Dynamic Trust Relationship Establishment in Federated Identity Management

Patricia Arias Cabarcos
Advisor: Florina Almenárez Mendoza

Abstract Federation in identity management has emerged as a key concept for reducing complexity in the companies and offering an improved user experience when accessing services. In this sense, the process of trust establishment is fundamental to allow rapid and seamless interaction between different trust domains. However, the problem of establishing identity federations in dynamic and open environments has not been fully addressed. In this scenarios it is desirable to speed up the processes of service provisioning and deprovisioning. This paper analyzes the underlying trust mechanisms of the existing frameworks for federated identity management and its suitability to be applied in the mentioned environments. After the main limitations are identified, we propose a generic extension for the SAML standard in order to facilitate the creation of federation relationships in a dynamic way between prior unknown parties. Finally, we provide some details regarding implementation issues and present a proof-of-concept application on how to enrich trust decisions by adding reputation data.

1 Introduction

Federation has emerged as a key concept for identity management. Its main goal is to share and distribute attributes and identity information across different trust domains according to certain established policies. The federation model enables users of one domain to securely access resources of another domain seamlessly, and without the need for redundant user login processes. Particularly, the most popular use case is Single Sign On (SSO), which allows users to authenticate at a single site and gain access to multiple sites without providing any additional information. Thus, separating identity management tasks from service provisioning is possible in order to reduce complexity in service providers. So service providers can concentrate on

University Carlos III of Madrid, Telematic Engineering Department, Avda. Universidad 30, 28911 Leganés (Madrid), Spain e-mail: ariasp@it.uc3m.es

their core business and also improve user experience when interacting with various administrative domains.

The main actors in a federation scenario, as depicted in Fig. 1, are:

- *Service Providers (SPs)*, entities which consume identity data, they rely on user authentication made by a third party; SPs are also called *Relying Parties* (RPs).
- *Identity Providers(IdPs)*, entities that assert information about a subject; IdPs are also called *Asserting Parties* (APs). IdPs focus on authentication of users and management of identity information, which can be shared with various SPs.
- *Users*, which interact (usually via a user agent, e.g. web browser) with SPs. They are the subject of the assertions.

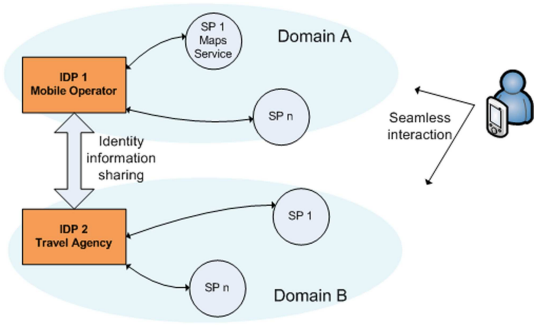

Fig. 1 Identity federation across two different administrative domains

In this example, sharing identity information between the Mobile Operator (IdP1) and the Travel Agency (IdP2) allows Bob to log in only once and gain seamless access to services and applications offered in different domains. By entering the Maps Service web page, a location-based service offered by the Mobile Operator, and present his credentials (e.g. username and password), Bob can follow a link to the Travel Agency web page, and access resources such as accommodation or restaurant information without re-authentication.

Identity federation implies many advantages, including typical use cases like cross-domain SSO, user account provisioning, entitlement management and user attribute exchange. However, current frameworks for identity federation have not been designed taking into account the requirements of dynamic and open environments. In these scenarios, companies should have an easy and agile way to provide services. But establishing relationships between entities is usually hard to scale, because trust must be preconfigured before any interaction between parties takes place.

Trust is a fundamental issue to address scalability in identity management for open dynamic environments. In fact, the flexibility of every federation framework is tied to the underlying trust model, often poorly defined or even out of the specifications scope. Thus, trust between parties involved in a federation process should

be managed in a dynamic fashion, avoiding or minimizing dependence on central administration and reducing preconfiguration needs. As a direct consequence, service provisioning and user interaction would be easier and more flexible, facilitating composition, enrichment and customization.

