liberty with a privacy policy description schema. Thus, the schema is defined once and can be added to any existing Liberty-based identity profile. Additionally, this approach allows the PC, using the DST protocol, to set (Fig. 6, steps 0 to 3) and query (Fig. 6, steps 4 to 8) the privacy policies associated with a given identity resource.

To set a privacy policy, the PC must play the role of a WSC and include the privacy policy as part of the *Create* or *Modify* element that is sent to the WSP to store the identity resource and its associated privacy policy. This mechanism does not require any changes in current DST protocols, as the policy is transmitted in the same way as any other information, namely, in the body of the message. On arrival, the WSP retrieves the policy and stores it in a policy repository.

The PC retrieves privacy policies that are associated with an identity resource using the DST protocol as well. In this case, however, the PC uses the Query operation. Once the policy has been retrieved, it is translated back into privacy preferences so that the user can understand it.

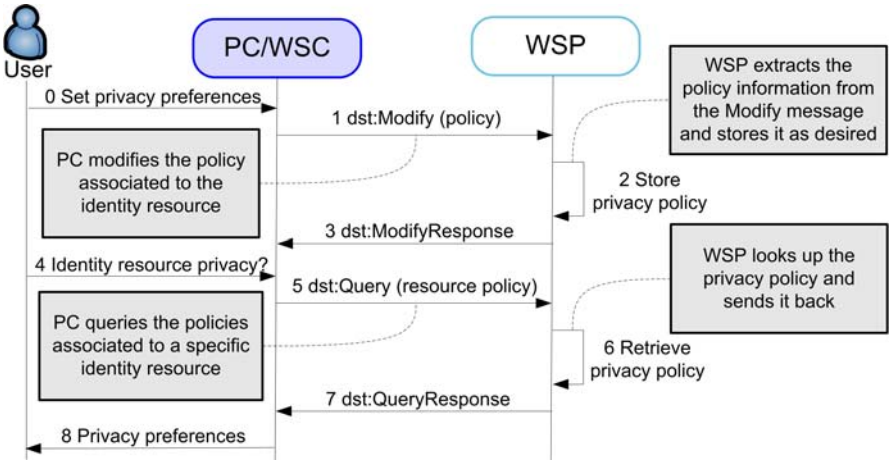

Fig. 6. The Privacy Controller sets and queries privacy policies.

Finally, privacy policies are enforced whenever a request for an identity resource is received by a WSP. To meet this requirement we have introduced three functions in the WSP (Fig. 7): Policy Enforcement Point (PEP), Policy Decision Point (PDP) and Policy Repository (PR).

The PEP catches the incoming identity resource request and retrieves information about the requesting entity (requestor WSC), the requested identity resource, the operation and the user to whom it refers. Then the PEP creates a request for the PDP with the information retrieved. When the PDP receives the request it retrieves the applicable policies from the PR and compares the request and the policies before sending a decision back to the PEP. When the PEP receives the decision it can allow the requested resource retrieval, deny the operation or pause the operation until further actions have been completed (e.g., asking the resource owner for permission). The last step can be implemented using a Liberty Interaction Service [16].

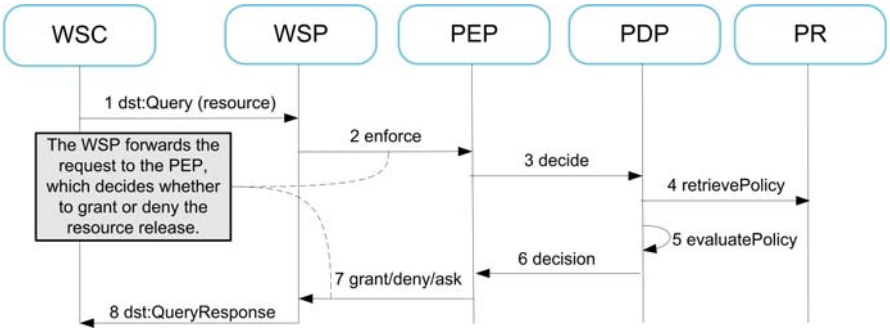

Fig. 7. Sequence diagram for privacy enforcement.

5 Validation

To validate our approach we have developed an identity network made up of five entities (Fig. 8):

- A standard IDP/DS that provides the basic infrastructure as defined by the Liberty Alliance architecture and protocols.
- Two modified identity-based WSPs. The first one, namely iProfile, is a personal profile containing postal address information. The second one, myPaymentBroker, is an identity-based payment service that stores credit card numbers. These WSPs have been modified to incorporate our contributions.
- A standard identity-based WSC; an online shop called goShopping. Once a user selects a product and decides to buy it goShopping queries myPaymentBroker to retrieve the credit card number to charge the user and, if successful, then queries iProfile to retrieve the delivery details.
- A Privacy Controller, which allows users to centrally govern their identity information in the identity network, to trace the use of the information, and to set privacy policies for its future governance.

