A Comprehensive Comparison of Trust Management Systems for Federation

Zhengping Wu

Department of Computer Science and Engineering,
University of Bridgeport,
221 University Avenue,
Bridgeport, CT 06604, USA
zhengpiw@bridgeport.edu

Abstract. Federation becomes pervasive in information sharing and collaborations over distributed systems, ubiquitous systems, and the Internet. Trust management plays a critical role to smooth collaborations and information sharing across different trust domains. The federation of trust management is a new direction for these networked systems. In this paper, the requirements and a set of evaluation metrics for federated trust management systems are briefly examined, and then a comprehensive comparison of extant trust management systems is made against these metrics. The purpose of this paper is not to provide an ultimate comparison covering all necessary features; instead, its purpose is to initiate a discussion and to offer a context in which to evaluate current and future solutions, in order to encourage the development of proper models and systems for federated trust management.

Keywords: trust management system, federation, comprehensive comparison, evaluation metrics.

1 Introduction

Demand for management of trust in networked systems introduces more complexity than ever when federation activities become pervasive. Many existing research efforts have focused on different aspects of trust management, and they use different criteria and metrics to evaluate their efforts. These criteria and metrics are often applicable only to the researchers' own systems. This paper briefly examines general requirements for federated trust management and a number of existing systems from different aspects, and uses a set of evaluation metrics to comprehensively compare these trust management systems. Using this set of metrics, a system's merits and weaknesses can be more clearly documented for system users and researchers. To illustrate the applicability, the metrics are applied to the comparisons between trust management systems from leading technology companies such as IBM, AT&T, and other extant trust management systems from the research community. However, the metrics introduced in this paper are not ultimate results; rather, they are intended to initiate a broader discussion of what is needed, and to offer a context in which to evaluate current and future solutions and to inspire new directions for federated trust management.

E. Bertino and J.B.D. Joshi (Eds.): CollaborateCom 2008, LNICST 10, pp. 402–415, 2009.

2 Requirements of Federated Trust Management

Ruohomma and Kutvonen [1] discuss the requirements of a trust management system from the viewpoint of trust's life cycle, which include initialization of a trust relationship, observation of the relationship, and actions caused by the relationship. But that is not enough. Wu and Weaver [2] discussed a set of more comprehensive requirements for federated trust management systems. Following this discussion, this section briefly examines the requirements of trust management systems from four different aspects from both fundamental and practical viewpoints. Fulfillment of these requirements will not only provide solid support of the functionalities for trust management proposed by Ruohomma and Kutvonen, but also support new requirements unique from federation activities and federated trust management. Trust representation, trust exchange, trust establishment, and trust enforcement are the four crucial aspects of which a federated trust management system needs to take care.

Trust representation needs a collection of languages to express various factors in federated trust management, plus a set of protocols to make different languages interoperable. Expression of trust facts such as credentials, trust intentions and trust behaviors [3], which describe the identities and their attributes, willingness to act or willingness to accept actions, and the externally observable properties of the actions themselves, is necessary in trust representation. All existing and future authentication technologies need to be supported, and all types of policies describing all possible trust intentions need to be accommodated in trust representation. Meanwhile interoperable protocols need to be provided for all possible expression formats across trust domains.

Trust exchange needs a secure method to communicate trust representations across trust domains. All the communication protocols in the network layers should be supported. And trust exchange is expected to offer not only communication channel security, but also message security. At the same time, users will expect trust exchange to guarantee not only end-to-end security, but also the integrity and privacy of the exchanged information. Although formalization and standardization is desired, no single format can express all types of factors in trust. There is neither a standard syntax nor distribution mechanism by which an authority can make trust-related information available for consumption by all potential relying parties. So trust exchange needs to provide a content interpretation service to translate trust information between different policy languages and message or token formats, and make the final action descriptions easy to understand and enforce.

Trust establishment provides a dynamic and flexible infrastructure to establish and maintain trust relationships across trust domains. Negotiation is a necessary process before trust relationships can be established. Trust establishment needs to support a set of general trust negotiation protocols that permit involved parties to establish and maintain trust relationships, as well as application-specific and content-triggered negotiation protocols. At the same time, privacy control mechanisms are needed in trust establishment to protect negotiating parties' privacy.

