Autonomous Trust for Web Services

M. Coetzee[1] and J. H. P. Eloff[2]

Information and Computer Security Architectures (ICSA) Research Group
Department of Computer Science, University of Pretoria, Pretoria, South Africa
email: [1]marijke@acm.org, [2]eloff@cs.up.ac.za

Abstract

Current approaches for Web Services do not support differentiated forms of trust, which is required when interacting with either partners or strangers. By inspecting information and evidence, a Web Service can over time, gain a sense of the trustworthiness of other services participating in its environment. In this paper, trust is discussed as it may relate to a Web Service environment. Trust management models, and an existing trust specification for Web Services are discussed. A novel approach to Web Service trust formation is presented. Trust is formed by using information that is published through Web Service standards, defined over and above a Web Service interface. The proposed approach is concerned with a notion of trust that includes more than cryptographic controls. It has mechanisms that allow a Web Service to manage trust autonomously, enabling different types of trust for different situations.

Key words

Virtual application, Web Services, WS-Policy, trust, trust ontology, trust formation

1. Introduction

The Internet has revolutionized the capacity to share information and services across organizations. As the business functionality of organisations becomes more digitised, it exhibits software characteristics such as virtuality and real-time operation. These characteristics are manifested through new and innovative developments, such as virtual applications. The notion of the virtual application is supported through Web Services technology [GOTT02] that enables organizations to exploit software as a service, through service virtualization. A Web Service is a type of service, identified by a URI that can be invoked through an Internet connection with SOAP [BOXD00] messages. Examples of virtual applications are retail portals, integrated travel planning and insurance brokering.

Virtual applications hold the promise of bringing together large numbers of collaborators across dispersed environments on a scale not known to the real world. The machine-to-machine interactions that support virtual applications indirectly constitute a quasi form of social collaboration. For such applications, access to resources under the control of one machine, must be granted to large numbers of users who are only known to another machine, in a different domain. The distributed architecture of virtual applications presents some difficulty as user credentials need to be verified across domains. Traditional authentication and authorisation mechanisms cannot rise to meet this challenge. Credential-based access control, where the capability of the user is conveyed in, for example, SOAP headers, provides a viable solution [HALL02] [COET04]. An important requirement is that participants trust each other.

Trust is a positive concept that expresses the belief that the other party will behave as expected, where the belief is based on the lack of contrary evidence [GAMB88]. From the current body of research [MARS94], [GRAN03], [DIMI03], properties of trust have been identified. Properties that are important to mention are: trust is dependent on a specific context or situation; it is a measurable belief that reflects its strength; it evolves with time through new experiences and observations and is subjective.

Considering the nature of Web Services, it is also important to understand how trust is formed between organisations. Trust relationships are influenced by market forces, social interactions, legal and assurance systems and insurance. A model of inter-organisational trust illustrates that trust is established in three stages [RATN01]. Firstly, *competence trust* is established through the trust and security-based mechanisms that are embedded in e-commerce technologies, to provide speed and real-time accurate information. Secondly, consistent positive behaviours from trading partners lead to credibility and reliability, which creates *predictability trust*. Lastly, *goodwill trust* focuses on organisational reputation and brand names, accomplished by enforcing best business practices.

In society, people learn to trust each other by collecting information through their own experiences, observations, and recommendations from others. For distributed systems, it has been argued that trust should be based on information as far as possible [JØSA96]. For Internet applications, trust management has been defined as the act of collecting, codifying, analysing and presenting evidence that relates to competence, honesty, security or dependability, to be able to form trust relationships with others [GRAN03]. These statements lead to the focus of this paper: to create a trust formation approach for Web Services by sourcing, analysing, and categorising the information available in the XML-based environment. The approach can give a Web Service the ability to determine the trustworthiness of others, at execution time, instead of determining it manually over an extended period. Existing policy languages such as WS-Policy [BOXD03] that defines service metadata can be used for this purpose.

This paper is structured as follows: section 2 gives a background to the problem. Section 3 presents a trust information ontology that describes useful concepts for trust formation between Web Services. Section 4 describes how information can be sourced from the underlying XML-based environment with machine-to-machine interactions. Section 5 attempts to illustrate how trust can be formed by a so-called trust engine. Section 6 concludes the paper.