Our solution does not introduce any impact on standard-based IDP/DS or WSCs implementations. Regarding WSPs, they must enforce privacy policies and

log transactions related to the release of personal information, which anyway may be compulsory according to privacy protection laws in some countries. Additionally, an extension to identity data services is needed, which has been done following Liberty recommendations. The extension defines a container for privacy policies governing the use of the information contained within the profile. This extension must be included in any data service supporting privacy policy management.

Liberty-based entities were developed using the open source access management and federation server platform and libraries Open Web SSO (OpenSSO) [17] by Sun Microsystems. With regard to the management and enforcement of privacy policies, we have used an open source implementation of XACML [18], also by Sun Microsystems. This implementation provides support for creating privacy policies in the PC and managing and evaluating requests against policies in the WSPs.

The PC and the WSC were deployed on a Tomcat server, while the IDP/DS and the WSPs were deployed in a Glassfish Server running the OpenSSO platform.

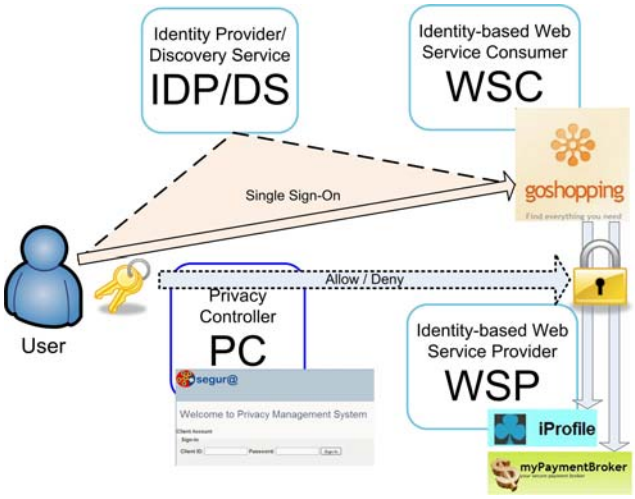

Fig. 8. Demonstration scenario for the prototype validation.

In our demonstration scenario, a user firstly signs in the PC to obtain a snapshot of her information distributed in the identity network (Fig. 3), that is, the personal information stored in iProfile and myPaymentBroker. From the PC interface, the user can also modify the values stored.

Additionally, the user can also review the history of use of any piece of this information. For example, Fig. 4 shows the screenshot that the user sees when she chooses to review the history of use of her credit card details stored at myPaymentBroker.

Since goShopping has been denied access to the credit card details, the user is not able to buy at this online shop. Thus, using the PC interface, the user modifies

her privacy preferences. Fig. 5 shows the screenshot that the PC presents to set new privacy preferences regarding her credit card details. Once the new preferences are set, the user is able to go back to goShopping and successfully carry out the purchase.

6 Related work

The approach to identity management described by Liberty Alliance specifications is a solution for the problem of how to manage shared identity resources in a network-centric identity management system. However, this solution does not allow for user-centric management of the scattered users' identity resources and privacy settings in the identity network. The Liberty approach forces users to know where every identity resource is stored and to set privacy preferences for every single node of the identity network. Moreover, it cannot possibly provide overall information about the flows of identity resources among entities since each entity only knows about others to which it has released information or from which it has retrieved information.

Recently, Liberty has announced the release of the Identity Governance Framework (IGF) [19], which aims to describe detailed privacy constraints and the mechanisms by which entities within an identity network can interact. IGF privacy constraints describe fundamental restrictions regarding the propagation, usage, retention, storage and display of identity data for entities involved in consuming and providing them. Unfortunately, the IGF is enterprise-oriented and thus does not allow for user-centric privacy management.

A promising initiative for user-centric privacy management is the PRIME Console [20], which is an interface to the user's identity management system. It allows users to create partial identities (pseudonyms) and associate personal data with each of these identities, assists the user in understanding privacy policies, makes decisions on the basis of the user's preferences and allows users to inspect the transaction history for their personal data. However, sometimes identity information is not initially disclosed by users. For example, in a network-centric identity system some entities can automatically retrieve and offer identity information regarding a user, such as geolocation. Therefore, the PRIME Console is not a comprehensive solution as it focuses only on user-centric identity systems and fails to support network-centric ones.

The HP Virtual Identity and Profile Broker (VIP Broker) [21] provides a single, centralised point of access to distributed identity data, allowing users to monitor relevant information flows. With the VIP Broker, users can also configure release policies, which will be centrally enforced whenever a request for personal information is received. Unlike the VIP Broker, our solution distributes the policies to the custodians of the information. Thus, privacy preferences are applied wherever the information is stored, avoiding the need for an intermediate broker. This is convenient from a business standpoint, since service providers usually want to retain control of the distribution of any information they own. Recently, Google has

announced the release of the Google Dashboard [22]. The dashboard lists some of the information associated with the Google services the user has subscribed and provides links to control their personal settings. However, unlike our Privacy Controller, Google Dashboard redirects the user to every service provider in order to modify the privacy preferences. Therefore, users are not provided with a comprehensive tool from where to centrally define their privacy preferences. In addition, users are not able to trace how their information has been used, which is one of the contributions of our proposal.