Trust enforcement includes a dynamic and flexible infrastructure to publish/discover trust intentions and trust behavior descriptions, and authorize trust behaviors. Trust intentions need to be known and understood by involved parties. Providing a way to publish these trust intentions and making these trust intentions

accessible to involved parties or the public are necessary. The trust enforcement infrastructure needs a mechanism to make all trust behaviors consistent with trust intentions. Because trust intentions change from time to time, the trust enforcement infrastructure needs to provide a mechanism to prescribe trust intentions and check the compliance of trust intentions dynamically. In human relationships trust intentions can be uncertain; sometimes they are just derived from feelings. The flexibility to handle this uncertainty is also necessary.

3 Examination of Extant Systems

Examining these requirements of six extant trust management systems from major vendors and international organizations can clearly show the need of an evaluation metrics for federated trust management.

- The IBM Trust Establishment Framework [4, 5] is a toolkit for enabling trust relationships between strangers, together with a set of trust enforcement mechanisms and a corresponding representation for policies. Although trust relationships are established in this system, the relationships are merely based on public key certificates, and are manually set by domain administrators. It lacks the capability for on-line negotiation. Its trust enforcement mechanisms are also based on public key certificates. They decide users' roles based on their certificates and certain policies. Enforcement decisions are based on validating certificates and mapping certificates' owners to roles. To represent trust-related information, the system only provides a trust policy language to describe rules that determine how to map entities to roles. IBM did not provide further information about its trust exchange mechanism because that was not the focus of the system.
- The AT&T PolicyMaker system [6] integrates trust enforcement and representation in a composite way. The enforcement engine uses certificates only to authorize the holder of the certificate to perform certain actions. It is not convenient to include certain user's attributes in these certificates. The corresponding control policy is expressed in a set of assertions, which can include programs provided by the resource server that are executed as part of compliance checking when a request is made, which makes this system unique. The KeyNote system [7] is a simplified version of the PolicyMaker system with certain extensions for a variety of Internet-based applications. The trust representation in the KeyNote system is a single, unified language for local policies and credentials. These policies and credentials contain predicates that describe the trusted actions permitted by the holders of specific public keys, which are called assertions. The trust enforcement becomes natural when these assertions are essentially small, highly structured programs. Credential assertions, which also serve the role of certificates, have the same syntax as policy assertions but are also signed by the principal delegating the trust. According to their specifications, the trust relationships implied in these systems need to be manually set by administrators. Thus both systems lack negotiation capability.

Exchange mechanisms are not described, but are assumed in the underlying infrastructures.

- The REFEREE system [8] is a rule-controlled environment and provides both a general policy evaluation mechanism for web clients and servers and a language for specifying trust policies. In the REFEREE model, trust enforcement is totally under the control of trust representation. REFEREE places all trust decisions under explicit policy control. Every action, including evaluation of compliance with policy, happens under the control of certain policies. That is, REFEREE is a system for writing policies about policies, as well as policies about cryptographic keys, certification authorities, trust delegation, or anything else. Establishment of trust relationships is ignored in this system, so it is not a complete federated trust management system. As with the PolicyMaker and KeyNote systems, exchange mechanisms are assumed in the underlying infrastructures.
- The Liberty Alliance is a digital identity standards group. The Liberty Alliance Project [9] is a consortium of technology vendors and consumer-facing enterprises formed "to establish an open standard for federated network identity." It aims to make it easier for consumers to access networked services from multiple suppliers while safeguarding security and privacy. It does not explicitly use trust management methods, but its specifications are closely linked to the SAML (Security Assertion Markup Language [10]) single sign-on standard, and they overlap with certain elements of WS-Security. So the Liberty Alliance has its own representation and enforcement requirement for trust-related activities, and mechanisms for establishment and exchange of trust relationships. Its specifications have been published in three phases: the Identity Federation Framework (ID-FF) came first in early 2003, the Identity Web Services Framework (ID-WSF) followed in late 2003, and the Identity Services Interface Specifications (ID-SIS) document was finalized in late 2004.
- Another interesting framework is GAA-API/TrustBuilder [11]. It is an integrated framework with two subsystems: Generic Authorization and Access-control API (GAA-API) [12] and TrustBuilder [13]. Although it uses adaptive trust negotiation and access control as a means to counter malicious attacks and does not offer critical features such as maintenance and updating of trust relationships, it does provide some basic functionalities for federated trust management. TrustBuilder provides functionalities for trust establishment, while GAA-API provides an enforcement engine to control compliance between a user's intentions (represented in policy files) and a system's behaviors. Various trust related information is represented in different formats such as policy files, databases, and credentials. Privacy protection for exchange of trust related information is also addressed. The most important feature of this framework is that it uses trust as its basis to protect security and privacy, and that is the main task of federated trust management systems.