2. Background

Trust between Web service requestors and providers will form the basis of all exchanges that may take place between them. A Web Service may be part of a community consisting of participants that have forged strong relationships of trust with each other over time. A level of goodwill can be created, that enables Web Services to share more information, and grant further and advanced access to others. In contrast, ad-hoc participants may introduce themselves to a Web Service for the purpose of a once-off transaction. It would be impractical to expect of a Web Service to have the same level of trust and give the same level of access to well known and to unacquainted parties.

In order to address trust for Web Services, the Web Services Trust Language or WS-Trust [DELL03] has been published. It allows interoperability between a Web Service requestor and provider that do not know each other, by enabling them to determine whether they can trust each others' asserted credentials. WS-Trust poses limitations to the establishment trust relationships as it does not enable a Web Service to treat partners and strangers differently. Trust established between Web Services is of binary format, it either exists or doesn't.

Current trust management solutions [BLAZ99], [RIVE96] are focused on solving the problem of distributed access control, and do not reflect the maturity of trust relationships. They present a number of shortcomings to Web Services participating in virtual applications. For instance, trusted third parties that are very often required may be absent or inaccessible for some partners of virtual applications; trust is strongly based on cryptographic controls, and is generally defined over the identity of partners; and trust management models are very complex and time-consuming to implement.

To address above-mentioned shortcomings, an autonomous approach to trust formation is required, where each Web Service makes independent trust decisions based on its own observations. The result from this paper is a trust formation approach that will be able to assign a degree of trust to Web Service requestors, for a given context, based on information that is sourced and evidence that is presented. Evidence may for instance consist of certificates for proof of identity, certificates describing competence, and risk assessments. In order to use information for the purpose of trust formation, a basic ontology of trust information needs to be defined. This is the focus of the next paragraph.

3. Trust ontology

Uncertainties about strangers and partners lead to lack of trust. To counter this, information can be used to form trust relationships. The strength of the trust formed is determined by the type of information that can be sourced. Information can be sourced, amongst other means, from references, experiences, recommendations, and XML-based policy documents.

References: A primary concern for a Web Service is a lack of knowledge about a new partner. A new partner first needs to prove its competence in a specific domain. If a new partner is endorsed by trusted authorities through references, it can be assigned a basic level of trust. References are statements in the form of certificates from independent third parties.

Recommendations: As it is not possible for a partner to evaluate all aspects of a given situation when making a trust decision, a Web Service can also rely on recommendations from others to form a trust relationship. A recommendation is an opinion obtained from another party, for a specific situation or context such as the delivery of goods or the quality of information provided. It is important to consider how much the third party can be trusted, and what trust can be extended to the party under consideration.

Experience: Trust is also created through the progressive gain of experience with others. Experience refers to the cumulative view of the result of interactions with a party in a context.

XML-based policy documents: For Web Service interoperation, a considerable amount of information has to be made available in a machine-readable format in different XML-based policies. Both the service provider and requestor can describe what they offer and demand from the other party. For instance, information that describes supported security mechanisms of a partner may be useful when trust relationships are formed.

The structure of the required information must be made explicit so that it can be understood by all participants. An ontology can assist the process of inter-operability and specification for the trust formation process [JASP99] The ontology presented here in figure 1 is a simple taxonomy that defines relevant concepts.

Three types of information exist. Firstly, s*tructural* information is based on the properties of the system or institution within which the trust relation exists [CHER96]. A Web Service may trust others because of the *belief* that it has of the other party. Finally, a Web Service may have *domain* information that reflects its own expertise. Each of the three categories of information that can be used to form trust will now be described in more detail:

Structural information: Structural assurances is information that can be used to give a Web Service the confidence that measures exist that can provide safeguards and reduce the risk when something goes wrong. Legal contracts, assurances and implemented security mechanisms may play a role in trust formation.

- Legal contracts - parties who have a contract with each other have indirect trust in each other, because the judicial system exists and will enforce the contract.
- Assurances - licenses and insurance policies provide additional safeguards to protect against risk.
- Security mechanisms - properties that can most affect a trust relationship are the identity of the services, and the security properties that services offer [KHAN01]. Security based mechanisms such as authentication, integrity, and confidentiality ensures timely, accurate and complete transmission and receipt of transactions. In addition, policies, procedures, and standards, encapsulated by best practise ensure smooth functioning of interactions. By establishing the identity of a partner with a digital certificate, a basic level of trust can exist as a Web Service can have the confidence that it is dealing with the right partner. A service will next need to establish to what extent the new or existing partner makes use of security mechanisms such as digital signatures, encryption mechanisms, authorization mechanisms, and best business practices that enforce quality and standards will impact trust formation.