The remainder of this document is concerned with showing the main challenges in dynamic identity federation and explaining a generic extension to a widely known identity standard in order to overcome these challenges. Section 2 reviews current technologies for identity federation, Section 3 provides a comparative analysis of the trust mechanisms underlying identity federation technologies and Section 4 describes our solution proposal. Next, Section 5 explains some implementation issues and presents a proof-of-concept application to demonstrate the feasibility of our idea. Finally, Section 6 summarizes relevant current work in the field and Section 7 ends with the main conclusions and future work.

2 Background: current Identity Federation frameworks

Identity federation can be accomplished by means of formal Internet standards, or using open source technologies and other openly published specifications. Among the main technologies for identity federation, **Security Assertion Markup Language (SAML)** [13] is the only formal internet standard nowadays. SAML defines an XML based framework to allow the exchange of security assertions between entities. Basically, it specifies four different elements: *Assertions*, which are statements related to authentication, attribute, or authorization about a Principal, issued by an IdP; *Protocols*, which define how and which *Assertions* are requested; *Bindings*, which define the lower-level communication or messaging protocols (such as HTTP or SOAP) that the SAML *Protocols* can be transported over; and *Profiles*, which are combinations of SAML *Protocols* and *Bindings*, together with the structure of *Assertions* to cover specific use-cases. In addition, there is another component, the *Metadata*, that can be used to specify how configuration information is shared between two communicating entities.

Also, closely related to SAML is the **Liberty Alliance** initiative. It was formed with the aim to establish open standards to easily conduct online transactions while protecting the privacy and security of identity information. The Liberty specifications, built on top of SAML, enable identity federation and management through features such as identity/account linkage, single sign on, and simple session management. The Liberty Alliance contributed its federation specification, ID-FF [11], to OASIS, forming the foundation for SAML 2.0, the converged federation specification that Liberty now recognizes.

Another solution to achieve identity federation is **OpenID** [14], which is defined as an open, decentralized, and free framework for user-centric digital identity. It is based on well-known existing internet technologies (URI, HTTP, SSL, Diffie-Hellman), and it is clearly oriented to be used in web scenarios. But OpenID is

mainly an authentication protocol and federation is achieved with extensions, such as [5] that allows some attribute exchange.

Finally, the **WS-Federation** [17] language defines mechanisms for brokering of identity, attribute, authentication and authorization assertions between realms, and privacy of federated claims. The specification is part of the larger Web Services Security framework (WS-*) and describes how to use WS-Trust [19], WS-Security [18] and WS-Policy [2] all together in order to provide federation between security domains.

As will be shown in Section 3, none of the above solutions define a suitable trust model to allow dynamic federation establishment.

3 Underlying Trust Models: a comparative analysis

Comparative studies of different identity management approaches [6] [12] show the main commonalities and differences regarding many different aspects: design centers, terminology, specification set contents and scope, user identifier treatment, security, IdP discovery mechanisms, key agreement approaches, as well as message formats and protocol bindings and trust. Here we aim to take a closer look at trust issues, because trust is the key to address scalability problems in the current identity federation systems. We focus this comparative analysis on the SSO use case because this feature is supported by all the studied identity management technologies.

SAML specifies a primary trust mechanism for a SSO operation, which consists of having a pre-existing trust relationship between the RP and the AP. The trust relationship establishment typically relies on Public Key Infrastructure (PKI) [3] since it is recommended. By using this model, federation implies the aggregation of large lists of providers that agree to use common rules and contracts. The drawbacks of this kind of trust model are well known: hard to deploy and maintain, and high dependence on central authorities.

In the case of OpenId, trust considerations are not addressed in the main specification and SSO can be performed between previously unknown parties without any configuration. However, a new OpenID specification called PAPE (Provider Authentication Policy Extension) [15], has been recently approved in order to enforce trust mechanisms. This extension provides means for a RP to request previously agreed upon authentication policies being applied by the OpenID Provider and for an OpenID Provider to inform an RP what policies were used. With PAPE, OpenID moves from a *trust-all-comers* philosophy to a situation in which the decision to trust can be based in the knowledge of the employed authentication mechanism. In other words, there is no trust model specified by OpenID, RPs must decide for themselves which providers are trustworthy, being their ponsibility to implement any policies related to the OpenID Provider's response. For these reasons OpenID is simple, lighter and more scalable.