Clearly, an evaluation metrics can be used to reveal the strengths and weaknesses as well as to help users choose a suitable solution from different extant trust management systems.

4 Characteristics of Federated Trust Management

Some characteristics exist in all trust management systems. These characteristics are categorized and summarized from three perspectives in order to offer a set of overall measurements for evaluation purpose.

4.1 Quality of Functionality

Quality of functionality is the most important set of measurements to be considered for evaluation, because it includes the most desirable qualities a user expects from a federated trust management system. Three basic qualities of functionality required for federated trust management are discussed below.

(1) Adaptability. Federated trust management systems need to accommodate the dynamic trustworthiness characteristics of trusted partners' behavior, who might suddenly lose competence or maliciously employ strategies to vary trustworthiness [15]. Federated trust management systems must update themselves frequently to accommodate dynamic trust relationships and evolutionary policies.
(2) Accuracy. Federated trust systems need to include suitable trust factors to correctly model and predict potential partners' future behavior and enforce their behavior. Accuracy of federated trust management systems can be measured in terms of the similarity between the system's calculated trust model or value and the trusted partner's true trustworthiness [14], or in terms of the consistency between a system's behaviors and its users' expectations.
(3) Reputation. Reputation is a measurement and indicator of an entity's (person, service or system) trustworthiness. It mainly depends on partners' experiences of cooperation with that person, using that service, or interaction with that system. Different partners may have different opinions regarding the same entity. The value of a reputation is defined as a comprehensive combination of rankings given to the entity by partners, and that value is the only representative factor of the trustworthiness of that entity.

4.2 Cost of Functionality

All federated trust management systems try to build trust relationships, construct trust models, exchange trust related information, and enforce trust behavior with minimal computational cost and time [16]. Computational efficiency can be gauged by the time needed to complete a specific task, so the time needed for a specific federation task is a useful metric.

(1) Duration. Execution duration measures the expected delay between the moment when a request for a certain trust management operation is sent and the moment when the operation is completed. The execution duration is the sum of the processing time and the transmission time over multiple domains. Execution time can be obtained via active monitoring.
(2) Cost. Cost is something (e.g., money, CPU cycles, time) that users have to pay for the system to fulfill their needs to manage trust across security domains. Users may need management of trust for online collaboration or resource

sharing such as checking credit or determining the amount of money the service requester has to pay to the service provider to get a commodity such as an entertainment ticket or monthly phone service. Providers of trust management either directly advertise the execution price of their services, or they provide means for potential requesters to inquire about it.

4.3 Usability

Usability of federated trust management systems is mainly reflected in user awareness. After a user has deployed a trust management system, operation should be autonomous unless the system actively requires a user's involvement. These two aspects are explained in detail. Other factors such as understandability, operability, user involvement, and user's acceptance will be discussed in the actual comprehensive comparison of existing systems.

(1) Indicator. Like other security and privacy protection systems that provide a security alert icon (e.g., the SSL icon in web browsers), federated management systems need to provide an indicator of the trustworthiness of a cooperating partner or potential partner in order to explicitly indicate trust and risk.

(2) User transparency. Since federated trust management handles the non-functional aspects of a system, the workflow and processing of the trust management system should be user-transparent. But when user involvement is needed, the system needs to provide good alert mechanisms to get the user's attention, and employ a good user interface for human machine interactions.

5 Comprehensive Comparisons of Extant Systems

Following the four functional requirement aspects and three important evaluation perspectives discussed in section 2 and 4, a set of comprehensive evaluation metrics will be discussed and used to compare the six major trust management systems, and especially for their support of federation activities. This set of evaluation metrics is divided into four groups and examined from three evaluation perspectives. After this comparison, strength and weakness of each system can then be analyzed. And direction for further improvement of each system can be identified. The evaluation data are collected from a number of server systems. Each system runs on a Xeon 2.80GHz processor, 512MB memory, and the Windows XP professional version.