Belief information: Categories of beliefs is information that will determine the extent to which a Web Service can trust others. Trust develops from hard concepts such as security mechanisms to soft concepts such as beliefs [RATN01]. A comprehensive study of over sixty papers covering a wide range of disciplines, has shown that beliefs can be categorised by honesty, competence, predictability and benevolence [CHER96].

- Honesty is the belief that agreements with a partner are made in good faith. It can be established through recommendations from trusted parties and experiences.
- Competence is the belief that a partner has the necessary skills to do a task. Information that can give a partner confidence in another is certificates from third parties such as ISO 9000 certificates, licenses, credit ratings, audit information and endorsements.
- Predictability is the belief that the actions of a partner are consistent so that a forecast can be made about what such a partner will do in a given situation. This can be achieved by inspecting SOAP messages that are sent and received and by recording for instance: the number of messages in error, the value of transactions, the number of transactions, and the validity of message details.
- Benevolence is the belief that a partner cares about the welfare of the other. It may be established over time as a partner realises the benefits gained from increased cooperation with another party.

It would be easier to establish the competence and predictability of a partner than the honesty and benevolence. These categories are not equally important for all situations. For instance, when a payment is made, the benevolence of a partner may not be important to establish, but rather the honesty and competence.

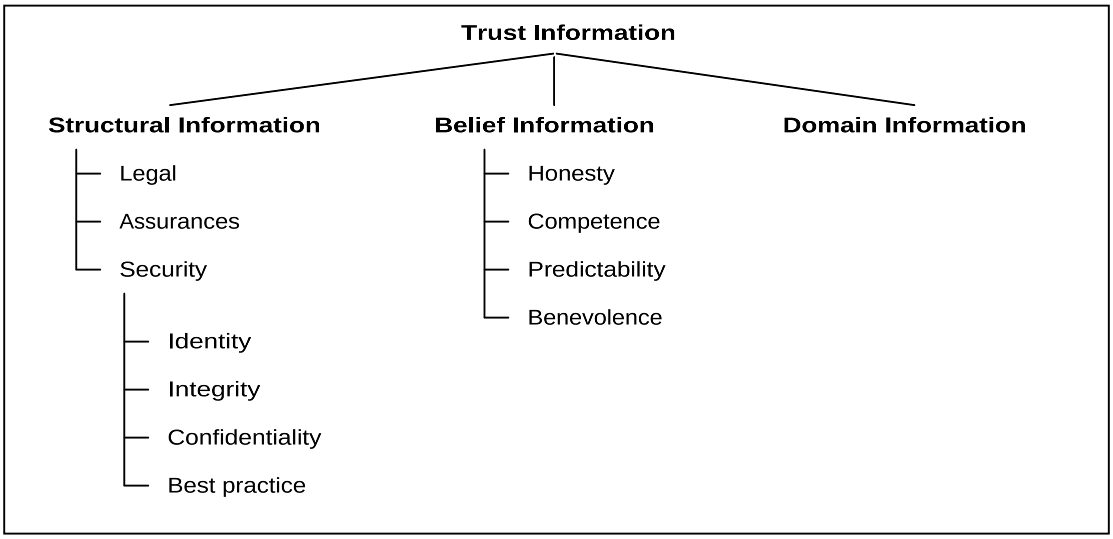

Figure 1: Ontology of trust information

Domain information: Domain information is expertise that exists within the environment of a Web Service that can be used to form trust relationships. It affects the trust extended to partners in a given situation. This information does not have to be sourced externally.

The information that is required for trust formation can be sourced either manually through administrator intervention or automatically by a machine. For instance, information sourced by a researcher on the financial position of a partner may be added to the pool of information by an administrator, whereas information stored in a digital certificate may be processed automatically by a machine. It is also possible that a combination can be used. The next paragraph shows that it is possible for the machines supporting Web Services to source a substantial portion of information in order to form trust relationships.