The trust topologies considered by WS-Federation and LA resemble PKI trust models between Certification Authorities (CAs). In the case of WS-Federation, IdPs

are equivalent to CAs. In the case of LA, the specification considers two possible contexts, business agreements and authentication, so IdPs and SPs are equivalent to CAs depending on the context. These models are typically implemented by means of trust lists containing trustworthy authorities and, sometimes, maintaining also lists of untrustworthy entities. These lists are manually configured by an administrator. Thus, the establishment of trust relationships is managed with formal contracts specifying policies and restrictions surrounding this relationship.

In WS-Federation, an administrator or other trusted authority may designate that all tokens of a certain type are trusted (e.g. all X.509 tokens from a specific CA). The security token service maintains this as a trust axiom and can communicate this to trust engines to make their own trust decisions.

Liberty bases Identity Federation on the concept of "Circle of Trust" (CoT), which means that entities must establish business and trust agreements in order to enable future interactions. Thus, CoTs defined by LA specify different kinds of trust relationships that can exist between two entities depending on the context. If the context is authentication, we can have direct or indirect trust relationships. On the other hand, a business relationship can be: *pairwise*, when directly links the two entities; *brokered*, when an intermediary (*"broker"*) is required; or *community*, when no relationship of any kind exists. So Liberty entities have a TAL or Trust Anchor List with the trustworthy entities for authentication purposes, and also have a BAL or Business Agreement List, containing those parties which are related to the entity via a business agreement.

Authority lists just allow us to take boolean decisions, which means that if the list contains an entry for an authority or a trustworthy path to reach it, then the decision will be positive. On the other hand, if the authority is unknown and there is no path to it, the decision will be negative. These mechanisms limit interaction in open environments, where the presence of unknown users is common and there is no previous configuration before interaction.

Table 1 Summary of Trust models in Identity Federation

IdM Technology	Trust Model
OpenID	No trust model defined, *trust-all-comers* philosophy, no preconfiguration required.
SAML	PKI recommended. Typically implemented with trust lists.
Liberty Alliance	Trust architecture based on CoTs. Follows SAML recommendation (PKI). Two relationship contexts: authentication, business. Hierarchical, peer-to-peer, mesh and hybrid topologies considered. Typically implemented with trust lists.
WS-Federation	Trust architecture based on WS-Trust. Peer-to-peer, mesh and hybrid topologies considered. Typically implemented with trust lists.

In Table 1 we summarize the main trust features of each identity system. To conclude, all the analyzed technologies typically handle trust management by means of trust lists together with PKI. The only exception is OpenID, which does not require

trust relationships to be established and just follows the *trust-and-accept all-comers* principle. So, it can be noted that none of the above identity management technologies include efficient trust models for dynamic environments, which implies an important challenge. Furthermore, the problem of establishing trust relationships between previously unknown entities willing to interact is not covered by none of the current frameworks or specifications.

4 SAML extension for Dynamic Federation

We think that a first step towards dynamic federation requires an extension of identity management technologies to enrich trust mechanisms. After reviewing the current frameworks for identity federation, we conclude that SAML is the most flexible to add extensions in order to achieve dynamic federation in a generic way. In addition, SAML is the mostly deployed federation solution and has been adopted by many well known providers (e.g. Google Apps). As described in Section 3, all the approaches except OpenID need trust to be preconfigured. In OpenID, despite there is no need for previous configuration, the extension mechanism seems to be rudimentary and less modular. Furthermore, while all the solutions are mainly concerned with web scenarios and the SSO use case, SAML offers abstraction enough to be applied to a wider range of situations [16]. So by including modifications in the abstract level we can assure its later application in more specific use cases. Also, SAML is the only standard nowadays and LA is based in its specifications, so it is more logical to introduce modifications in SAML that could be later adopted by other technologies based on it.