5.1 Trust Exchange

Since the exchange of trust related information utilizes an underlining communication infrastructure and facilities, evaluation of functionality for trust exchange is focused on other aspects. The most important aspect of trust exchange is secure and reliable transmission of trust related information at the message level. Thus the security token is the cornerstone. Broadly speaking, a number of types of security tokens have been proposed and implemented using different trust models. These trust models include the X.509 standard Public Key Infrastructure (PKI) trust model, the Pretty Good

Privacy (PGP) trust model, the Simple Public Key Infrastructure (SPKI), and the Simple Distributed Secure Infrastructure (SDSI). The capability to accommodate extant security tokens and future token types is an important aspect for any federated trust management system. To measure this aspect, expert judgment needs to be used to answer whether a system's trust exchange facility has an open architecture and to what degree. It is also required to count the number of extant security token types supported and answer "which token types are they"?

Another important aspect for trust exchange is the capability to resolve semantic conflicts. Extant research in information semantic interoperability can be categorized into three types: mapping-based, intermediary-based, and query-oriented approaches. Answering the following questions can measure the semantic interoperability for trust exchange and the measurement can be constructed accordingly.

- Are mapping-based interoperation methods supported?
- Are intermediary-based interoperation methods supported?
- Are query-oriented interoperation methods supported?

The performance measurement is mainly for the overheads of token exchange, semantic interoperation service, and privacy protection.

Ideally, trust exchange is completely automated, but sometimes this is not possible. User involvement refers to whether or not the user is actively engaged in the trust exchange process. A comparison of six major systems' exchange capability is illustrated in table 1. The "yes/no" answers the question of whether a token type or a semantic

Table 1. Comparison of six major systems for trust exchange

Trust exchange		Security token exchange				Semantic interoperation		
		Open architecture	Token type 1	Token type 2	Token type 3	Mapping-based method	Intermediary-based method	Query-based method
Quality of functionality	IBM TE	No	PK Cert.	N/A	N/A	Yes	No	No
	PolicyMaker	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	KeyNote	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	REFEREE	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Liberty	Yes	PK Cert.	X.509	SAML	Yes	No	No
	TrustBuilder	No	PK Cert.	X.509	N/A	Yes	No	Yes
Cost of functionality (overhead)	IBM TE	N/A	1ms	N/A	N/A	N/A	N/A	N/A
	PolicyMaker	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	KeyNote	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	REFEREE	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	Liberty	N/A	3ms	5ms	N/A	N/A	N/A	N/A
	TrustBuilder	N/A	2ms	4ms	N/A	N/A	N/A	N/A
User involvement	IBM TE	No				No		
	PolicyMaker	N/A				N/A		
	KeyNote	N/A				N/A		
	REFEREE	N/A				N/A		
	Liberty	No				No		
	TrustBuilder	No				Yes (when needed)		

method is supported. Overheads of token exchange and interoperation are measured on experiment servers running exchange services within a local network. Liberty Alliance Project can support the most types of tokens, whereas PolicyMaker, KeyNote and REFEREE do not use tokens at all. The overhead for token exchange is almost negligible for each system discussed here if there is any.

5.2 Trust Representation

As for the functionality of trust representation, completeness is the first aspect to be considered; next is the accuracy measurement of trust representation. Because there are diverse trust factors involved in federated trust management, systems' capabilities to accommodate these diverse trust factors are also important. These trust factors can be classified into objective factors and subjective factors. Objective factors include various facts and other objectively measurable elements in federated trust management. Objective factors can be classified into three categories: identity information, privilege information and trust knowledge. Identity information includes a representative element (identification number or name) and other credential information. Privilege information describes allowable actions and behaviors in a system. Trust knowledge covers other supportive information for establishment, monitoring and enforcement of trust relationships. Subjective factors include trust intentions (represented in policies) and reputations (represented in trust values). Two questions need to be answered for these trust factors:

- Whether a federated trust management system includes these trust factors?
- What are the representative qualities of these trust factors?

The answer for the first question is yes or no. The answer for the second can be from a fuzzy set {low, medium, high}, or can be a numeric value from an interval such as [0,1]. Extending the work of Klos and Poutré [17], reputation accuracy can be calculated from the difference between the calculated reputation, which is formed using a Beta probability from the action history of the target entity, and the actual reputation from other entities' votes.