4. Sourcing information for trust formation

Machine-to-machine automation plays an important role to enable unacquainted Web Services to inter-operate with each other. For this purpose, information is made available in a machine-readable format in the form of meta-data over and above the service interface. A WSDL file lists the operations, parameters and data types used by a service and its schema allows the exposure of functional incompatibilities that may exist between services. Thereafter, UDDI (Universal Description, Discovery, and Integration) provides a registry of businesses and web services. It describes businesses by their physical attributes such as name and address, industry classification, and technical details of the services that they provide. As UDDI registries are not moderated, entries are often invalid [MODI02]. Both WSDL and UDDI do therefore not provide information to enable partners to establish a measure of trust with each other.

As it is necessary to inform requestors of the non-functional aspects of a Web Service, WS-Policy [BOXD03] defines another layer of information. A WS-Policy document may be associated with a specific operation or message, or a set of services. By defining non-functional requirements in a separate layer from the service interface, implementation code is unaffected. Even though Web Service requestors may not necessarily be service providers themselves, they can also use WS-Policy to communicate requirements to Web Service providers [REMY04]. Quality-of-service issues such as security, privacy, performance and availability that are required and supported, by the Web Service provider or requestor, for SOAP messages to be exchanged, are defined. The next paragraphs show how WS-Policy can be used to publish information such as supported security mechanisms, independent references, and recommendations. It is impossible to determine whether a service is not truthful about its properties. Information such as dishonest behaviour can be recorded as a bad experience, and used in further trust formation.

Security mechanisms: WS-Policy currently communicates to others the security mechanisms it supports through a set of security policy assertions defined within the WS-Security specification [ATKI02] To enable secure communication with another party, an endpoint needs to know whether the other party supports WS-Security, and which security tokens can be used. For instance, an endpoint may support any combination of UsernameToken, Kerberos ticket, or certificate, but may prefer a certificate. An endpoint must also determine if the other party requires signed messages, and what token type must be used for the digital signatures. If encryption is required, the other party must know when to encrypt the messages, which algorithm to use, and how to exchange a shared key with the service. These security requirements are addressed by the `<wsse:SecurityToken>, <wsse:Integrity>` , and `<wsse:Confidentiality>` elements. This information allows an endpoint to trust another based on supported security mechanisms.

To enable the further establishment of trust, WS-Policy is now extended here with new structures to enable the establishment of trust based on references and recommendations.
References: The existence of references can be revealed to prospective partners in a WS-Policy document. As a business publishes its own references, a partner needs to confirm the validity of the information with the issuing party. It would be important to know the type of reference, location where it can be found, date created, expiry date and issuing authority. A partner must verify each one before it can be used. The WS-Policy document has the following structure:

```
 <wsref:Reference>
    <wsref:Type>…………………</wsref:Type>
    <wsref:Location>…… some uri…</wsref:Location>
    <wsref:Name>…………………</wsref:Name>
    <wsref:Issuer>…some uri…<wsref:Issuer>
    <wsref:IssueDate>…………</wsref:IssueDate>
</wsref:Reference >
```

Recommendations: A Web Service can publish a list of partners in the WS-Policy document from whom it would accept recommendations. Prospective partners can use this list to get recommendations that will be trusted by the Web Service. The name and location of the trusted partner can be published, as shown below:

```
<wsbp:Partners>
     <wsbp:Name>………………</wsbp:Name>
     <wsbp:Location>…………………</wsbp:Location>
<wsbp:Partners>
```

Recommendations signed by the issuer are returned to the service in the format shown below. It includes the referee name and location, context of the recommendation, value or degree and the date of creation, and date of expiration. Context and value elements may be defined uniquely for a provider in a schema. As the recommendation is in a machine-readable format, with context and values defined by a schema, it can be understood dynamically by the service.

```
  <wsrecm:Recommendation>
      <wsrecm:RefereeName>..................<wsrecm:RefereeeName>
      <wsrecm:RefereeLocation>..................<wsrecm:RefereeeLocation>
      <wsrecm:Context>..................<wsrecm:Context>
      <wsrecm:Value>..................<wsrecm:Value>
      <wsrecm:DateCreated>..................<wsrecm:DateCreated>
      <wsrecm:DateExpire>..................<wsrecm:DateExpire>
 <wsrecm:Recommendation>
```

Structural information can thus automatically be established by inspecting the WS-Policy documents of partners to establish whether they comply with the security requirements of the Web Service. References of partners can be used to establish other assurances. Belief information can be established by processing references of partners, recommendations that may accompany requests, and recording experiences by inspecting SOAP messages. Domain information will be used to reason about presented information and evidence. It will be made part of the trust evaluation process through administrator interventions.

