# PROGRESSES IN OPEN IDENTITY MANAGEMENT SYSTEMS FOR THE BANKING SECTOR

*Ion-Petru POPESCU*[1]

**Abstract**

*The introduction of (open) identity management systems, in the financial organizations, beyond purely technical means, is a very costly and provocative enterprise. In spite of this, for the decision taker, the impact and, in the same time the opportunities provided by such systems seem to be difficult to grasp, to say at least. This presentation explores all the relevant aspects of an ex-ante classification and the general evaluation of main categories of IM systems. As such, one proposes the examination of this vast domain via an interview type - classification study for a better understanding of the nature of the IM systems, even more so, because these specific system stand between IT production and IT security systems. With this in mind, the accent is laid on the general nature of the implementation IM systems, with the presentation being focused on a pros vs cons analysis for such future projects during the bank's strategic decision investment placing phase.*

**Keywords:** *Online transactions, services, business to business, processes, identity management, business process, authentication, identity mode, domain, federation, OpenID*
**JEL Classification: M15, O32**

## 1. INTRODUCTION

Looking at the current online world, performing transactions as online banking, online shopping or communicating in social networks has become an inherent part of life. Hereby, personal, identity-related data plays a major role, since for many activities a service provider requires details about the identity of a user. Traditional approaches for identity management, like the application-centric or isolated model (Jøsang, 2005), require users to register with every single service and to re-authenticate each time they use a service in another trust domain. Over the time users register with several applications on the Internet and collect many digital identities together with their corresponding authentication credentials. This leads to a number of well-known problems. To name a few, users have for example difficulties to remember their passwords, and also bear a great burden to keep their account information up-to-date (cf. Bertino, Martino, Paci & Squicciarini, 2009; Ahn, Nam Ko & Shehab, 2008).

To overcome the limitations of the closed domain, open identity management models emerged as a way of sharing identity information across several trust domains in a controlled manner. The basic principle behind these new identity models is to manage and keep identity data in multiple trust domains, at so called *identity providers*, and to share this information with applications and services that are willing to rely on it. Hence, these applications and services are also called relying parties. Open protocols and standards as OpenID (The OpenId Foundation, 2007), Information Cards (OASIS, 2009) or WS-Federation (Lockhart, et al., 2006) already exist and form the backbone of the new models.

Nevertheless, the adoption of open identity management models has not set off tangibly, yet. The acceptance of the new models mainly depends on the willingness of services and applications to rely on information that they retrieve from foreign sources that are outside their own trust domain. Up to today, this willingness is very low. Each service provider

[1]PhD Student at The Bucharest University of Economic, ion.petru.popescu@gmail.com

usually forms an isolated identity domain. Looking at the reason, this development is a little surprising. Organizations often have strict legal requirements and policies for the management and storage of user data. Moreover, a company's user database constitutes often one of the most valuable assets of a company. Therefore, it is not surprising that organizations find it hard to give up this control and to rely on user information from a partner.

However, especially, with regard to the Internet, we can find many use cases that do not require a strong trust relationship to rely on identity attributes from someone else. Often the user can enter information into his account that does not require any verification. It really depends on what a digital identity is used for. If the user logs on to a site to prove on repeat visits that it is the same user, it does not matter whether his digital identity matches with his "real-life identity" as long as it is always the same digital identity he uses to log on. Only if critical transactions are performed, as ordering an item or paying for a service, the integrity of provided user data is required to hold the user liable in case anything bad happens. Taking all these considerations into account, it becomes obvious that the willingness to believe in identity data from a foreign source is closely related to the trust level that is required by the transaction a user wants to perform. The more critical a transaction is, the more assurance into the identity of a user will be required by the relying party to accept identity data.

In order for relying parties to match the transaction requirements with those of their partners, proper assessment mechanisms for identity assurance are needed. In order to ease the process, identity assurance frameworks have been proposed and developed as a mean to define a global trust level that allows an immediate comparison between participants even though they might not know each other.

This article is divided into two parts. The first part focuses on the classification of identity management models. As suggested by Jøsang et al. (2005) there are four models, which are the outcome of a development that is still ongoing. Accordingly, they are introduced in the chronological order given by this development. The four models are: *application-centric*, *centralized*, *user-centric* and *federated identity management*. Each model is described with its strengths and weaknesses, typical use cases, as well as the technologies to implement the model. Selected technologies for open identity management as OpenID, Information Cards and WS-Federation are described afterwards.

