A Context-aware Decentralized Identity Platform for the Social Web

Andrei Vlad Sambra
TELECOM SudParis
CNRS Samovar UMR 5157
9 rue Charles Fourier, 91011 Evry, France.
andrei.sambra@it-sudparis.eu

Maryline Laurent
TELECOM SudParis
CNRS Samovar UMR 5157
9 rue Charles Fourier, 91011 Evry, France.
maryline.laurent@it-sudparis.eu

ABSTRACT
This paper presents a context-aware and decentralized identity platform, which in turn can be used to create social networks or collaboration platforms. Its originality lies in providing an increased privacy and control over a user's online identity, user group management, resource ownership and content sharing.

This paper addresses the shortcomings of current identity and resource management systems, especially the lack of context in which data sharing takes place on the Internet. Moreover, it discusses the advantages for users to have a decentralized resource management system, while at the same time remaining in control of the data they share, as well as the device on which it is stored.

Keywords
online identity, group management, context-aware Web services, privacy, linked data, Semantic Web, WebID

1. INTRODUCTION
Over the past decade, we have witnessed a dramatic increase in the number of web services offering social interactions between users. These services come in different forms and shapes, such as social networks, content management systems (CMS), software forges, blogging tools, or collaboration services in general. One noticeable aspect is that most web services manage people and groups as simple lists of users. For this matter and to the detriment of users, privacy is often found as an additional feature and it is not implemented by design. While it is true that some people join public communities in order to find new friends who share common interests, others would simply want to have more control over their privacy.

One may argue that better privacy policies may reduce the risk of exposure. However, even if users decide to protect their public data or even remove their accounts, there is no guarantee that the process is instant and permanent, since most of the countries have voted laws requiring that online services store user data for several months up to one year or even longer.

Another missing aspect of resource management in online communities reflects on the lack of human perception, to which we will now refer to as *context*. By context, we mean particular situations or relationships in which the user can be found. For example, an employee may not have a similar relationship with all his/her co-workers, and thus he/she does not share the same *perspective* on that group of people. Just because one member has the same relationship with the rest of the group does not necessarily mean that the reciprocal is also true. If one person does not view someone as a trusted member, it will not share information with him/her. It is important that each group member should be allowed to create his/her own representation of the same group, based on the *personal context* that is appropriate to them.

This paper proposes an innovative decentralized identity platform, focused on user privacy and data control. First we provide a description of current classic identity management systems. Next, we present the Semantic Web [2] and WebID [12] as enablers for our solution. Furthermore we describe our proposal, followed by current challenges and conclusions.

2. RELATED WORKS
Up to now, identity and resource management have been left at the discretion of individual services. There have been no official proposals for unified solutions or standards ensuring context-based optimal management of users, groups and resources. We have currently noticed two major trends in account management. The first one is the so-called *account silo* effect, where resources are not managed across domains and applications, and therefore there is no need for interoperability. The second one involves *federated* systems, where cross-domain authentication and user account management are required. However, in the latter case, user credentials cannot be used outside the scope of that particular federation.

2.1 Account silos
In the case of "silos", support of particular services usually leads to creating dedicated local accounts for each user, tying and limiting the user to a particular service and/or resource. Furthermore, users have no control over how their

personal account data are used by the service. For example, private data collected from users can be sent to third party companies for advertising purposes.

Another important issue deals with authentication and identification. Many services authenticate users based on username and password combinations. In that respect, federated and single sign-on services like OpenID [11] have proven to be quite useful. However, implementing a cross-domain authentication system does not only require huge efforts from large entities for making everything compatible, but also powerful trust relationships. In addition, once the authentication has been performed, services still require that users have local accounts.

Users and groups are usually managed on site, using standalone systems. The rationale behind such systems is that companies have better control over user actions while allegedly offering better security. Companies and in general large online businesses thrive on data mining their users for advertising purposes. In most of the cases they offer "free" sign-up for their services and provide people with numerous attractive features, encouraging them to provide more personal data. In these cases, the users are not the customers but mere products offered to the real customers, i.e. third party advertising companies.

2.2 Federated Identity Management

Federated Identity Management (FIM) is a system that enables companies with several different technologies, standards and use-cases to share their applications by allowing individuals to use the same login credentials or other personal identification information across security domains. FIMs were introduced as a potential solution to centralized systems. Implicit in this definition is trust. The fact that various providers have formed a circle of trust among themselves means that there must exist a certain level of trust, sufficient enough to be willing to exchange messages between companies. When these messages contain the authentication and authorization credentials of users, allowing users from one company to access resources in a federated system, we obtain a federated identity management system. A direct advantage of FIM is the Single Sign On (SSO) capability, allowing users to move from one service provider (SP) to another with no need for additional authentication.

