1. Introduction

Using the Internet has become part of the daily habits of a constantly growing number of people, and there are few human activities that can be performed without accessing the enormous amount of information that it provides. More recently, the process of accessing has evolved to become a bidirectional exchange of information, and we are today in a phase where the user is a consumer but also a producer of information. This trend started with the appearance of the first blogs, evolved with the paradigm the virtual reality, and has reached its current maturity with the emergence of the so called social networks.

But as the opportunities for publishing one's thoughts, pictures, videos and whereabouts grows, the need for having these blobs of information associated to our own selves creates the need of providing information about our identity, so that one can be recognized and remembered, and our contribution does not remain anonymous. This, of course, raises the issue of managing identities, and since it takes place in a digital world, we may refer it as the issue of *Digital Identity Management*.

In fact, one of the biggest challenges that the evolution of the Internet is facing today, is related to the question of Identity Management [1].

2. Centralized Identity Management

This problem is not new, and has been addressed by almost all enterprise or institution, large or small, in the course of the last decade. As the usage of computers and information systems became essential to every commercial or institutional activity, the correct management of user identity has risen as a fundamental condition for streamlining the business processes, and myriads of men hours were spent designing and implementing Identity Management systems. The problem then was to solve the issue of making every corporate or institutional user present the same set of *identity credentials* to all the platforms that constituted the institution's information systems.

Managing identity includes the following steps:

- Provisioning, which designates the process of creating a set of credentials for every user or employee, such as username and password to enforce *authentication*, and the set of attributes defining their role and privileges to implement *authorization* and *access control*;
- Storing, which involves the creation of a repositories with the credentials of all the users, so that all applications can retrieve them when needed;
- Enforcement, which consists on using the users' credentials to implement the authentications and authorization policies which are defined for a particular application and a specific user;
- Maintenance, which designates the possibility of changing and updating the credentials content and attributes
- De-provisioning, that designates the removal of all the information associated with users or employees, whenever their association with the institution terminates.

Last but not least, one of the most important goals of identity management systems is to implement *Single Sign-On* access, which consists on allowing users to authenticate only once to the system, by providing their credentials in a unique transaction. From then on, all the platforms of the information system become aware of the users' identity and attributes,

and do not need to validate them again. This process involves the establishment of strong trust relationships between the identity management system and all the applications, which, depending on the technologies and heterogeneity of the components involved, may be straightforward or extremely hard to implement. Even today, when almost all corporations and institutions rely on complex information systems to support their business or activity, correct identity management practices are not always implemented everywhere, and Single Sign-On often remains the Holy Grail always pursued but rarely attained.

3. Web Identity Management

When we consider the implementation of identity management in the Internet, one of the first things to consider is the enormous disproportion of the scale factor. Users in the internet are counted by billions and this fact only, completely changes the nature of the problem, obviously quantitatively, but also qualitatively.

Whereas in the confined environment of corporations, centralized repositories and management systems may be used to implement the identity management cycle, in the Internet there is a immeasurable number of users, an equivalent number of applications or platforms which reclaim user credentials and attributes for authentication and authorization, and to make things even more complicated, nothing prevents users to have several identities (*personas*) which they may impersonate, according to the environment where they are required to identify themselves.

Besides, due to the decentralized nature of the Internet, it is neither possible nor desirable that one single entity becomes responsible for the management of the multiples identities that each user may adopt. In fact, the extreme fragmentation of individual characteristics and the global scope of their validity implies that the users themselves users should be the sole managers of their identity. This fact implies a complete paradigm shift in the way identity is managed in the Internet, which led to the concept of *User Centric Identity Management* (UCIM) [2].

UCIM went through many phases in the recent years. When the so called electronic commerce platforms appeared at the turn of the century, users started to be queried to provide some form of identity that could guarantee the payment of the items they bought. This was generally done by creating an account with whatever credentials they wished, as long as it was associated with a valid method of payment, generally an internationally accepted credit card. Therefore, one the first UCIM platforms were implemented by on-line shops, which as the volume of on-line transactions increased dramatically, became important identity hubs, aggregating millions of users.

When the type of interactions in the Internet evolved from monodirectional many-to-one to bidirectional many-to-many, this model of UCIM became inadequate to support the new access paradigms. In fact, users became clients of thousands of on-line shops, members of myriads of communities, and subscribed innumerable information seeds. Initially, they were obliged to keep track of every subscription credentials they provided to each site social hub or shop, or the same credentials were used to all subscriptions, which created obvious confidentiality problems.

In order to solve this nature of problems several technologies were developed, which implemented the concept of Web Single Sign-On. *OpenId* [3] is one of the first protocols to provide this kind of functionality. It relies on the following 3 entities:

1. An *Identity Provider* (IdP), which stores user identifiers and credentials, and responds to authentications queries

2. A *Relying Party* (RP), the entity that protects the resource to which an unauthenticated user wishes to access, redirects the request to the user's IdP, and that requests or denies access according to IdP response.