The second part of the article focuses on identity assurance and its role towards a reliable identity management in open networks. Existing identity assurance frameworks are presented and their shortcomings and limitations are discussed. To overcome the limitations, the final part of the article suggests trust levels for identity attributes as an emerging trend in future research.

## 2. BACKGROUND - IDENTITY MANAGEMENT: DEFINITION AND APPROACHES

Identity management is one of the most pervasive parts of IT systems. The reason is that almost all IT systems operate on assets such as personal data of customers and employees that need to be protected. Thus, in order to allow a person access to the system, the systems needs to know something about the identity of the subject. This knowledge is required to make access control decisions (Benantar, 2006).

Identity management comprises the whole process of gathering information about a subject, storing this information in an account and distributing the information along the systems requesting parts of a user's digital identity. As described by Windley (2005), a digital identity follows a continuous lifecycle: An identity is created (provisioned) in the beginning, propagated and used for a certain time and destroyed once it is not needed anymore. Most works share the same or a similar definition of a *digital identity*, compare for instance

(Windley, 2005), (Bertocci, Serack, & Baker, 2007), (Jøsang, Fabre, Hay, Dalziel, & Pope, 2005). A digital identity is a limited set of attributes represented by a unique identifier such as an account name or number, which is associated with a person's "real-life" identity. It is the task of *identity management* to control and facilitate this lifecycle.

**Conceptual Evolution of Identity Management: From Domain-Based Identity Management Systems to Open Identity Management Models**

Over the years, identity management has undergone a substantial change. In the beginning, computers were mostly isolated systems, hardly connected to the outside world, and mostly supporting the processes inside a company. Over the years, the IT system landscape has changed to a highly computerized and interconnected world, in which services are offered and consumed over the Internet. Naturally, this also had implications to the identity management. Caused by the change of system structures in IT, a number of identity management models emerged, each addressing the particularities of new IT system landscapes.

Coming from the traditional, application-centric models, the centralized identity management model emerged to provide a more efficient identity management inside closed domains. Again, with the shift of IT systems from the closed to the open world of the Internet, new open identity management models emerged which were designed specifically to address the open nature of these environments. Some of the identity management sources (see for instance Chappell, 2010), even state, that there is a major paradigm shift about to happen which leads from the traditional domain-based identity management approaches such as the isolated-/application-specific or the centralized identity management to open identity management systems. The shift affects in particular the basic conceptual approach how digital identities are represented in IT systems and shall be explained in the following.

**Domain-Based Identity Management Systems**

*Domain-based Identity Management Systems* are the traditional approach to manage users in applications. In the very early years of computing, application developers were in need for a way to recognize their application users in a reliable manner. The solution was mostly to augment applications with authentication and identity management features. In order to represent a user in the system, the notion of an *account* was created. An account is a unique identifier that is associated with an authentication secret, such as a password, to authenticate a user on subsequent visits.

As the application-specific identity management resulted in highly proprietary and isolated identity management solutions, centralized approaches were developed to take the burden of managing user accounts from the application developers. In the centralized approach, all users within a domain are managed by a single entity that provisions identity data to the applications. In particular with the advent of multi-user operating systems (e.g. Windows 95) centralized identity management was henceforth available with the platform and could be used directly in the applications. Later centralized identity management found widespread adoption inside the homogeneous networks of organizations.

**Open Identity Management Systems**

Domain-based identity management systems usually support a fixed identity model and use protocols that are only available inside closed domains. Beyond the borders of a closed domain, a world of heterogeneous networks emerges that does not know about identity attributes within closed domains. Also, many technologies, including for example Kerberos, usually reach only to the border of homogenous networks. Therefore, new technologies were

needed to deal with the heterogeneous environment of the Internet and to allow applications to share identity information seamlessly across domains.

Open identity management systems evolved as a way to connect existing identity management systems over the Internet. Besides interoperability between different technologies, open identity management models also feature an open identity model that allows any type of identity attribute (claim) to be included in a specific digital identity. Each attribute is named with an abstract identifier (e.g. a URI), which can be used by applications to access the attribute. The formats to exchange tokens are open and extensible to incorporate the identity model. In addition, the identity management system translates between different attribute/claim dialects, in order to provision applications with the attributes/claims they understand.