The first FIM protocol was Microsoft Passport, a proprietary closed-source system [9, 8]. The current and most widely used open standard in use is SAML [10] followed by OpenID [11]. A more elaborated version of SAML is Shibboleth [6], which is getting increasingly popular. There are also enterprise-level solutions, like IBM and Microsoft's WS-Federation [1], as well as WebSEAL [7].

For all FIM solutions, note that the resource site or service provider needs to identify the user's identity provider (IdP) in order to redirect the user to the appropriate authentication service. This is known as the *discovery problem*. A second drawback is that the SP has to be sure of the authenticity of the returned authentication statement and the fact that it identifies the current user. Still, addressing these two problems usually involves having to use globally unique IDs based on DNS names (i.e. email in the case of OpenID) or to pre-configure one or a small number of IdPs into the SP and force the user to use one of them.

2.3 Positioning our works

Due to the decentralized and user-centric nature of our solution, the problems appearing in centralized and FIM systems are resolved.

As opposed to silo accounts, in our solution, user accounts are directly managed by their owners. Also, users only need to manage a single profile, as opposed to multiple profiles, each located on a different website (e.g. Facebook, Google+, MySpace, etc.). Users have access to fine-grained privacy policies regarding who can access their resources as well as under which circumstances. However, based on different access policies, users can benefit from the same advantages as if using multiple profiles (e.g. family, friends, e-commerce, etc.). We are also considering adding the option of having multiple distinct profiles, linked to the same resources (i.e. pictures, videos, blog posts, etc.).

Even though there exist certain similarities between our proposal and project Diaspora*[1] (i.e. both decentralized and user-hosted), our solution serves as an identity platform on which additional services can be enabled, while the latter only serves as a distributed social network.

User-centricity also means that federated systems are no longer required. Each time someone requires information, he/she only needs to query the user's profile (namely the user's personal space) instead of a local database belonging to a distinct service provider or querying a FIM. Furthermore, users control both the data they allow access to, and more importantly the device where data reside. The following section describes several technologies helping to achieve this control.

3. KEY CONCEPTS

An alternative to "silos" comes in the form of personal user spaces, based on Linked Data [3]. These spaces would at least contain a user profile, built within the *Semantic Web*, and additionally offering different authentication options.

3.1 Semantic Web

The term Semantic Web was first defined by Tim Berners-Lee and it refers to the web of data [2]. The Semantic Web should be considered in some ways like a global database, or better yet an information space. Its goal is to be useful not only for human-to-human communication, but also that machines can be able to participate and help.

The most important obstacle leading to mass adoption of the Semantic Web is that most of the information on the Web is designed for human readers. For example, if we use a database with well defined meanings for its columns, the structure of the data would not be evident to a robot browsing the web. For this reason, the Semantic Web provides languages for expressing information in a machine processable form. The most common data models used by the Semantic Web are the Resource Description Framework[2] (RDF)

[1]https://joindiaspora.com/
[2]http://www.w3.org/RDF/

and N3[3]. They are based upon the idea of making statements about resources (in particular Web resources) in the form of subject-predicate-object expressions. These expressions are known as triples.

The Semantic Web (as a component of the coming Web 3.0) allows applications to communicate between one another without having to rely on application programming interfaces (APIs). This means that data will be easily portable, thus easily enabling cross-domain applications and services. In the Semantic Web, data are structured in ontologies. An ontology formally represents knowledge as a set of concepts within a domain, and the relationships between those concepts. More information on Semantic Web ontologies will be provided in the following sections. Among them, the most relevant for our proposal is the Friend-of-a-Friend ontology.

3.2 Friend-of-a-Friend

RDF has been extended using the Friend-of-a-Friend (FOAF) [5] vocabulary to allow the Semantic Web community to define an open-data social graph. This ontology defines links between people, a description of them and their properties using RDF. In this model, a uniform resource identifier (URI) refers to FOAF data representing a person, a group, or their agents and their respective relations.

FOAF collects a variety of terms; some describe people, some groups, some documents. Different kinds of applications can use or ignore different parts of FOAF.