3. A *Personal Identifier* that represents the user, which is obtained according to a specific method.

Obtaining an OpenId identifier is a simple procedure, which involves no proof of identity apart from a valid email address. It consists of the following steps:

1. A user requests a personal identifier from an Identity Provider, which can contain any valid string of characters provided by the user (name, number, etc..)
2. The Identity Provider requests information from the user, which is of their sole responsibility, including the user's credentials and an email address that has to be valid. No matching between the user credentials and email address is required
3. The IdP sends an email to address provided, containing a unique random code that has been generated for email validation
4. The user accesses the IdP and provides the code received, and if it matches the one sent, the email address is validated and the personal identifier is associated with it
5. The IdP stores the Personal Identifier and associates it with the information provided by the user
6. The user my subsequently modify their personal information that was initially provided.

An OpenId identifier is in fact an URL, which contains the user provided string of characters (nickname, number …) and the name of the IdP: http://user.provider.domain.

The OpenId authentication protocol obeys to a specific sequence of interactions, at the end of which the user identity is accepted by the Relying Party, although it never receives nor accesses the user's authentication credentials. A simplified version of this sequence is the following:

1. An unauthenticated user tries to access a specific resource (a site, web shop, social network …) following the OpenId protocol, designated by the Relying Party (RP).
2. The RP requests the user's OpenId personal identifier.
3. The user provides their OpenId identifier, generally by filling a form.
4. The RP extracts the user's IdP URL from the personal identifier.
5. The RP redirects the user request to the URL of the corresponding IdP.
6. The IdP presents a login form to the user, which they fill with the credentials initially provided when the identifier was created.
7. If the credentials are accepted, the IdP returns the user's request to the RP, containing an assertion which guarantees that a successful authentication was performed and that the user is legitimate.
8. The RP validates the assertion and allows the user to access the required resource.

In the authentication sequence described above, the validation of the IdP assertion performed by the RP is based on a specific secure functionality, called digital signature, which implies that the RP and IdP exchange a pair of cryptographic keys (one public and a another private), through a well known security algorithm (the Diffie-Hellman Key Exchange). However, if we consider the registration sequence, an OpenId identifier can be obtained by anyone that possesses a valid email address, and no other kind of identity verification is performed.

From these remarks we may conclude that, although the exchange of information between the RP and the IdP is performed under adequate security conditions, the user is in fact the sole entity responsible for the validity of the information exchanged, which is the key characteristic of a *User Centric Identity Management* platform like OpenId. Taken this into account, it is legitimate to ask the following question “If no one can guarantee that users are in fact who they pretend to be, isn´t this complex process of authentication completely useless?” There are several answers to this question.

If we consider the user point of view, this process guarantees that as long as they have not disclosed their credentials to someone else (or that no sophisticated security attack has been perpetrated), no one else can impersonate this *particular identity*. This is exactly what is needed in most situations, where users subscribe to a specific shopping site or social network, and they wish to keep their specific information private. Besides, independently of whom they are, users are definitely associated with their specific identifiers, and every time they return to a given site, they have exactly the same treatment as before. This provides exactly the same behavior as if the user had registered to a site using an ad-hoc identity, except that they don´t need to create an ad-hoc identity for every site or social network they wish to register with.

From the point of view of the relying party, this authentication process solves the following issues: first it guarantees that independently of the users’ real identities, every time they return to its site, they are exactly the same person as before. This allows the establishment of whatever policies are used to determine users’ behavior, interests, maintain private data associated with the account, etc. Second, it greatly simplifies the user identity management tasks, since part of the provisioning process and the complete authentication process are outsourced to the user and to the Identity Provider.

These reasons are valid enough for this type of Identity Management to present real advantages for both parties, and the growing adoption of platforms of the type of OpenId by the most important players in the Internet, like Google, Microsoft, Facebook and others is a good proof of its relevance. In fact, many of these companies developed their own versions of identifiers, Identity Providers, and authentication protocols, etc, and the current landscape of Identity Management in the Internet is governed by a few key identity hubs, that provide identifiers and authentication mechanisms, that most of the less important players tend to adopt in order to benefit from the existing identity communities. One of the relevant example of this situation is the fact that many entertainment content providers, like IMDb (Internet Movie Database), allow users to login to their accounts using Google or Facebook indetifiers.

4. References

[1] “Identity in an Information-Centric Internet”, MIT Privacy & Securuty Working Group, 2008, http://cfp.mit.edu/CFP_Papers/Identity%20Whitepaper%20v13_clean-3.pdf

[2] "User Centric Identity Management", A. Jøsang & S. Pope, 2005 Information Technology Security Conference , Gold Coast, Australia, http://persons.unik.no/josang/papers/JP2005-AusCERT.pdf

[3] “Get Ready for OpenID”, Rafeeq Ur Rehman, 2008, Editor Conformix Technologies, ISBN-10: 0972403124.