## 3. CLASSIFICATION OF IDENTITY MANAGEMENT

Today, mainly four different identity models can be found in the literature (see for instance Jøsang, Fabre, Hay, Dalziel & Pope, 2005 or Benantar, 2006).

A commonly used classification is the one in domain-based and open identity management models that has been introduced before. Looking at the typical area of usage, another vertical classification into end-user oriented and business-oriented models can be found. While the application-centric and user-centric identity management is often used in the private sector, the centralized as well as the federated identity management can be found predominately in the business context. The federated identity management in particular targets business-to-business scenarios and the centralized identity management is used by organizations to allow their employees and members to access various applications within a single domain.

In the end-user domain, the user-centric models targets end-users authentication on the Internet while the application-specific identity management is best suited for the domain of a single application. In the following, each of the four identity models shall be characterized in detail.

### 3.1 Application-Centric Identity Management

In the application-centric / isolated identity management, each application takes care of its users itself. This implies that all users need to register with the application/service during a registration step before actually using the system.

Since the beginnings of computing, application developers and designers are challenged with the need to identify users on subsequent uses of their applications. Examples include the need to personalize views and/or to ensure that the right users get access to the systems. The application-centric identity management was one of the very first approaches to add identity management features to applications. The way, it works is, that authentication and identity management features are directly built into the applications, resulting in highly proprietary, isolated identity management solutions. Each application has its own identity model containing a set of attributes appropriate for this specific application.

#### Advantages and Disadvantages

While the application-centric model works well for single applications, it seriously lacks scalability when being applied to environments with multiple applications or services. Just consider a service-oriented architecture with a multitude of services. In such an environment, implementing identity management for each service is not recommended, as the possibilities to compose services to applications and to reuse services in various contexts

would be lowered tremendously. Moreover users would need to register for each service separately. This does not only lead to an explosion of user accounts, but also bears significant security risks as users are overstrained with the sheer number of passwords they need to manage properly. A classical, but from the security point of view very dangerous, strategy is here for example to use the same easy-to-remember password for multiple accounts. Also when looking at the application provider side, the application-centric identity management is not always the best option. Setting up proper registration procedures is a costly and time-consuming task. On the other hand, using the application-centric model the application has its own identity model and hence full control over the identity attributes as opposed to being bound to a pre-defined identity model.

**Usage Scenarios**

While inside closed domains, as in most companies and organizations, the longing for single-sign-on features has widely replaced the application-specific model with more centralized solutions, one could observe its revival in the open world of the Internet in recent years.

Just think of some popular web applications as Facebook, Ebay or Amazon, that all require users to register. One of the reasons is its easy initial set up that does not require the establishment of strong trust relationships between identity providers and relying parties.

**Technologies**

Most applications have a built-in username and password database to store and manage its users.

### 3.2 Centralized Identity Management

In the centralized identity management model, all users within a domain are managed by a central instance, which provides identity provisioning for multiple applications within the same domain.

Multi-user operating systems (e.g. Windows 95) were among the first systems to offer a centralized identity management. All functionality to manage digital identities of users was built directly into the platform and therefore separated from the applications itself. As a result, application developers could concentrate on their actual business and did not need to care about developing a complete identity management for their applications. The only thing to care about was the integration of the identity management upon which they built. Today, a centralized identity management system is part of almost every company or unit that is in some way organizationally independent. As a result identity information about all members within the organization is hold at a central place and can be accessed by applications within the domain. This also leads to major benefits for the users of the system. Since identity management is centralized, users have to authenticate only once to access a multitude of applications. This feature is called single-sign-on. Also, user data is not stored redundantly in separate places and therefore easier to maintain.

**Advantages and Disadvantages**

Separating the business logic from the identity management systems and centralizing it yields significant advantages for developers and users alike. Application developers are relieved from implementing a whole identity management system. Their only task remains to integrate their application into the identity provider platform. The downside, however, is that these environments usually have a pre-defined identity model, which is fixed and most often not extensible. This means application developers are restricted to the capabilities of the

underlying platform and can usually only use the identity attributes that are given by the used identity system.