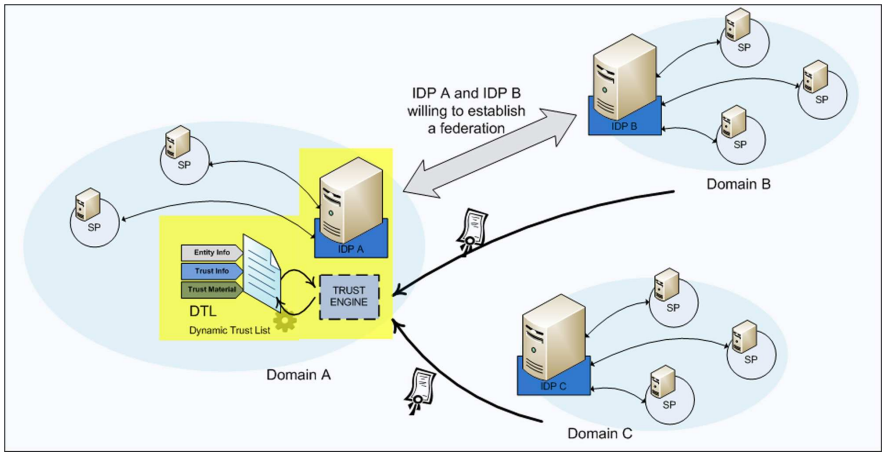

Fig. 2 SAML Extension for Dynamic Trust Establishment in Identity Federation

Therefore, we aim to enhance SAML in order to fullfil some requirements regarding trust establishment that are crucial for dynamic interaction:

- Minimize dependence on central servers or previous configuration, making entities more autonomous and capable of taking trust decisions
- Model trust evolution over time, as it has a clear impact in risk management and trust decisions
- Take advantage of common knowledge, by means of requesting and collecting external information
- Enrich trust mechanisms (not only certificate-based trust)
- Allow seamless interaction between the main actors involved in identity management scenarios

To achieve the above goals we introduce the idea of adding a new component, the **Trust Engine**, which will be in charge of processing every trust-related data to decide if an entity is trustworthy or not. The trust information can be obtained from external or internal sources. The Trust Engine will allow entities to maintain a dynamic store instead of a static list with trust data, that we call **Dynamic Trust List or DTL**. With this philosophy we propose an initial extension model, depicted in Fig. 2. We also define how to use reputation data as external information to enrich trust decisions.

We focus on the point of trust negotiation to establish federations, which means that entity discovery is out of scope. Although the aim of this solution is to be generic enough to be applied to different profiles and use cases, we will focus on analyzing the SSO scenario to start building a prototype (see Section 5).

Next, let us take a deeper look into the main modifications that will be required in SAML entities and specifications in order to understand the improvements and implications.

4.1 Trust engine

In order to include dynamic features in the process of trust establishment, SAML entities must be extended with a Trust Engine. This component is the responsible of processing external and internal trust information and performing DTL updating. Also, decisions to trust will be made by this logical block.

The internal trust information can be obtained from the DTL. Furthermore, other data as internal policies could be useful when applying custom trust levels. e.g. transient or attribute federation may have less requirements than permanent federation as it implies less personal user identity information disclossure.

On the other hand, external trust information can be obtained from other entities. To give an example, information can be gathered from entities belonging to the same domain of the target of federation, or even from entities of different domains. Such entities may have had a previous relationship with the target entity (as shown in Fig. 2). For example, many distributed reputation solutions have been proposed [8]

that can be suitable to implement on top of SAML to allow trust information exchange. Later in this section we will propose a new message format, as well as a protocol, to exchange reputation data over SAML.

The trust engine can be enriched with more complex functionality, e.g. by adding a risk manager or a policy manager block. If we consider timing, analysis of cached trust material, update policies, etc., a better trust management can be achieved. To give an example, using policies to determine when to ask for reputation information offers the capability of implementing different dynamic trust models, in order to select the more appropiated for each situation.

