Enabling SAML for Dynamic Identity Federation Management

Patricia Arias Cabarcos[1], Florina Almenárez Mendoza[1], Andrés Marín López[1], Daniel Díaz Sanchez[1], P. Arias[1] et al.

University Carlos III of Madrid, Telematic Engineering Department, Avda. Universidad 30, 28911 Leganés (Madrid), Spain {ariasp, florina, amarin, dds}@it.uc3m.es

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 that form part of Next Generation Networks (NGNs), where it is desirable to speed up the processes of service provisioning and deprovisioning, has not been fully addressed. 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. This analysis is mainly focused on the Single Sign On (SSO) profile. 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 give some details of implementation and compatibility issues.

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 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.

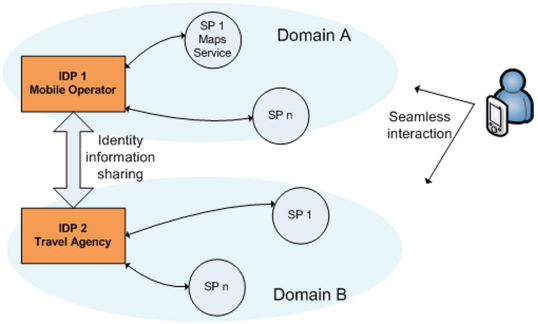

Fig. 1. Identity federation across two different administrative domains

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 services 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.

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 and the related work, Section 3 provides a comparative analysis of the trust mechanisms underlying these technologies and Section 4 describes our solution proposal. Finally, Section 5 explains some implementation issues and Section 6 contains conclusions and future work.

2 Background

Identity federation can be accomplished by means of formal Internet standards, such as the OASIS SAML specification [15], or using open source technologies and other openly published specifications, like the Liberty Alliance Identity Federation Framework (ID-FF) [13], Shibboleth [9], OpenID [17] or WS-Federation [21]. Next, the main technologies for identity federation are introduced with the aim to provide a background knowledge to the reader. Also, we include a summary of the current related work regarding dynamic identity federation.

2.1 Reviewing the current solutions for identity federation

- **Security Assertion Markup Language (SAML)** defines an XML based framework to allow the exchange of security assertions between entities. Basically, SAML 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.
  SAML is highly flexible, in fact, all of the above components can be extended. Furthermore, there is another extension point: the *Metadata*. The *Metadata* can be used to specify how configuration information is shared between two communicating entities. For instance, these data can include an entity's support for given SAML *Bindings*, identifier information, and Public Key Infrastructure (PKI) [5] information.
- **OpenID** 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.
  A major feature of OpenID is user-centricity, which means that users can decide which IdP they trust the most to authenticate them. In fact, users can also become their own IdP without the need of registration or authorization from a third party. Thus, the OpenID protocol does not rely on a central authority to authenticate a user's identity.
  But OpenID is mainly an authentication protocol and federation is achieved with extensions, such as [7] that allows some attribute exchange. Thus, the

OpenID specification is rigidly defined and it only covers a narrow range of Web SSO use cases. Despite having great benefits, such simplicity and no preconfiguration requirements before interactions, the main limitations are that trust, security and privacy are still not thoroughly examined.

- **Liberty Alliance (LA)** 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, to OASIS, forming the foundation for SAML 2.0, the converged federation specification that Liberty now recognizes.
- **Shibboleth** is a project of the Internet2 Middleware Initiative that has developed an architecture and open-source implementation for identity federation based on SAML specifications. It is mainly intended to be used in educational environments (e.g. to help students and faculty in accessing online shared resources from federated Universities). The Shibboleth software provides a federated single sign-on and attribute exchange framework, also including extended privacy functionality and allowing the user's browser and their home site to control the attributes released to each application.
- **WS-Federation** defines mechanisms to allow different security realms to federate, such that authorized access to resources managed in one realm can be provided to security principals whose identities and attributes are managed in other realms. This includes 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 [23], WS-Security [22] and WS-Policy [3] all together in order to provide federation between security domains.

2.2 Related Work

As will be shown in the next section, none of the above solutions define a suitable trust model to allow dynamic federation establishment. In this sense, there are various parties working with SAML to allow easy deployment and minimize mutual beforehand configuration steps.

The Internet2 group, Ping Identity and Stockholm University are working in Distributed Dynamic SAML [10] [6] to deal with challenges regarding deployment, scalability and interoperability. They aim to achieve: 1) distribution, in the sense of changing the operations of typical multi-party SAML federations to be less dependent on central administration; and 2) dynamism, which implies various means to support discovery and autoconfiguration instead of static prearrangement between parties. Thus, the group is developing proposals [19] [4] to be promoted in various communities, including potential submissions to the OASIS Security Services Technical Committee for consideration as standards.

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.

But the idea of relying on signed metadata is quite similar to just relying on X.509. There are only two ways to establish trust in the metadata signatures: based on metadata signing certificates together with a traditional PKI or using out-of-band certificates as a form of pre-shared keys for signature validation. The proposal is focused on reducing the manual steps but it does not address dynamism in the sense of trust establishment and evolution. 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 without some preconfigured information. 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.