Because federated trust management deals with different trust factors across security domains, such a system needs to provide flexibility to allow subjective adjustments to objective trust factors. For example, within one security domain, the system architect can be expected to have access to enough information to assure that a trust representation accurately reflects the accuracy of the underlying authentication technology; but for inter-domain federation, special managerial exceptions should be allowed. For example, if a user from a foreign domain wants to use the fingerprint template obtained at the foreign domain, but the local administrator does not fully trust that template because of lack of knowledge concerning the foreign domain's authentication devices and methods, the local administrator may lower the trust level of that fingerprint template to a password equivalent one in order to reduce risk in the local system. This is an operational or managerial adjustment, and it is subjective. So this question needs to be answered to evaluate a federated trust management system's adaptivity for trust representation:

- Can users make subjective adjustments to objective trust factors?

Table 2. Comparison of six major systems for trust representation

Trust representation			Objective factors			Subjective factors	
			Identity	Privilege	Trust knowledge	Policy	Reputation
Quality of functionality	Completeness	IBM TE	Yes	Yes	No	Yes	No
		PolicyMaker	Yes	Yes	Yes	Yes	No
		KeyNote	Yes	Yes	No	Yes	No
		REFEREE	Yes	Yes	Yes	Yes	No
		Liberty	Yes	Yes	Yes	Yes	No
		TrustBuilder	Yes	Yes	Yes	Yes	Yes
	Accuracy	IBM TE	High	High	N/A	Yes	N/A
		PolicyMaker	High	High	Medium	High	N/A
		KeyNote	High	High	Medium	High	N/A
		REFEREE	High	High	Medium	High	N/A
		Liberty	High	High	High	High	N/A
		TrustBuilder	High	High	High	High	Medium
	Adjustment	IBM TE	Yes	Yes	N/A	Yes	N/A
		PolicyMaker	No	No	No	N/A	N/A
		KeyNote	No	No	No	N/A	N/A
		REFEREE	No	No	No	Yes	N/A
		Liberty	No	No	No	Yes	N/A
		TrustBuilder	No	Yes	Yes	Yes	No
Cost of functionality	Cost	IBM TE	Low	Low	N/A	Low	N/A
		PolicyMaker	Low	Low	Low	Low	N/A
		KeyNote	Low	Low	Low	Low	N/A
		REFEREE	Low	Low	Low	Low	N/A
		Liberty	Medium	Medium	Medium	Medium	N/A
		TrustBuilder	Medium	Medium	Medium	Medium	Medium
	Duration	IBM TE	Low	Low	N/A	Low	N/A
		PolicyMaker	Low	Medium	Low	Medium	N/A
		KeyNote	Low	Medium	Low	Medium	N/A
		REFEREE	Low	Medium	Low	High	N/A
		Liberty	Medium	Medium	Medium	Medium	N/A
		TrustBuilder	Medium	Medium	Medium	Medium	Medium
Usability	Understandability	IBM TE	Low	High	N/A	Low	N/A
		PolicyMaker	High	Low	Low	Low	N/A
		KeyNote	High	Low	Low	Low	N/A
		REFEREE	High	High	High	High	N/A
		Liberty	Medium	Medium	Medium	Medium	N/A
		TrustBuilder	High	High	High	High	Medium
	User acceptance	IBM TE	High	High	N/A	Medium	N/A
		PolicyMaker	High	Medium	Low	Medium	N/A
		KeyNote	High	High	Low	Medium	N/A
		REFEREE	High	High	High	High	N/A
		Liberty	High	High	High	High	N/A
		TrustBuilder	High	High	High	High	Medium

The evaluation of performance for trust representation is quite straightforward. It consists of measurements of cost and duration. For example, the cost of identity information includes the cost for different authentication technologies deployed within a security domain and the cost needed to train users; the duration measurement needs to

cover the time used for different authentication procedures with different technologies and the time used to transform authentication templates into formal representations. The evaluation of usability mainly focuses on user interfaces, which includes the understandability of the user interfaces and the acceptance of users. Table 2 shows a comparison of six major systems' representation capability. The "yes/no" answers the question of whether an objective or subjective factor is supported; the "high/medium/low" measures the level of satisfaction for a certain measurement. Using the "quality" and "usability" criteria, table 2 identifies the strengths of TrustBuilder over other systems, which are the support for reputations, the adjustment capability for trust factors, and the usability of the representations for various trust factors such as trust knowledge and policies. Since permitted privileges in KeyNote and PolicyMaker sometimes are small structured programs, it is difficult to understand the actual meaning of these privileges and the entire policy rules. And these small structured programs can also be embedded into credential assertions (a type of trust knowledge), which share the same syntax with privileges. It is even more difficult to understand various types of trust knowledge in policy rules written by administrators and users.