On the user side, centralized identity management allows to implement a single-sign-on for multiple applications within the same domain. The direct benefit for the user is that they can logon to several applications using a single account and are relieved from remembering authentication credentials for every single application. However, caution has to be taken as a single centralized instance also bears risks for the users. Since all applications have access to the same database of users, trust is an important factor. While in the isolated identity management model, a user only needs to trust on a single application, in a centralized environment, he needs to have faith that all applications within a domain will use his identity data in the intended way. Usually this is easy in a closed environment of a single computer, it is also possible in the restricted environments of a company's network, in which the employee has no other choice but to trust on the company's network and applications. However, in an open environment as the Internet, each participant usually forms its own trust domain and it becomes increasingly difficult to believe that all participants act in the best interest of the users. Trust between domains does not come natural and requires a certain effort to be established. The more trust a transaction requires, the more effort it takes. Just imagine for example two companies establishing a partnership. As soon as transactions between the two involve a risk for either side, legally binding contracts are set up to balance the risk.

Another example showing how important it is that users have trust in the manager of their data is Microsoft's Passport (later.NET Passport, now Windows Live ID). Microsoft Passport offered a centralized identity management for the Internet. Several applications on the Internet could make use of Passport to authenticate users by their MSN account. However, the user base never really exceeded the number of MSN users. As the system was not based on open standards and Microsoft was the only identity provider, many users and application developers choose not to use it (cf. also (Cameron, 2005)). Later Microsoft developed Information Cards and Cardspace as an identity management system for the Internet that allows having not only one, but multiple sources to manage identity data and that is based on open standards.

**Usage Scenarios**

Following up on the discussion of advantages and disadvantages, the centralized identity management is in particular suitable for centrally managed environments as the closed domain of an organization. Indeed, almost every organization, which forms an independent administrative domain, uses the centralized identity management to store information about users at a central place and foster a single-sign-on for various applications within the company's network.

**Technologies**

Within the closed domain of an organization, usually a centralized directory, such as Active Directory, is used to hold the identity data. For the authentication of users by the applications in the domain, several technologies can be used. One technology that is often used is Kerberos; other possibilities include SAML or public key infrastructures.

### 3.3 User-Centric Identity Management

In the user-centric identity management, the digital identity of users is managed by various sources, the identity providers. The user is in the centre of all interaction and chooses upon request the digital identity he wants to use to authenticate with a certain application or service.

The user-centric identity management uses an approach to identity management that is very similar to the way identities are used in the real world. In our daily life, every one of us possesses a number of identity cards to prove certain claims upon request. Just think of a driver's license to prove that one is eligible to drive a car, a passport to prove our citizenship or an ATM card to prove our ownership to a bank account. All these cards are issued by different authorities, our real-world identity providers. We carry them around and show them upon request. For example in order to retrieve a discount in the movie theater, our student card is requested by the ticket salesperson as a proof of us being a student.

User-centric identity management works exactly this way. Every user in the online world has his identity data stored with one or more identity providers. Instead of using one identity provider all the time, the user can choose which identity provider he wants to use for a certain application or service. This identity provider is contacted to assert a certain claim, such as "This user's name is Bob." This assertion, the counterpart of the real-world identity card, is given to the requesting application. As in the real world, the application provider now decides whether it trusts on assertions by this identity provider and either accepts the information or requests further proof for the same claim.

**Advantages and Disadvantages**

As the name suggests, the user-centric identity management puts the user into the heart of all decisions. The user decides with which identity provider(s) he wants to register an account with. The user also decides which of his identity providers he contacts to make assertions about his identity. And the user also decides which of his identity attributes are given to the requesting party. The clear benefit is, that the user enjoys more privacy and has full control over his data and knows who is using it and when. In fact, in the decentralized model the identity providers typically do not know where and what a user is using his identity for. Only the relying parties, such as a service or an application, know about the identity provider(s); otherwise they would have no basis for making a decision to trust an assertion.

**Usage Scenarios**

With the dynamic establishment of trust relationships and the relying parties deciding which identity providers to trust, the user-centric identity management fits best to non-business scenarios. Usually it is used in situations with a low demand for identity assurance, such as registering for social web sites as for example weblogs or forum discussions. In fact, it is often used as an alternative way for website users to fill in registration forms or as a way of identifying a user on repeat visits. In both cases, the web site might not be interested, whether the user's digital identity matches with his real-life identity, as long as it is always the same user it interacts with.