FOAF descriptions are themselves published as linked documents in the Web (e.g. using RDF/XML, N3, etc.). The result of the FOAF profile is a network of documents describing a network of people and properties. Each FOAF document is itself an encoding of a descriptive network structure. Although these documents do not always agree or tell the truth, they have the useful characteristic that they can be easily merged, allowing partial and decentralized descriptions to be combined in interesting ways.

3.3 WebID

One of the most important aspects of our proposal deals with authentication in the Semantic Web. A very interesting solution we decided to adopt comes in the form of WebID [12], an authentication system based on FOAF and TLS. We have chosen WebID because it helps alleviate the difficulty of remembering different logins and passwords combinations that users face when authenticating on multiple websites.

WebID's simplifications create a cascade of benefits. Being a Web Architecture compliant protocol, trust can be moved from the Identity Provider to the Web of relations. This approach would in fact address the issues present in federated identity management systems, described in Section 2.2.

Please consider the following example of WebID-based authentication process, described in Figure 1. The three key elements in this example are *User 1* (i.e. the user's browser), *User 2* (i.e. a friend's personal user space to which he/she wants to authenticate), and finally the *IdP* (i.e. User 1's Freedombox[4]). The authentication process starts as soon as User 2 demands a client certificate from User 1, denoted by (1). Next, User 1 replies by selecting a browser certificate (2). Please note that users can have multiple certificates, and based on specific privacy levels, they can refer to different profiles (e.g. family, friends, e-commerce, etc.). In this case, the User 1 selects a certificate matching his/her relation to User 2.

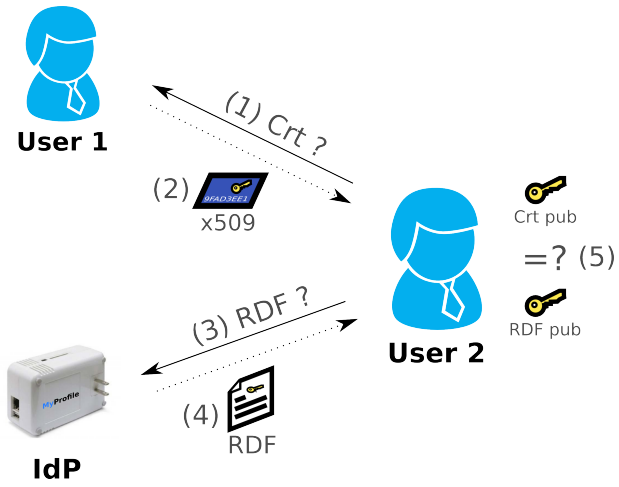

Figure 1: WebID-based authentication.

Next, User 2 queries User 1's IdP (3) for an RDF file containing his/her profile, based on the URI found inside the certificate's *SubjectAlternativeName*. The returned RDF document (4) contains several user profile elements, the most relevant being the identity URI and the public key elements (i.e. modulus and exponent). Next, User 2 compares the identity and public key elements found in the certificate to the identity and public key elements found in the retrieved profile (5). If a match is found, User 2 successfully authenticates the User 1, and based on privacy policies corresponding to User 2, additional profile data can be made available by User 1.

Next, we would like to present several reasons which helped us to choose WebID.

3.3.1 *Comparison with OpenID, BrowserID and Federated SSO*

WebID was inspired by OpenID but improved in a number of meaningful ways. OpenID solves account multiplication issue by allowing users to login to different sites using the same global identifier. By comparison, the WebID protocol is much simpler, requiring only one additional connection over the initial HTTP request connection for the resource. Furthermore, WebID offers user-interaction simplicity. OpenID requires the user to remember and type an OpenID URL, while WebID hides the URL in the X.509 certificate allowing the browser to offer select-and-click interaction. This is very helpful especially on cell phones and small devices.

[3]http://www.w3.org/DesignIssues/Notation3

[4]http://wiki.debian.org/FreedomBox

BrowserID[5] is very similar to OpenID, both using email addresses as identifiers. As opposed to OpenID, BrowserID offers a few subtle advantages, the most important being that it does not involve the identity provider (i.e. the email provider) in the login process, thus increasing privacy. However, both OpenID and BrowserID only concern the authentication process, without being able to provide additional data about the user. In the end, the user still needs to create a local profile on each website to which it authenticates.

To conclude, WebID allows users to authenticate securely to any website in the world, without the need to fill out any new forms, whilst giving that site conditional access to the user's profile data.