4.2 Dynamic Trust List

In SAML implementations every entity is usually configured with a static TAL before any interaction between parties takes place. This list contains the digital certificates associated to every other entity which is considered trustworthy. Protocol messages whose digital signature cannot be validated against the TAL are rejected. Thus, trust does not evolve over time, because interaction experiences are not taken into account, community knowledge is not exploited, distrust and ignorance are treated in the same way, and the automatic establishment of trust relationships between unknown entities is impossible.

The preconfigured TAL model poses important obvious limitations in dynamic open environments. Instead of a static TAL, the system maintains an enhanced Dynamic Trust List with more complete information: entity data and its associated trust information (e.g. reputation scores, trust level, previous interaction results, etc.) and trust material (e.g. keys, credentials, etc.). The list will be dynamically updated by the Trust Engine under specific events such as receiving recommendations from other entities or when a successful interaction ends. In order to allow the exchange of trust related data, it is required to define new protocols and messages that extend the SAML specifications.

4.3 Reputation Exchange over SAML

According to Josang [8], reputation can be considered as a collective measure of trustworthiness based on the referrals or ratings from members in a community. Thus, this information can be combined with other trust related data, such as the history of past interactions, to take richer trust decisions. However, the application of this dimension of trust to identity management scenarios has not been fully addresed yet. In the following, we propose a SAML extension to take advantage of collecting reputation data.

Adding reputation support to SAML implies modifications to both *Assertions* and *Protocols*. As it was explained before, specifications contemplate three different

kinds of *Assertions*: Authentication, Attribute and Authorization. For the reputation exchange to be possible, we have defined a new one, called *Reputation Assertion*. It has been defined according to the extensions mechanisms explained in the SAML core specification [10], so compatibility is assured. Basically, the defined *Assertion* contains a new custom <Statement> type, called <ReputationStatement>.

On the other hand, the reputation exchange protocol is based on the standard SAML Authentication protocol, so query/response formats are compliant with the rules defined for extending the schema. Thus, the communication flow has two steps: 1.) the entity in the requester role sends a <ReputationRequest>; and 2.) the entity in the responder role returns a <Response> containing a *Reputation Assertion* in case of success, or an error message in case of failure.

```
<Assertion ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
   IssueInstant="2009-09-09T00:46:02Z" Version="2.0"
   xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
   <Issuer>
      Entity1.com
   </Issuer>
   <Subject>
      <NameID
        Format=
        "urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified">
        IdP.domain2.com
      </NameID>
   </Subject>
   <ReputationStatement>
      <Score>
        <ScoreValue>
           6
         </ScoreValue>
      </Score>
      <RepContext>
           "authentication"
      </RepContext>
   </ReputationStatement>
</Assertion>
```

Fig. 3 Reputation Assertion

In Fig.3, we can see the structure of a *Reputation Assertion* conveyed in a response to a Reputation Request. The <Subject> section indicates the identifier of the entity for which reputation data has been requested. Also, it can be seen that the <ReputationStatement> section contains all the reputation related information. For now, this information contains the reputation score and the context, to illustrate for what situation the reputation was made.

5 Dynamic SSO using reputation data: a proof-of-concept

5.1 Conceptual description

As a first approach we have developed a proof-of-concept application to show the viability of the proposed solution. The idea comes from the problem depicted in Fig. 2, where interaction between unknown parties is desired to enable flexible co-operation between providers and quick delivery of services to the users.

Here, we aim to demonstrate that collecting external information allows dynamic trust establishment and facilitates this kind of interactions, otherwise impossible or insecure. For this purpose, we work with the simplest SAML-based SSO scenario: a user, an IdP and a SP. In this situation, if providers are unknown to each other, which usually means that no formal contracts or trust preconfiguration exists, they will not interact.

In the identity management infraestructure used for the experiments, the user must introduce the URI where the metadata document of his IdP is located. This must be done if the providers are unknown. After that, metadata are stored and all the certificates contained in it are considered trustworthy and will be used to validate SAML messages. The main limitation regarding trust issues is that no trust model is defined since the SP will interact with any provider introduced by the user.