5.3 Trust Establishment

The main functionalities of trust establishment are establishment and maintenance of trust relationships. These two aspects need to be considered simultaneously to evaluate a federated trust management system's capabilities associated with trust establishment. Support of negotiation is the most fundamental requirement, so the questions below need to be answered to measure the generality of a system.

- Does the system support negotiation protocols specific for that type of application or that domain?
- Does the system support general or user-defined negotiation protocols?

To evaluate the adaptivity of a system's trust establishment capability, its support for different trust models needs to be examined. The questions below are related.

- Does the system support direct trust establishment?
- Does the system support indirect trust establishment?

Privacy protection is required when a user's private attributes are disclosed. Expert evaluation of this capability will be required. Observation and maintenance of a trust relationship's evolution and update is also one of the main tasks for trust establishment, so it should be included in the evaluation.

The performance of trust establishment for a federated trust management system can be measured by the average time used for the establishment process and its user involvement. The degree of user involvement is also a good measurement of usability. Table 3 shows a comparison of six major systems' establishment capability. The "yes/no" answers the question of whether certain functionality is supported. I simulate the establishment process between a hospital domain and a pharmacy domain using IBM Trust Establishment Framework, Liberty Alliance Project, and TrustBuilder with only one challenge and one verification. The numerical data is obtained from this one-challenge-and-one-verification negotiation protocol. The results show that IBM Trust Establishment Framework can only support direct negotiations, and it is more efficient

Table 3. Comparison of six major systems for trust establishment (A trust establishment simulation with one challenge and one verification)

Trust establishment		Application specific negotiation	General negotiation	Direct establishment	Indirect establishment	Privacy protection
Quality of functionality	IBM TE	Yes	No	Yes	No	No
	PolicyMaker	N/A	N/A	N/A	N/A	N/A
	KeyNote	N/A	N/A	N/A	N/A	N/A
	REFEREE	N/A	N/A	N/A	N/A	N/A
	Liberty	Yes	Yes	Yes	Yes	No
	TrustBuilder	Yes	Yes	Yes	Yes	Yes
Cost of functionality (Duration)	IBM TE	40ms	N/A	40ms	N/A	N/A
	PolicyMaker	N/A	N/A	N/A	N/A	N/A
	KeyNote	N/A	N/A	N/A	N/A	N/A
	REFEREE	N/A	N/A	N/A	N/A	N/A
	Liberty	200ms	200ms	200ms	500ms	N/A
	TrustBuilder	100ms	100ms	100ms	1s	N/A
User involvement	IBM TE	No	N/A	No	N/A	N/A
	PolicyMaker	N/A	N/A	N/A	N/A	N/A
	KeyNote	N/A	N/A	N/A	N/A	N/A
	REFEREE	N/A	N/A	N/A	N/A	N/A
	Liberty	No	No	No	No	N/A
	TrustBuilder	No	No	No	No	No

than Liberty Alliance Project and TrustBuilder for this simple one-challenge-and-one-verification negotiation protocol, because IBM Trust Establishment Framework is implemented in C and does not use web services as its communication interface. On the other hand, compared with IBM Trust Establishment Framework, Liberty Alliance Project and TrustBuilder can support indirect trust establishment and provide more flexibility, which supports adjustments to dynamic trust relationships.

5.4 Trust Enforcement

The most important functionality of trust enforcement is to guarantee compliance between trust behaviors and their corresponding governing policies stating users' trust intentions. This part includes enforcement of trust intentions, definition of trust intentions, revocation of privileges, checking of the validity of identity information and trust knowledge, and privacy protection.

To measure this compliance guarantee, it is important to count the number of compliance failures (or false revocations) within a system running for a certain period of time, and to examine a system's capability to resolve trust intention conflicts with or without human involvement. For example, if two trust intentions describe the constraints for the same trust behavior, the system should either be able to determine which one has superior authority or allow the domain administrator to decide which one is to be used. Thus two questions below need to be answered to measure this functionality.