4. CONTEXT-AWARE DECENTRALIZED IDENTITY PLATFORM

What we propose redefines the term "identity platform", adding additional layers on top of a simple identity provider. Our solution builds a real graph of a user's identity, starting from the possibility to create and manage a personal profile, to adding items of interest as well as media files (e.g. pictures, videos, etc.). Additionally, our proposal can be used as a communication tool in the process of creating user groups, as well as sharing resources based on specific contexts. Section 4.3 describes the complete process.

To better understand how our proposal works, we decided to present its functionalities as stand-alone features.

4.1 Personal profile

Based on Linked Data and the Semantic Web, personal profiles can easily be created. Here, the profile is a collection of user attributes described using the FOAF ontology. Through WebID, the profile can be extended to provide authentication by including at least one public key belonging to an X.509 browser certificate. A simple representation of a profile is provided in Figure 2.

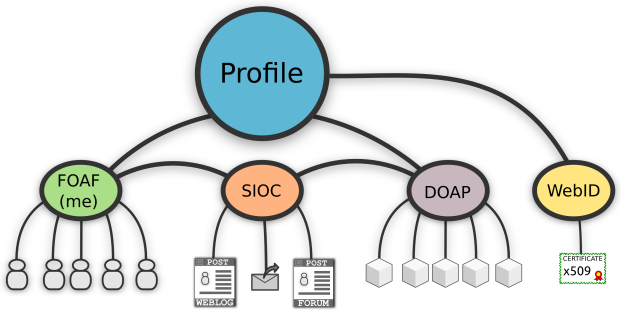

Figure 2: A typical profile document.

Depending on the user's social interactions on the Web, the profile can also contain resources like images, videos, blog and forum posts, or even mailing list messages, all being described using FOAF and the Semantically-Interlinked Online Communities (SIOC) [4] ontology. Other ontologies like the Description of a Project[6] (DOAP) can be used to describe project data belonging to a user. For example, each time a user posts a comment on a forum or a blog post, the contents of that particular comment is hosted on the user's device. However, this practice can lead to an extensive increase in bandwidth utilization. Section 5.1 goes into detail on this problem.

4.2 Notification system

Based on the Semantic Pingback [13] protocol, a significant component of the proposed system is the way it manages *notifications*. The so-called notifications are messages stored on the sender's personal user space, containing a text message as well as additional elements (e.g. sharing a photo with a caption text). In this case, as opposed to normal notification systems, the other users must subscribe to a notification feed. This step is done as part of the process of befriending a user. Next, each time a user pushes a new notification message, it gets published on a feed specific to the context to which it belongs (similar to news categories for RSS/Atom feeds). If subscribers exist for that specific feed, and if they are authorized to access it, they can then see the new message. For example, one can publish a photo and make it available only within his/her "family" context.

With this system, only useful traffic is going through the network. An additional advantage is that the user sending notifications must host the messages on his/her personal user space. As a consequence, users gain in terms of data control, but they might lose device resources (disk space).

As with most messaging protocols, spam is an important factor. When using our solution, even if senders can afford the cost of hosting a huge number of messages, the recipients for these messages do not necessarily need to receive it (as it currently happens for email). In fact, no unsolicited messages will ever be received, since users first need to subscribe to feeds in order to receive notifications, effectively eliminating spam.

4.3 Group creation process

Managing connections between people on the Web leads to an increasing demand for efficient organization structures. Creating a standard group of users can sometimes become problematic, since not all users share the same relationship with the rest of the members. Therefore, a context-aware system must be used to address the issues which arise when managing groups.

Please take for example Figure 3. The only reasonable assumption we make here is that users already know eachother, and they can be found in eachother's list of "known" friends. To begin, we shall consider *User 1* as the process initiator. The first step towards the creation of a decentralized group is to create his/her local *view* of the group, including a list of group members – *User 2* and respectively *User 3*. Once the list has been created with a local name, a unique group identifier (GID 1) is assigned to the group. *User 1* now sends the list of users as well as the GID and a short group description (e.g "Work colleagues") to each member he/she previously added. This step is denoted in Figure 3 as (1).

[5]https://browserid.org/

[6]http://trac.usefulinc.com/doap

It should be noted that the local name of the group (e.g. "Work colleagues I like") is not being communicated to the other members since this information is part of *User 1*'s local context, which means it is information that is pertinent only to this specific user.