Furthermore the user-centric identity management is also beneficial in all scenarios in which the set up of contracts to establish trust is not feasible. This is for example the case between an Online Shop and the government. In this case, the government takes the role of the identity provider, which is trusted by the online shop without an underlying contract. In fact, the trust relationship is based on the pure existence of an authority and its reputation as a trustworthy partner.

**Technologies**

The user-centric identity management model is the newest addition to the identity management world and therefore technologies have just developed recently. Technologies have been designed specifically for the open and decentralized environment of the Internet, which is the main use case for this young identity management model. Popular technologies include OpenID (The OpenId Foundation, 2007) and Information Card (OASIS, 2009).

### 3.4 Federated Identity Management

In the federated identity management several independent trust domains form a *Circle of Trust*, in which all participants agree on trusting each other's assertions about user authentication and attributes for the purpose of access control and single-sign-on.

Similar to the user-centric identity management, federated identity management also is one of the new open identity management models that aim at providing user authentication and access control in the global context of the Internet. Looking at the Internet today, we mainly find an environment of independent trust domains formed by independent organizations. There is no central instance to manage identity data, but many isolated identity islands, each having its own identity management system. Isolation allows companies to retain control over their users and identity management systems. As organizations usually have very different legal requirements and policies for identity management, they find it difficult to give up this control. In consequence, it would be almost impossible for them to agree on a common centralized solution with their partners and customers.

Using federated identity management there is no need to give up this control in order to allow members of one trust domain to use their digital identities in a partner domain. The basic principle to make this happen is the trusted federation relationship established between identity providers and service providers. Identity providers and service providers affiliate into federations by agreeing upon common obligations and policies that each federation member needs to adhere to. This process is usually accompanied by contracts each federation member signs. As a result, a Circle of Trust forms, in which assertions about the authentication of users and attributes are shared among the federation members.

Technically, each federation member stays in control of its own identity management system, but augments this with additional federation features that allow users to link (federate) their digital identities between the federation members. Certain identity management functions like authentication or provision of identity attributes are then offloaded to the identity provider(s) in the federation. Identity consumers on the other side receive this identity and authentication information from the providers and use it as if it was coming from their own identity management system.

**Advantages and Disadvantages**

Federated identity management is primarily a way to allow single-sign-on (SSO) between partner organizations regardless of organizational borders. Members of one organization such as employees of a company can link their account with accounts they might have with other organizations in the same federation. Once linked, a member can access all connected accounts by authenticating just once, allowing him to sign in to a number of application at the same time. Of course, the more members the federation has, the more a user will benefit from SSO.

However, there is also a downside to the federated identity management concerning the privacy of the users. In a federation the identity provider "sees all"; that means it knows which relying parties a subject visits. Given this information and the identity data of the user, a malicious identity provider could track the users behavior and would hold a rather comprehensive profile of a user. Also, without proper protection mechanisms, identity providers and service providers are in the position to match different digital identities of the same user for the purpose of creating an even more comprehensive user profile that can be used to provide personalized offers. Another threat arises from account linkage, once the user password is compromised.

### Usage Scenarios

Due to the federation agreements necessary to build up the Circle of Trust, federated identity management is mostly used in business-to-business scenarios. Typical use cases include the authentication of employees of one company in a partner company or the federation of companies offering complementary services for their customers. Using federation, employees can get easier access to shared project resources or can use services of the partner company, such as booking a business trip, without an additional authentication step. In the second case, customers are offered to link their accounts between the collaborating companies in order to create a better shopping experience.

### Technologies

Technologies for federated identity management exist mainly in the field of service-oriented architectures. In the past, two initiatives have formed to develop a standard for the interoperable exchange of identity information across organizational borders on the basis of web services. As a result, we find today on one side the specifications of the Liberty Alliance (now Kantara Initiative (Kantara Initiative, 2010) with SAML 2.0 (Cantor, Kemp, Maler, & Philpott, 2005) as a standard to describe identity information in an interoperable format and on the other side, WS-Federation (Lockhart, et al., 2006), a specification developed by IBM and Microsoft.