- Is the system able to solve trust intention conflicts?
- Does the resolution process need human intervention?

Table 4-A. Comparison of six major systems for trust enforcement (first half)

Trust enforcement			Definition of trust intentions	Enforcement of trust intentions	Revocation of privileges	Validation of trust information	Privacy protection
Quality of functionality		IBM TE	Yes	Yes	Yes	No	No
		PolicyMaker	Yes	Yes	Yes	Yes	No
		KeyNote	Yes	Yes	Yes	Yes	No
		REFEREE	Yes	Yes	Yes	Yes	No
		Liberty	Yes	Yes	Yes	Yes	No
		TrustBuilder	Yes	Yes	Yes	No	No
Usability	User's awareness	IBM TE	Low	Low	Low	N/A	N/A
		PolicyMaker	Low	Low	Low	Low	N/A
		KeyNote	Low	Low	Low	Low	N/A
		REFEREE	Low	Low	Low	Low	N/A
		Liberty	Low	Low	Low	Low	N/A
		TrustBuilder	Low	Low	Low	N/A	N/A
	User interface's acceptance	IBM TE	N/A	N/A	N/A	N/A	N/A
		PolicyMaker	N/A	N/A	N/A	N/A	N/A
		KeyNote	N/A	N/A	N/A	N/A	N/A
		REFEREE	N/A	N/A	N/A	N/A	N/A
		Liberty	N/A	N/A	N/A	N/A	N/A
		TrustBuilder	N/A	N/A	N/A	N/A	N/A
	User interface's operability	IBM TE	N/A	N/A	N/A	N/A	N/A
		PolicyMaker	N/A	N/A	N/A	N/A	N/A
		KeyNote	N/A	N/A	N/A	N/A	N/A
		REFEREE	N/A	N/A	N/A	N/A	N/A
		Liberty	N/A	N/A	N/A	N/A	N/A
		TrustBuilder	N/A	N/A	N/A	N/A	N/A

Due to the dynamic trust relationships and ever-changing contexts for federated trust management, a federated trust management system needs to be able to detect changed trust relationships and updated context information. One performance measurement is the extra time (overhead) used by the trust enforcement processes. Usability can be evaluated by answering below questions.

- Are users aware of the trust enforcement process?
- Are users comfortable with using the interface?
- Do users know how to use the interface?

Table 4 shows a comparison of six major systems' enforcement capability. The two halves (table 4-A and table 4-B) compare different aspects of a systems' enforcement capability. The "yes/no" answers the question of whether certain functionality is supported; the "high/medium/low" measures the level of satisfaction for a certain usability measurement. Although this is only a rough comparison, the difference between Liberty Alliance Project, TrustBuilder, and other systems can be easily identified. Liberty Alliance Project and TrustBuilder provide better quality and usability. Through this comparison, the fact that Liberty Alliance Project provides better usability than TrustBuilder for most trust enforcement capability can also be identified. Validation of trust information and dynamic trust relationship detection are supported by Liberty Alliance Project but not by TrustBuilder.

Table 4-B. Comparison of six major systems for trust enforcement (second half)

Trust enforcement			Compliance check	Trust intention conflict resolution	Human intervention	Dynamic trust relationship detection	Context information update
Quality of functionality		IBM TE	0 failure	No	No	No	No
		PolicyMaker	0 failure	No	No	No	No
		KeyNote	0 failure	No	No	No	No
		REFEREE	0 failure	No	No	No	No
		Liberty	0 failure	Yes	No	Yes	Yes
		TrustBuilder	0 failure	Yes	No	No	Yes
Usability	User's awareness	IBM TE	Low	N/A	N/A	N/A	N/A
		PolicyMaker	Medium	N/A	N/A	N/A	N/A
		KeyNote	Medium	N/A	N/A	N/A	N/A
		REFEREE	High	N/A	N/A	N/A	N/A
		Liberty	High	Medium	N/A	Medium	Medium
		TrustBuilder	Low	Low	N/A	N/A	Low
	User interface's acceptance	IBM TE	N/A	N/A	N/A	N/A	N/A
		PolicyMaker	N/A	N/A	N/A	N/A	N/A
		KeyNote	N/A	N/A	N/A	N/A	N/A
		REFEREE	N/A	N/A	N/A	N/A	N/A
		Liberty	N/A	N/A	N/A	N/A	N/A
		TrustBuilder	N/A	N/A	N/A	N/A	N/A
	User interface's operability	IBM TE	N/A	N/A	N/A	N/A	N/A
		PolicyMaker	N/A	N/A	N/A	N/A	N/A
		KeyNote	N/A	N/A	N/A	N/A	N/A
		REFEREE	N/A	N/A	N/A	N/A	N/A
		Liberty	N/A	N/A	N/A	N/A	N/A
		TrustBuilder	N/A	N/A	N/A	N/A	N/A