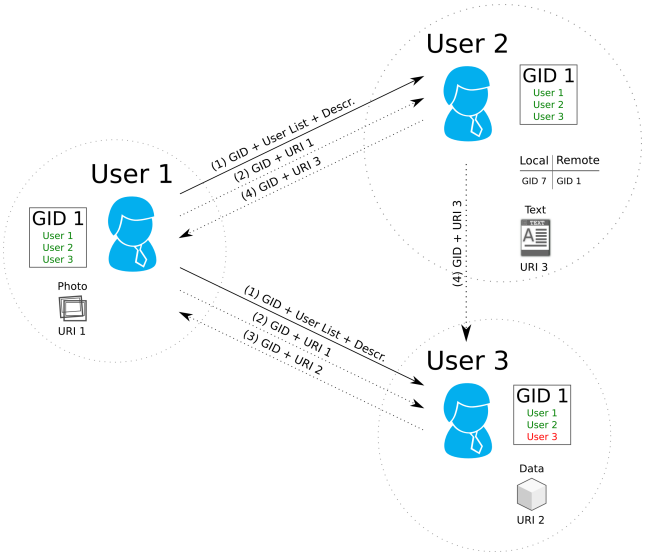

Figure 3: Group management and resource sharing.

Next, each member is informed through a notification, that *User 1* has recently included them into a new group, while at the same time inviting them to join that specific group. They now have the choice of accepting or rejecting the invitation. Please note that there is no need for *User 2* and *User 3* to provide any acknowledgement to *User 1*.

For instance, assume *User 2* accepts the invitation. The GID and user list can now be stored under a local group name. While attempting to do so however, it is possible that this action can trigger a local duplicate alert for GID 1, meaning that *User 2* is already member of a different group with the same GID value. This issue can be easily addressed by using a local table, matching local GIDs (already in use) to remote ones. Here, the remote GID value received (e.g. GID 1) will be considered as GID 7 for future references. Next, *User 2* uses the list of group participants proposed by *User 1*, since he/she shares the same relationship with the rest of the group as *User 1* does.

In the case of *User 3* who does not have a good relationship with *User 2*, he/she would like to have this user removed from his/her local list. To do so, a new local list is created, using the same GID but omitting *User 2*. Since no GID duplicate was found, the modified group can now be saved under a local name.

4.4 Sharing resources

Once the group has been created, users can immediately start sharing resources. For instance, *User 1* wants to share a photo with this specific group (Figure 3). To do so, he/she first assigns the resource to be part of this group, so that only its members can gain access to it. Next, *User 1* publishes the URI of the photo to the members of the GID to which it belongs. Group members can now access and view the resource by dereferencing the URI and then authenticating themselves to *User 1*'s system.

The physical location of the photo is on the owner's device, so if at any point *User 1* decides to stop sharing a particular resource, he/she could simply remove it from the list of resources belonging to a specific GID. The most important aspect of this system is that resource owners remain in control of the data they share, also avoiding data duplication by the group management system.

4.5 Enabling other services

The proposed solution is far from being *yet another identity provider*. When building their personal profiles, users can also provide a list of interests, which can then be used to build and offer personalized recommendation services. This feature is very important, as it allows the profile data to be used on other websites. For example, someone is looking to buy a book on an e-commerce website and he/she spends a week browsing through different categories, adding titles to a personal list of favorites and clicking "like" buttons. At the end, all this information is added to that person's personal user space. The next time he/she wants to buy a book, perhaps using a different website, the website will ask for permission to access the user's list of favorites. Since there is only one list and this list is always kept up to date, the website is certain that it has access to the user's latest personal preferences. Having access to *fresh* information can be a powerful incentive for online companies, thus eliminating the need for profile tracking.

To conclude, probably the most important advantage is that all modifications performed on a profile are instantly available to everyone requesting data, with the owner's permission, of course.

5. THE PLATFORM

Work is already in progress for a prototype platform, which will soon be released as open source software under a GNU license. The platform currently supports a handful of features, among which are the possibility to create an online identity (i.e. a profile) and to use it for authenticating to WebID-enabled websites. A minimal notification system is also available, allowing users to send short messages to each other, as well as to inform others when a user adds them to their list of people they know.

5.1 Challenges

Since our solution is under active development, certain aspects still need to be addressed. We are currently facing several major dilemmas.

The first one is the ability to import and export the complete structure of a user's identity. This is exceptionally challenging since all privacy policies enforced by the user must be preserved.

The second one deals with what happens when users run out of storage space on limited devices. We are currently investigating the possibility of using a distributed file system.

The third one is about keeping control over data ownership after resources are shared. A solution can come in the form of resource attributes, similar to how software licenses describe what usages are allowed for a particular piece of code.