Originated as two separate specifications, latest efforts have driven a development towards interoperability between both specifications. (cf. OASIS Cover Pages, 2008)

## 4. OPEN IDENTITY MANAGEMENT STANDARDS AND TECHNOLOGIES

Past experiences have revealed that traditional solutions as the application-specific and the centralized identity management work well in closed domains, but fail when they are applied to open environments containing multiple trust domains. Kim Cameron explains in his Laws of Identity (Cameron, The Laws of Identity, 2005) the successes and failures of digital identity systems. One of the main findings is that an identity management for the Internet needs multiple identity providers (Pluralism of Operators and Technologies). As the Internet was built without a central instance, there will also never be a central instance to manage identities. Instead we have several identity providers, either administrated by the government or big players as Google or Facebook. Nevertheless, potentially any application provider can take the role of an identity provider. Therefore, interoperability between different identity systems is one of the main requirements for an identity management for the Internet. Looking at the Internet, a number of technologies have been designed to address these needs.

### OpenID

OpenID is a very light-wide protocol providing a single-sign-on for browser-based applications. It is built on top of HTTP. A digital identity in OpenID is represented as a unique URI, which contains besides the username of a subject also information of the identity provider. In order to authenticate with OpenID at a supporting web site, the user enters this unique URI instead of username and password. The website as the relying party redirects the user to his identity provider including a request for authentication. If the user is already logged in at his identity provider (a session already exists), the identity provider answers with an authentication assertion and the user can log on to the website without further authentication requests (direct single-sign-on). If the users has not authenticated at his identity provider previously, he is redirected to the login page of his identity provider and asked to log

in, for example by means of a user name and password. Upon successful authentication, the identity provider sends an authentication assertion back to the requesting party, the web site.

**The Identity Metasystem, Information Cards and CardSpace**

The Identity Metasystem denotes an open architecture for the interoperable exchange of identity information between identity providers, relying parties and the user. As identity management solutions differ in the way identities are described and exchanged over protocols, the Identity Meta System aims at connecting different solution by adding an abstraction layer on top of existing solutions that hides the specifics of each identity system (as in the case of IP over Ethernet and Token Ring). The Identity Metasystem describes concepts that are equal in all identity solutions in an interoperable format and specifies how identity information can be translated between different systems.

Information Card is a concrete implementation of the abstract concept of the Identity Metasystem. It has been developed by Microsoft and implemented in several frameworks such as Suns Metro Web Service Stack or the .NET Framework.

Information Card uses unique URIs to describe identity attributes in a global context, so called claims. For example, if a relying party requires the name of a subject, it refers to this attribute as http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname.

In order to request identity information from a user, the relying party sends a request based on the specifications for web services to a piece of software running on the computer of the user. This piece of software is called Identity Selector and holds the information about all identity providers a user has registered his identity with. Upon request from a relying party, such as a web site or a web service, the identity selector matches the requested attributes with the attributes each identity provider of the user can provide and presents the user with a set of matching providers. From this set, the user can choose the identity provider he wants to use and is requested to authenticate. Upon successful authentication, the identity provider asserts the requested identity attributes by writing them into an interoperable format and signing the values with its private key. Once the relying party retrieves the information, it can check its integrity and use the information for example for the purpose of access control.

CardSpace is an implementation of an identity selector that is installed per default in Windows Vista. Other implementations that can be used in the context of Information Card include Bandits DigitalMe (The Bandit Project) or various plug-ins for Safari and FireFox.

**WS-Federation**

WS-Federation is a specification in the field of service-oriented architectures. As a specification for federated identity management it sits on top of other web service specifications such as WS-Trust (Nadalin, Goodner, Gudgin, Barbir, & Granqvist, 2007), WS-MetadataExchange (Ballinger et al., 2006) and others. WS-Federation provides mechanisms and protocols to establish a Circle of Trust, the federation, between trust domains with the intention to exchange identity information between the federation partners. Microsoft provides an implementation of the WS-Federation specification as part of their.NET Framework.

As part of the federation process, WS-Federation specifies the following mechanisms: exchange of federation metadata to create a federated relationship, account linking of accounts in different trust domains, management of pseudonyms to protect the user's privacy, as well as single-login and–logout.

## 5. FUTURE RESEARCH DIRECTIONS: TRUST LEVELS FOR ATTRIBUTES

The main purpose of an identity service is certainly the provisioning of identity attributes of registered users to parties that are willing to rely on it. For this reason, identity attributes are usually in the main focus of the provisioning process. Services can request certain identity attributes and will get the corresponding values if entitled to. This is the way claim-based identity in its core works. The unit of all transactions is the claim, which is an identity attribute that is subject to verification by an identity provider and is issued upon request to the relying party.