7 Conclusions

In this article we have presented a comprehensive solution, the Privacy Controller, to help users to actively control the privacy of their personal information while using services in the digital world. The Privacy Controller leverages various identity management technologies (consistent with the Liberty Alliance specifications) to help users understand how much personal information about them is being stored, who has accessed it and how it will be safeguarded in the future.

The lack of simple tools and inadequate privacy-awareness among users are two major obstacles that prevent the involvement of users in privacy control. Therefore, enhanced usability and better default settings play a fundamental role. We have described different measures that our Privacy Controller introduces to support users in defining their own privacy preferences. Nevertheless, we feel that there is room for improvement in this area and thus we are evaluating user modelling techniques that allow the automatic generation of user privacy preferences.

For our solution to be practical, collaboration and coordination with service providers is a must: Service providers would have to provide new interfaces and capabilities to allow external control of personal data. We envision that in the near term governments will force providers to declare the personal information they store and to provide standards-based mechanisms to interact with it. As a matter of fact, in Europe recent privacy protection initiatives [23] have introduced legislative principles such as '*privacy by design*' and '*accountability*'.

In the future we hope to extend the Privacy Controller to other identity networks such as social networks, which are huge and often unregulated sources of personal information. We feel that such Web sites urgently need mechanisms to provide semantic interoperability between different identity networks. The use of ontologies as information mediators might be a promising future direction.

Acknowledgments

This work has been partially supported by CDTI Ministry of Science and Innovation of Spain, as part of the SEGUR@ project (https://www.cenitsegura.es/), under the CENIT program, CENIT-2007/2004.

References

1. Bonneau, J. and Preibusch, S.: The Privacy Jungle: On the Market for Data Protection in Social Networks. In the 8th Workshop on the Economics of Information Security (WEIS 2009), (2009)
2. Liberty Alliance Project, http://www.projectliberty.org
3. Kantara Initiative, http://www.kantarainitiative.org
4. Cantor, S. et al.: Assertions and protocols for the OASIS Security Assertion Markup Language (SAML) Version 2.0. OASIS Standard. OASIS Security Services TC (2005)
5. Goodner M. and Nadalin, A. (eds.): Web Services Federation Language (WS-Federation) Version 1.2. OASIS Standard. OASIS Web Services Federation (WSFED) TC (2009)
6. Kainulainen, J. and Ranganathan, A. (eds.): Liberty ID-WSF Data Services Template Specification, Version 2.1. Liberty Alliance Project (2006)
7. Bhargav-Spantzely, A., Camenisch, J., Gross, T. and Sommer, D.: User centricity: A taxonomy and open issues. In the Second ACM Workshop on Digital Identity Management (DIM'06), pp. 493-527. IOS Press, Amsterdam (2007)
8. OpenID Web site, http://openid.net/
9. Light-Weight Identity Web site, http://lid.netmesh.org/wiki/Main_Page
10. Jones, M. and McIntosh, M. (eds.): Identity Metasystem Interoperability Version 1.0. OASIS Standard. Identity Metasystem Interoperability (IMI) TC (2009)
11. Information Card Foundation, http://informationcard.net
12. Higgins, http://www.eclipse.org/higgins/
13. Windows Cardspace, http://www.microsoft.com/windows/products/winfamily/cardspace
14. Alrodhan, W.A. and Mitchell, C.J.: Addressing privacy issues in Cardspace. In the Third International Symposium on Information Assurance and Security, pp 285-291. IEEE Computer Society, Washington (2007)
15. Moses, T. (ed.): Extensible Access Control Markup Language (XACML), Version 2.0. OASIS Standard. OASIS eXtensible Access Control Markup Language (XACML) TC (2005)
16. Aarts, R. and Madsen, P. (eds): Liberty Id-WSF Interaction Service Specification, Version 2.0-errata-v1.0. Liberty Alliance Project (2007)
17. OpenSSO, https://opensso.dev.java.net/18.
18. Sun's XACML Implementation, http://sunxacml.sourceforge.net/
19. Madsen, P. (ed.): Liberty IGF Privacy Constraints Specification, Version 1.0. Liberty Alliance Project (2009)
20. Leenes, R., Schallaböck, J. and Hansen, M.: PRIME White Paper, Version 3. PRIME Project, (2008)
21. Hewlett-Packard Development Company: HP Virtual Identity and Profile Broker. Hewlett-Packard (2007)
22. Google Dashboard, http://www.google.com/dashboard/
23. Article 29 of the Data Protection Working Party, The Future of Privacy - Joint contribution to the Consultation of the European Commission on the legal framework for the fundamental right to protection of personal data, 02356/09/EN, 01 Dec 2009.