Thus, to enable dynamic and secure interaction we have added a new trust logic which modifies the usual operation diagram. The communication flow in this case follows this sequence:

1. Alice accesses a service offered by a SP.
2. SP needs to authenticate Alice, so it performs IdP discovery in order to determine who should be asked for user authentication.
3. SP checks local configuration data to see if the discovered IdP is known (i.e. metadata stored in cache).

   - In case of **known IdP**, SP proceeds with the SSO protocol as usually.
   - In case of **unknown IdP**, the new trust layer executes the logic to gather reputation information about the IdP. If the processing of gathered information results on a positive decision (i.e. trusted entity), then the SP downloads IdP metadata and initiates SSO as usually. On the contrary, the user is informed that her IdP is not considered trustworthy and operation is not possible.

The original workflow and the modified one are depicted in depicted in Fig.4. We can see that the explained proof-of-concept application clearly proves that the core ideas of our proposal are workable. Although the Trust Engine has been integrated within the SP, it could be used in any provider. It can be also seen that the trust logic is easy to integrate as an external API with current open source identity federation toolkits. So, with this test application, the great advantages of enriching trust decisions by exploiting community knowledge can be apreciated. In the following, we will explain the implementation details.

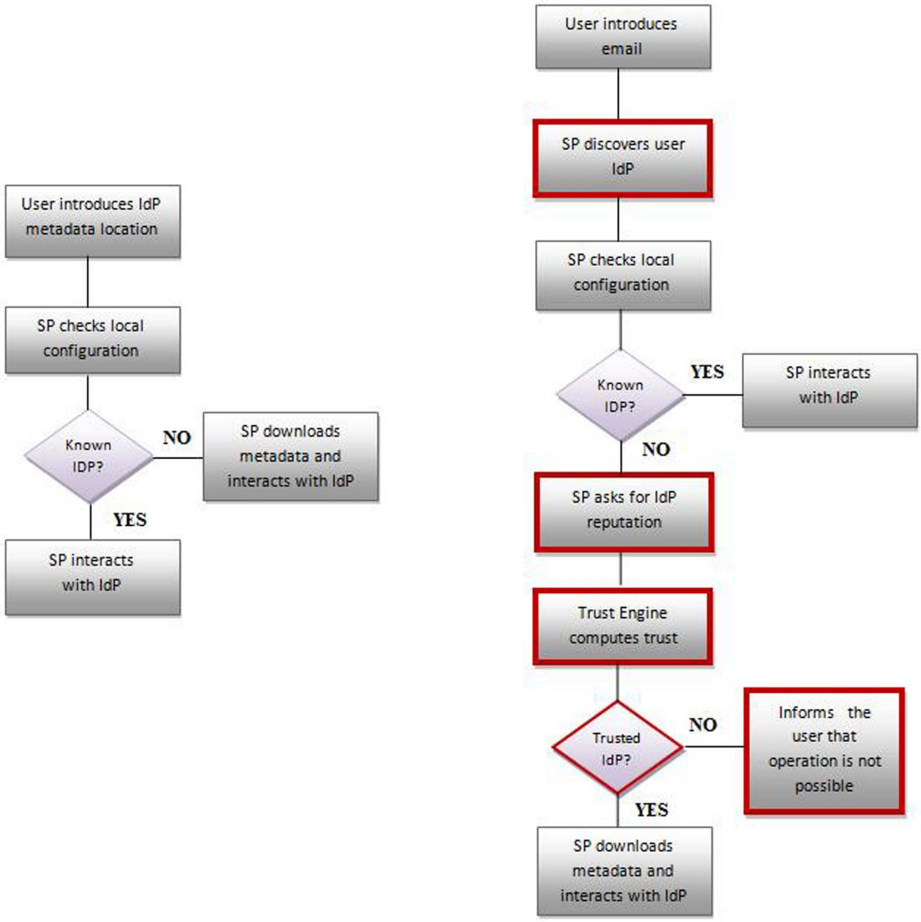

Fig. 4 Left: Original workflow in the SP used in the evaluation. Right: modified workflow in the SP according to the proposed extension.