Trust and assurance frameworks on the other side put the identity providers and their processes, mechanisms and technologies to safeguard user identities in the middle of all considerations. An identity provider is assessed by basically all the mechanisms and technologies in place, such as credential and token management, legal aspects and storage of identity data that have an influence on the degree of confidence a relying party can put into the assertions of this identity provider.

Certainly, all these considerations are worthwhile. However, with regard to open identity management and its claim-based approach, an important part is missing to close the gap between the identity provider-centric view of identity assurance frameworks and the attribute-or claim-based view of open identity management models.

As has been stated in (Thomas & Meinel, 2009) trust should be defined on the same granular level as the identity values themselves. This means, that the decision to trust should not only be made between the issuing and the relying party on a all-comprising level–as this is also very difficult to achieve—but for each identity attribute, which is exchanged, separately. To give an example, we could consider a university that is trusted to make right assertions about whether a user is a student, but not about whether this user pays its telephone bills.

Work exists (Thomas & Meinel, 2010) which proposes a layered trust model that distinguishes between the overall trust into an identity provider and the trust into the identity of a user. The first layer, the trust into the identity provider, is defined as the degree of confidence that an identity provider has the proper mechanisms in place to makes right assertions. In particular, this also includes the legal situation as well as the adherence to governmental guidelines and laws. Based on this general trust relationship that is supposed to be relatively static, a second layer of trust is added. This layer, which is called identity trust, is defined as the degree of confidence a relying party can put into the identity of the subject of the assertion. This layer is based on the first layer and separates the static properties that are mainly related to the identity provider as a trusted entity from the properties that are subject to vary over the course of time. This includes for example attribute values, that are subject to expire after a certain time or which are entered into the system without verification and maybe verified later on if needed. The way an attribute has been verified and the source where it comes from make up the trust level of the attribute. Additional research is required to find out exactly which factors are relevant to assess the trust level for attributes and in which way the overall trust level of the identity provider relates to the trust levels of attributes.

Time has shown that the traditional centralized identity management does not work in open environments as the Internet and Service-oriented architectures and therefore new approaches to identity management are necessary to serve the needs of these networks. In this book article, we have drawn the conceptual evolution of identity management models from the classical domain-based approaches to the new open identity management models. We provided a classification of identity management models and discussed each of the four existing models, application-centric, centralized, user-centric and federated identity management with their advantages and disadvantages. A focus has been laid on the latter two, which belong to the category of open identity management models. Open identity

management models support the management of identity data in multiple domains and facilitate identity information to be passed seamlessly across organizational borders. The basic idea is to share data between the entities holding identity information (the identity providers) and those consuming it (the relying parties) in a controlled manner. In addition to the conceptual background, technologies to implement user-centric and federated identity management have been described in this article.

Although all these technologies are available today, open identity management systems are still used very rarely. Instead we can observe that the Internet is still an environment of mostly isolated identity islands. As a possibly reason for this situation, we found out that the willingness to rely on information that comes from another than the own security domain is very low. Organizations find it hard to trust on identity management processes they have no insights into. Thus trust into identity assurance processes seams to be a major factor for the success of these new identity models.

In this article, we defined identity assurance and discussed several identity assurance frameworks that aim at providing a standard assessment for identity management processes. We further discussed their limitations. As an area for future research, we showed the benefits of having trust levels for attributes in addition to the trust levels for identity providers that are defined by existing assurance frameworks.

## REFERENCES


Ahn, G.-J., Nam Ko, M., & Shehab, M. (2008). *Portable user-centric identity management,* International Information Security Conference. Boston, MA: Springer.

Audun Jøsang, J. F. (2005). *Trust requirements in identity management,* Newcastle, Australia: Australasian Information Security Workshop 2005.

Ballinger, K., Bissett, B., Box, D., Curbera, F., Ferguson, D., Graham, S., et al. (2006). *Web services metadata exchange* (WS-MetadataExchange). W3C.

Benantar, M. (2006). *Access control systems: Security, identity management and trust models*. Berlin, Germany: Springer.

Bertino, E., Martino, L., Paci, F., & Squicciarini, A. (2009). *Security for Web services and service-oriented architectures*. Berlin, Germany: Springer.