3 Underlying Trust Models: a comparative analysis

Comparative studies of different identity management approaches [8] [14] 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.

As it was mentioned in section 2, SAML specifies a primary trust mechanism for a SSO operation. It consists of having a pre-existing trust relationship between the RP and the AP. The trust relationship establishment typically relies on PKI since it is recommended. Shibboleth adopts this same model so that 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) [18], 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 responsibility 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 Kerberos tokens from a specific realm or 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. This mechanisms limit interaction in open environments, where the presence of unknown users is common and there is no previous configuration before interaction.

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 en-

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.
Shibboleth	Follows SAML recommendation (PKI). 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.

vironments, 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

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. Our goal is to design and incorporate dynamic trust models in order to facilitate the interaction between different actors involved in an identity management system.

A popular approach to addressing security challenges in these environments is the use of distributed trust mechanisms. By analyzing previously gathered information, such as certificates or reputation scores, a trust decision can be made to interact with other entities in the system with some assurance.

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. 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 [20]. Also, SAML is the only standard nowadays and LA and Shibboleth are 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. But SAML-based federations have challenges scaling

IdP and metadata discovery, protocol binding choice, attribute and nameID usage, key management and trust establishment.

Therefore, we propose a SAML extension that allows entities to include external trust information but still maintains compatibility with the existing trust mechanisms that are mostly employed today. The main benefits of this extension are:

- 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 reputation information
- Enrich trust mechanisms (not only certificate-based trust but also reputation-based decision making)

To achieve the above goals we consider that SAML should be extended to allow the collection of reputation information, which means defining an XML representation of reputation data to be included in the Assertions or in the Metadata, and also describing an exchange protocol for requesting and sending reputation messages.

SAML is defined in a modular way, by including modifications in the abstract level we can assure its later application in more specific use cases. With this philosophy we propose an initial extension model, depicted in Fig. 2.

We present a generic solution, consisting of a SAML extension, as a first step towards dynamic federation. We focus on the point of trust negotiation before establishing a federation, 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).

Basically we introduce the idea of SAML entities maintaining a dynamic store instead of a static list with trust information. The new Dynamic Trust List (DTL) is automatically updated according to the establishment and evolution of trust relationships. This solution implies modifications to allow gathering of external trust information and also new functionality must be added to process these data and manage the DTL. Next, all the new concepts and implications are detailed.

4.1 Dynamic Trust List

In SAML implementations every entity is usually configured within a 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 decisions are just boolean. 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 under specific events such as receiving recommendations from other entities or when a successful interaction ends. In order to allow secure exchange of trust material, it is required to define key agreement protocols (e.g. by adding Diffie-Hellman) because these mechanisms are not included in none of the SAML specifications.

4.2 Trust engine

In order to include the previously described dynamic features in the process of trust establishment, SAML entities must be also extended with a trust engine. This component is the responsible for 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). In this field, many distributed reputation solutions have been proposed [11] that can be suitable to implement in top of SAML to allow trust information exchange.

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.

5 Implementation issues

In order to evaluate our proposal, we have deployed our own identity management infrastructure. As a first phase, we have chosen ZXID [24], a light open C library that implements the full SAML 2.0 stack, to set up a test scenario. So we have developed a SP under ZXID. For deploying the IdP, we are using Authentic [2], because ZXID does not include an IdP implementation. Authentic is a

Quixote application. Quixote is a framework for developing web applications in Python. Authentic is a Liberty-enabled identity provider based on the lasso library [12] that also supports SAML 2.0 metadata. These libraries use OpenSSL as underlying cryptographic library and Apache2 as the web server.

The configuration of the mutual trust relationship between the SP and the IdP has required some slight changes in the SAML metadata generated by them. In the ZXID libraries, we had to remove the binding `POST SIMPLE SIGN` because it is a draft binding that has not been implemented by the lasso yet. Regarding the IdP metadata, we had to add the inclusion of a complete x509 certificate instead of the public key like the SAML specification defines.

After the successful integration between SP and IdP, we have tested as the SSO and Single Logout (SLO) of a user between these two providers is possible. The user's identity information only contains a user and password, but this must be extended to include other kind of credentials. For this, the support of data managers by ZXID has to be improved, because so far ZXID only supports text files. The database or LDAP support is a future work, as reflected in the whole test scenario architecture (Fig. 3).

In the Fig. 3 we can also see that for making authorization decision SAML is based on XACML [16], which has not been implemented yet. We have our own XACML-compliant authorization system [1], which has been successfully deployed and extended for taking into account trust information in the access control policies. So we are going to integrate our SP with such system to grant or to deny permissions

Now, we are firstly including dynamic trust lists (DTLs) and are also developing services in different administrative domains in order to test more complex scenarios, interoperability between federation domains, management of the user's identity information, etc. In this way, we could test our SAML extensions and evaluate their performance.

Acknowledgement. The authors want to thank Rosa Sánchez for her work on the deployment of the SAML identity management infrastructure.

6 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 decisions but also reputation based decisions and permits the inclusion of reputation information in order to take richer trust decisions.