6 Conclusion

Again, the purpose of this paper is to initiate discussion of the requirements and evaluation metrics for federated trust management, to offer a context in which to evaluate and compare current and future solutions, and to encourage the development of proper systems for federation in networked systems. A well-defined, general-purpose, federated trust management system cannot be implemented before researchers and developers understand the needs of federation in addition to the needs of traditional trust management. And a comprehensive benchmark is also needed for in-depth comparisons and evaluations.

References

1. Ruohomaa, S., Kutvonen, L.: Trust Management Survey. In: Herrmann, P., Issarny, V., Shiu, S.C.K. (eds.) iTrust 2005. LNCS, vol. 3477, pp. 77–92. Springer, Heidelberg (2005)
2. Wu, Z., Weaver, A.C.: Requirements of federated trust management for service-oriented architectures. International Journal of Information Security 6(5), 287–296 (2007)

3. Grandison, T., Sloman, M.: A survey of trust in internet applications. IEEE Communications Surveys and Tutorials (Fourth Quarter, 2000), http://www.comsoc.org/pubs/surveys/
4. IBM: IBM Trust Establishment Policy Language, http://www.hrl.il.ibm.com/TrustEstablishment/PolicyLanguage.asp
5. IBM: Access Control Meets Public Key Infrastructure, or: Assigning Roles to Strangers. In: Proc. of IEEE Symposium on Security and Privacy (2000), http://www.hrl.il.ibm.com/TrustEstablishment/paper.asp
6. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: Proc. 1996 IEEE Symposium on Security and Privacy, pp. 164–173 (1996)
7. Blaze, M., Feigenbaum, J., Keromytis, A.D.: KeyNote: Trust Management for Public-Key Infrastructures. In: Christianson, B., Crispo, B., Harbison, W.S., Roe, M. (eds.) Security Protocols 1998. LNCS, vol. 1550, pp. 59–63. Springer, Heidelberg (1999)
8. Chu, Y.-H., Feigenbaum, J., LaMacchia, B., Resnick, P., Strauss, M.: REFEREE: Trust Management for Web Applications. World Wide Web Journal 2, 127–139 (1997)
9. Liberty Alliance: Liberty Alliance Complete Specifications ZIP Package (2008), http://www.projectliberty.org/resource_center/specifications/liberty_alliance_complete_specifications_zip_package_22_june_2008
10. OASIS Security Services TC: Security Assertion Markup Language (SAML) V2.0 Technical Overview (2008), http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf
11. Ryutov, T., et al.: Adaptive Trust Negotiation and Access Control. In: Proc. of 11th ACM Symposium on Access Control Models and Technologies, pp. 139–146 (2005)
12. Ryutov, T., Neuman, C.: The Specification and Enforcement of Advanced security Policies. In: Proc. of the 2002 Conference on Policies for Distributed Systems and Networks (2002)
13. Winslett, M., Yu, T., Seamons, K.E., et al.: The TrustBuilder Architecture for Trust Negotiation. IEEE Internet Computing 6(6), 30–37 (2002)
14. Klos, T., Poutre, H.L.: Using Reputation-Based Trust for Assessing Agent Reliability. In: Proc. of the AAMAS-2004 Workshop on Trust in Agent Societies, pp. 75–82 (2004)
15. Fullam, K., Barber, K.S.: A Temporal Policy for Trusting Information. In: Proc. of the AAMAS-2004 Workshop on Trust in Agent Societies, pp. 47–57 (2004)
16. Ghanea-Hercock, R.: The Cost of Trust. In: Proc. of the AAMAS-2004 Workshop on Trust in Agent Societies, pp. 58–64 (2004)
17. Klos, T.B., Poutré, H.L.: Decentralized reputation-based trust for assessing agent reliability under aggregate feedback. In: Proc. of the 7th Workshop on Trust in Agent Societies, pp. 75–82 (2004)