5.2 Implementation issues

In order to evaluate our proposal, an identity management infrastructure has been deployed. As a first phase, we have chosen ZXID [20], a light open C library that implements the full SAML 2.0 stack, to develop a SP. For deploying the IdP, we are using Authentic [1], which is a Liberty-enabled identity provider based on the lasso

library [9] that also supports SAML 2.0 specifications. These libraries use OpenSSL as underlying cryptographic library and Apache2 as the web server.

Based on the described infraestructure we have introduced the modifications proposed in Section 4. For this purpose, we have partially developed the Trust Engine component, including only the functionality to ask for reputation data. We have also implemented a ReputationRequester and a ReputationResponder, which are software processes that can be added to SAML entities. As its names suggest, the Requester is in charge of asking for reputation data regarding a provider and the Responder is the responsible of answering to reputation requests. Both the format of the *Reputation Assertions* and the exchange *Protocol*, follow the description provided in Section 4.

So, the SP has been modified to use the new trust functionality. The user interface allows to introduce an email, which will be used to derive the IdP name. Then, changing the reputation value associated to this IdP in the database accesed by the ReputationResponder, or adjusting the threshold reputation value in the SP, the provider takes different decisions to cooperate, allways in a dynamic fashion.

It is to mention that the application also simulates the IdP and Metadata discovery processes depicted in diagram of Fig.4. We use a function to derive the IdP name from the user email, but any other discovery service or application could be used since the SAML specification sets themselves are neutral with respect to any particular IDP discovery mechanism.

6 Related Work

As far as the related work is concerned we find several research initiatives in the field. Among others, the following lines are closer to our work:

The Internet2 group, Ping Identity and Stockholm University are working in Distributed Dynamic SAML [7] [4] , a proposal centered around discovery and easiness of configuration. The main important aspects of their contribution are that the partner keys used to sign and validate SAML SSO messages are included in the SAML metadata document, and trust in these keys is derived from the established trust in the metadata document itself. Also, the metadata document must be signed and the X.509 certificate chain used to validate the signature is included in the document. Thus, each partners trust anchor list just contains the root CA certificates. They focus on reducing the manual steps but does not address dynamism in the sense of trust establishment and evolution. So, although the process is lighter and federations are established more rapidly, trust continues to lie in pre-established arrangements, with no evolution over time and entities cannot take autonomous decisions. Furthermore, the proposal is tied to certificate based trust decisions and it is focused on the web SSO profile, but we think that a more general solution is needed that can be applied to a broader range of federation use cases.

Also, Boursas et al. [21] have studied the dynamic management and expansion of Circles of Trust. The main idea is to maintain a repository where trust relation-

ships are stored. This repository is accesed by the providers in order to find a path to unknown entities and derive transitive trust relatioships. Basically, they define algorithms and workflows to allow the establishment of dynamic relationships. However, problems such as the integration of the proposal in identity management scenarios or the exchange of trust information over current federation protocols, are not addresed.

In other recent work [22], the authors identify possible trust patterns in identity federation topologies and perform a risk analysis as the base to discuss the trust requirements of each pattern, which sets an important base for modeling trust in these scenarios.

To summarize, current proposals do not provide a general solution that address how to extend a federation framework independent of the transport protocol or use case. Finally, we aim to address more dimensions of trust, such as risk management, that are not considered in the presented approaches.

7 Conclusions and Future Work

We have reviewed the main current frameworks to achieve identity federation, identifying its main drawbacks to be deployed in dynamic open environments. Underlying trust models are too rigid to allow an agile way of establishing relationships between entities, specially when it comes to interaction with previously unknown parties.

Among all the current approaches, SAML offers the required flexibility, abstraction and modularity to be extended for its application in dynamic open environments. Thus, we present a SAML extension, which allows not only certificate based trust but permits to take richer decisions based on different trust dimensions such as reputation or history of past interactions.

For now, we have demonstrated the feasibility of the presented ideas. This proof-of-concept has served to face some important challenges posed by the proposal, namely integration with identity frameworks and modification of the SAML standard. From this starting point, we identify the following tasks as our future working lines: add components to model risk and trust evolution, complete the definition of the new *Reputation Assertions* and *Protocols*, define DTL structure and its associated management tasks, study how to define contexts for reputation and perform evaluation tests in more complex scenarios.