Now, we are implementing the proposed SAML extension, which require to define a XML syntax to express reputation information, to describe a mechanism

of reputation information request/response, to enhance SAML entities with a trust engine capable of dealing with dynamic trust lists, and finally to perform evaluation experiments in a federation scenario.

References

1. Almenárez, F., Marín, A., Campo, C. and García, C.: TrustAC: Trust-Based Access Control for Pervasive Devices. In: 2nd International Conference Security in Pervasive Computing (SPC'05).
2. Authentic: Liberty-compliant Identity Provider.http://authentic.labs.libre-entreprise.org/.
3. 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.
4. Cantor, S. (ed): SAML V2.0 Metadata Interoperability Profile, Working Draft 01, 2008.
5. 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
6. 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.
7. Hardt, D., Bufu, J. and Hoyt, J.: OpenID Attribute Exchange 1.0.http://www.openid.net.
8. Hodges, J.: Technical Comparison: OpenID and SAML - Draft 06. January, 2008.
9. Internet2.: Shibboleth Architecture. http://shibboleth.internet2.edu
10. Internet2.: Distributed Dynamic SAML, October 2007. https://spaces.internet2.edu/display/dsaml/
11. 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.
12. Lasso, Liberty Alliance Single Sign-On. Availabe at http://lasso.entrouvert.org/
13. LA.: Liberty ID-FF Protocols and Schema Specification.http://www.projectliberty.org
14. Maler, E. and Reed, D.: Options and Issues in Federated Identity Management, IEEE Security & Privacy, 6(2):16-23, March/April 2008.
15. 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.
16. OASIS.:eXtensible Access Control Markup Language (XACML) (2003). http://www.oasis-open.org/apps/org/workgroup/xacml/
17. OpenID.: OpenID Authentication 2.0. DEcember 2007. http://www.openid.net
18. Recordon, D., Jones, M., Bufu, J., Daugherty, J. and Sakimura, N.: OpenID Provider Authentication Policy Extension 1.0.http://www.openid.net,
19. Stockholms Universitet, A profile for distributed SAML metadata management, Distributed Dynamic SAML version 7 (Gruesome Gorilla), October 2007.

20. 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.
21. WS-Federation.: Web Services Federation Language version 1.1. December, 2006.
22. 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.
23. Nadalin, A., Goodner, M., Gudgin, M., Barbir, A. and Granqvist, H.: WS-Trust 1.3, OASIS Standard. March 2007.
24. SymLabs.: ZXID: Open SAML implementation in C. http://www.zxid.org

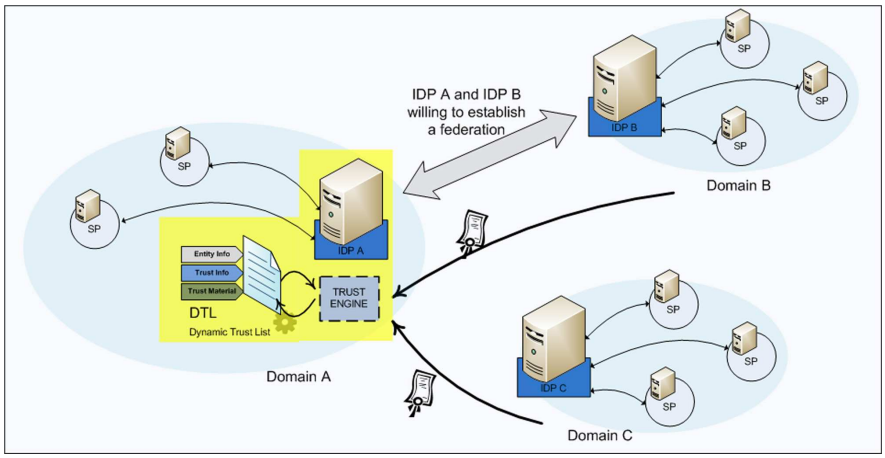

Fig. 2. SAML Extension for Dynamic Trust Establishment in Identity Federation

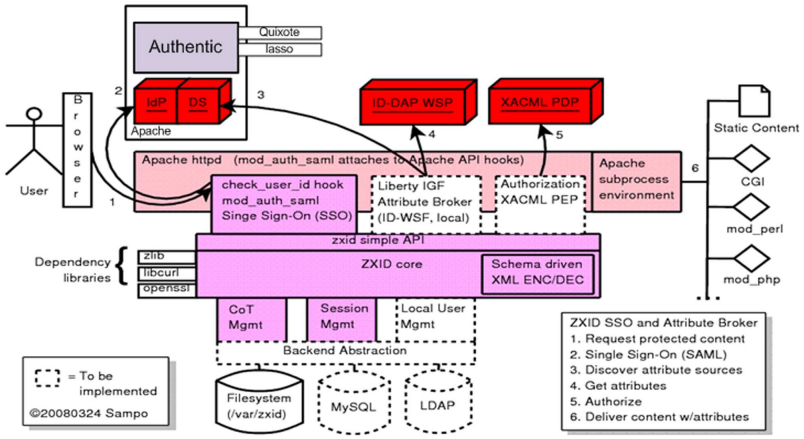

Fig. 3. SSO Test Scenario Architecture