5.1.1 Importing / exporting identities

Nobody likes being forced to use one identity solution over the other, meaning that users must always be allowed to choose their favorite platform. Also, sometimes projects are no longer maintained, forcing people to look for alternatives. In these cases, it is imperative that users have the means to import or export their data. Even if most services already provide user data in common formats like CSV or XLS, there is no way to preserve the privacy policies set in place by the user. We believe that only by using the Semantic Web can a true graph of a user's identity be preserved across platforms.

5.1.2 Storage space

Certain users like sharing incredible amounts of data (e.g. photos, videos, etc.), adding up to hundreds of gigabytes. These users face multiple dilemmas. For example, they have to decide on using efficient, low-power communication devices like plug computers or smartphones, or using expensive servers. This trade-off also involves certain privacy issues, as small devices are found directly under the user's physical control (e.g. Freedombox[7]), while large servers are usually located in data centers, out of the user's physical reach. Conversely, some users prefer hosting their data into data centers, as they don't have to worry about uptime, accessibility or backups. However, users should have in mind how important physical control of devices is before making their decision.

By taking advantage of the Semantic Web and public key cryptography, users can build a web of trust that can also be used for storing data remotely, on devices belonging to trusted friends. We are investigating the possibility of using this web of trust as a form of cloud storage.

5.1.3 Resource attributes

Resource attributes can define how users interact with different resources. For example, such attributes can be used to prevent "re-posting" a resource by a third party who is not authorized to do so. A picture could be described using attributes like *ownership, expiry date, re-share/re-post, taggable*, where *ownership* refers to the picture's owner (e.g. the person who took it), *expiry date* can be used for short term resource availability, *re-share/re-post* allows or denies re-sharing/re-posting the picture, and finally, *taggable* can be used to allow or deny others to "tag" or name people in the picture.

Although we realize there is no way to actually enforce other users or applications to respect this set of attributes, at least it can help to define a set of "best practice" rules.

6. CONCLUSION

In this paper, we tried to emphasize the advantages of switching from a silo-based user account (profile) system to a decentralized user-controlled one. Not only would this system provide better control of a user's online identity, but it would facilitate and improve the way in which we currently interact with other people within the Web. Additional advantages from using our solution lead to a significant reduction in network load, as well as making spam impractical. These advantages result from using a *poll* system, where no messages are explicitly being sent to the other participants. In our solution, online data remains under the user's control, on a device controlled by the user. Additionally, creating and managing groups of users takes place according to specific contexts. More importantly, we offer privacy by design instead of an additional feature.

[7]http://wiki.debian.org/FreedomBox

7. REFERENCES

[1] BEA, IBM, Microsoft, R. Security, and VeriSign. Ws-federation: Passive requestor proïňĄle. *http://www-106.ibm.com/developerworks/webservices/*, 2003.

[2] T. Berners-Lee. Semantic web road map. *http://www.w3.org/DesignIssues/Semantic.html*, 1998.

[3] T. Berners-Lee. Linked data-the story so far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[4] J. Breslin, A. Harth, U. Bojars, and S. Decker. Towards semantically-interlinked online communities. *The Semantic Web: Research and Applications*, pages 500–514, 2005.

[5] D. Brickley and L. Miller. Foaf vocabulary specification. 2005.

[6] M. Erdos and S. Cantor. Shibboleth-architecture draft v05. *http://middleware.internet2.edu/shibboleth/docs/draft-internet2-shibboleth-arch-v05.pdf*, 2002.

[7] IBM. Enterprise security architecture using ibm tivoli security solutions. *http://www.redbooks.ibm.com/abstracts/sg246014.html*, 2002.

[8] D. P. Kormann and A. D. Rubin. Risks of the passport single signon protocol. *Computer Networks 33*, 2000.

[9] Microsoft-Corporation. .net passport documentation, in particular technical overview and sdk 2.1 documentation. *http://www.passport.com and http://msdn.microsoft.com/downloads*, 2001.

[10] OASIS-Standard. Security assertion markup language (saml). *http://www.oasis-open.org/committees/security/docs/*, 2002.

[11] D. Recordon and D. Reed. Openid 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital identity management*, pages 11–16. ACM, 2006.

[12] H. Story, B. Harbulot, I. Jacobi, and M. Jones. Foaf+ssl: Restful authentication for the social web. In *Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2009)*. Citeseer, 2009.

[13] S. Tramp, P. Frischmuth, T. Ermilov, and S. Auer. Weaving a social data web with semantic pingback. *Knowledge Engineering and Management by the Masses*, pages 135–149, 2010.