Bertocci, V., Serack, G., & Baker, C. (2007). *Understanding Windows CardSpace: An introduction to the concepts and challenges of digital identities (independent technology guides)* (Vol. 1). Amsterdam, The Netherlands: Addison-Wesley Longman.

Cameron, K. (2005). *Kim Cameron's identity weblog*. Retrieved 2010, from *http://www.identityblog.com/stories/2004/12/09/thelaws.html*

Cameron, K. (2005). *The laws of Identity*. Retrieved July 2010, from *http://www.identityblog.com/stories/2005/05/13/TheLawsOfIdentity.pdf*

Cantor, S., Kemp, J., Maler, E., & Philpott, R. (2005). *Assertions and protocols for the OASIS security assertion markup language (SAML) V2.0. Organization for the Advancement of Structured Information Standards*. OASIS.

Chappell, D. (2010). *Digital identity for. NET applications: A technology overview*. (C. &. Associates, Producer). Retrieved July 2010, from *http://msdn2.microsoft.com/en-us/library/bb882216.aspx*

Cover Pages, O. A. S. I. S. (2008, October). *Microsoft 'Geneva' framework supports SAML 2.0, WS-Federation, and WS-Trust*. Retrieved July 2010, from *http://xml.coverpages.org/ni2008-10-29-a.html*

e-Authentication Initiative. (2007). *E-authentication guidance for federal agencies*. US.

InCommon Federation. (2008). *Identity assurance assessment framework*. Retrieved 2010, from *http://www.incommonfederation.org/docs/assurance/InCIAAF1.0Final.pdf*

Jøsang, A., Fabre, J., Hay, B., Dalziel, J., & Pope, S. (2005). *Trust requirements in identity management* Newcastle, Australia: Australasian Information Security Workshop 2005.

Jøsang, A., Fabre, J., Hay, B., Dalziel, J., & Pope, S. (2005). *Trust requirements in identity management*. In Buyya, R., Coddington, P. D., Montague, P., Safavi-Naini, R., Sheppard, N. P., & Wendelborn, A. L. (Eds.), *ACSW Frontiers*, 44 (pp. 99–108).

Kantara Initiative. (2010). *Website*. Retrieved 2010, from *http://kantarainitiative.org/*

Kylau, U., Thomas, I., Menzel, M., & Meinel, C. (2009). *Trust requirements in identity federation topologies*. International Conference on Advanced Information Networking and Applications (AINA-09). IEEE.

Lockhart, H., Andersen, S., Bohren, J., Sverdlov, Y., Hondo, M., Maruyama, H., et al. (2006, December). *Web services federation language* (WS-Federation), Version 1.1.

Nadalin, A., Goodner, M., Gudgin, M., Barbir, A., & Granqvist, H. (2007). *WS-Trust 1.3*. *Organization for the Advancement of Structured Information Standards*. OASIS.

National Institute of Standards and Technology. (2006). *Electronic authentication guideline*.

OASIS. (2009, July). *Identity metasystem interoperability*, version 1.0. OASIS Standards.

Office of the e-Envoy, UK. (2002). *Registration and authentication - e-Government strategy framework policy and guidelines*. Retrieved from *http://www.cabinetoffice.gov.uk/csia/documents/pdf/RegAndAuthentn0209v3.pdf*

The Bandit Project. (n.d.). *Digital me identity selector*. Retrieved 2010, from *http://code.bandit-project.org/trac/wiki/DigitalMe*

The OpenId Foundation. (2007). *OpenID authentication 2.0 - Final specification*. Retrieved 2010, from *http://openid.net/specs*

Thomas, I., & Meinel, C. (2009). *Enhancing claim-based identity management by adding a credibility level to the notion of claims* International Conference on Services Computing. Bangalore, India: IEEE.

Thomas, I., & Meinel, C. (2010). *An identity provider to manage reliable digital identities for SOA and the web 9th Symposium on Identity and Trust on the Internet*. Gaithersburg, MD: ACM.

Vittorio Bertocci, G. S. (2007). *Understanding Windows CardSpace: An introduction to the concepts and challenges of digital identities (independent technology guides)* (Vol. 1). Amsterdam, The Netherlands: Addison-Wesley Longman.

Windley, P. J. (2005). *Digital identity*. O'Reilly Media.