References

1. Authentic: Liberty-compliant Identity Provider.http://authentic.labs.libre-entreprise.org/.
2. Bajaj, S., Box, D., Chappell, D., Curbera, F., Daniels, G., Hallam-Baker, P., Hondo, M., Kaler, C., Langworthy, D., Nadalin, A., Nagaratnam, N., Prafullchandra, H., Riegen, C., Roth, D., Schlimmer, J.(ed.), Sharp, C., Shewchuk, J., Vedamuthu, A., Yalinalp, ., Orchard, D.: Web Services Policy 1.2 - Framework (WS-Policy), W3C Member Submission, April 2006.
3. Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk., W.: Internet X.509 Public Key Infrastructure: Certificate and Certificate Revocation List (CRL) Profile. IETF Network Working Group, RFC 5280, 2008.http://www.ietf.org/rfc/rfc5280.txt
4. Harding, P., Johansson, L. and Klingenstein, N.: Dynamic Security Assertion Markup Language: Simplifying Single Sign-On. In: IEEE Security & Privacy, 6(2):83-85, March/April 2008.
5. Hardt, D., Bufu, J. and Hoyt, J.: OpenID Attribute Exchange 1.0.http://www.openid.net.
6. Hodges, J.: Technical Comparison: OpenID and SAML - Draft 06. January, 2008.
7. Internet2.: Distributed Dynamic SAML, October 2007.https://spaces.internet2.edu/display/dsaml/
8. Josang, A.,Ismail, R. and Boyd, C.: A survey of trust and reputation systems for online service provision. In: Decis. Support Syst.,43(2):618-644, 2007.
9. Lasso, Liberty Alliance Single Sign-On. Availabe at http://lasso.entrouvert.org/
10. Cantor, S., Kemp, J., Philpott, R. Maler, E.: Assertions and Protocols for the OASIS Security Assertion Markup Language(SAML) V2.0. OASIS Standard, March 2005.
11. LA.: Liberty ID-FF Protocols and Schema Specification.http://www.projectliberty.org
12. Maler, E. and Reed, D.: Options and Issues in Federated Identity Management, IEEE Security & Privacy, 6(2):16-23, March/April 2008.
13. Ragouzis, N., Hughes, J., Philpott, R., Maler, E., Madsen, P., Scavo, T.(eds).: Security Assertion Markup Language (SAML) V.2.0 Technical Overview. OASIS Comittee Draft 02. March, 2008.
14. OpenID.: OpenID Authentication 2.0. DEcember 2007. http://www.openid.net
15. Recordon, D., Jones, M., Bufu, J., Daugherty, J. and Sakimura, N.: OpenID Provider Authentication Policy Extension 1.0.http://www.openid.net,
16. Tschofenig, H., Hodges, J., Peterson, J., Polk, J and Sicker, D.: SIP SAML Profile and Binding, draft-ietf-sip-saml-06.txt, IETF Internet-Draft.
17. WS-Federation.: Web Services Federation Language version 1.1. December, 2006.
18. Nadalin, A., Kaler, C., Monzillo, R. and Hallam-Baker, P. (eds.).: Web Services Security: SOAP Message Security 1.1 (WS-Security 2004), OASIS Standard Specification. February 2006.
19. Nadalin, A., Goodner, M., Gudgin, M., Barbir, A. and Granqvist, H.: WS-Trust 1.3, OASIS Standard. March 2007.
20. SymLabs.: ZXID: Open SAML implementation in C. http://www.zxid.org
21. Boursas, L. and Danciu, V.A.: Dynamic inter-organizational cooperation setup in Circle-of-Trust environments. In: Network Operations and Management Symposium, 2008. NOMS 2008.
22. Kylau, U., Thomas, I., Menzel, M. and Meinel, C.: Trust Requirements in Identity Federation Topologies. In: Proceedings of the 2009 International Conference on Advanced Information Networking and Applications, AINA'09.