Mechanisms must exist at a Web Service and its partner to support the publication of policies, the interchange of references and recommendations, and recording of experiences. Protocols ensure that messages are sent correctly, so that they are understood by communicating parties.

5. Trust formation

In order to enable a Web Service to form trust relationships with others, the addition of a trust engine is proposed. The trust engine should be accommodated in the Web Service architecture, before service interaction. The trust engine is knowledgeable about the requirements of the Web Service, the standards that are used and the threshold of trust that is required. It inspects WS-policy documents of partners to source information. When requests are sent to a Web Service with references and recommendations, its trust engine will intercept the request to as to verify the validity of information with independent third parties. It records experiences by inspecting SOAP messages. It mediates all trust related interactions with partners and strangers. It intelligently processes all information that it sources, or that is presented to it.

The trust engine assigns values to information that is sourced. Values may range between 0 and 1 [MARS94], [GAMB88]. Trust is formed based on the strength of structural information, the beliefs that a Web Service holds about a partner, and its domain expertise, which may be reflected as weights that it assigns to trust values to reflect their importance to the Web Service. Any change in the information sourced can alter the trust value.

A calculation producing a trust value must be made that will reflect the cumulative trust that a Web Service holds towards a partner in a given context. The calculation may be based on various techniques such as an algorithmic or fuzzy approach. It is also important to be able to create a trust value that shows the difference between distrust and ignorance. The trust engine thus gains a sense of the trustworthiness of a partner.

6. Conclusion

This paper discusses the formation of trust between Web Services who participate in virtual applications. Web Services would rather interact with honest and reliable partners, as it will minimize its exposure to risky transactions. An autonomous trust formation approach is proposed that includes a degree of trust. A computed trust value will allow a Web Service to reason about relevant information and evidence before making a decision. Decisions about whom to trust are based on the properties of partners, their environment, and trusting beliefs that the Web Service holds over the partner.

It is shown that the machines supporting Web Services can source some of the information that is required for trust formation. Further research will focus on how information can be exchanged with partners, and the calculation of the trust value. Fuzzy logic seems useful as it uses natural language labels. Statements such as “the partner has a *high* level of competence” represent intervals rather that exact values, which may be more natural to implement.

7. Acknowledgement

The financial assistance of the Department of Labour (DoL) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the DoL.

8. References

[ANDE03] Anderson A. et. al., XACML 1.0 Specification, 2003, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

[ATKI02] Atkinson B. et al., Web Services Security (WS-Security), Version 1.0 April 5, 2002, http://www.verisign.com/wss/wss.pdf

[BELL03] Bellwood T., Clément L., Von Riegen C., et. al. UDDI Version 3.0.1, 14 Oct 2003, http://uddi.org/pubs/uddi-v3.0.1-20031014.htm

[BLAZ99] Blaze M., Feigenbaum J., Ioannidis J., and Keromytis A., “The KeyNote Trust-management System, version 2,” IETF, RFC 2704, September 1999.

[BOXD00] Box D., Ehnebuske D., Kakivaya G., Layman A., Mendelsohn N., Nielsen H. F, Thatte S. and Winer D., “Simple Object Access Protocol (SOAP) 1.1”, http://www.w3.org/TR/SOAP/, May 2000

[BOXD03] Box D., Web Services Policy Framework (WS-Policy), 2003 http://www.ibm.com/developerworks/library/ws-policy/index.html

[CHER96] Chervany N. L. and MgKnight D. H. The meanings of trust. Technical Report 94-04, Carlson School of Management, University of Minnesota, 1996.

[COET04] Coetzee M., Eloff J. H. P., Towards Web Services access control, Computers and Security, Vol 23, No 7, Elsevier publishers, UK

[DELL03] Della-Libera G. et al., Web Services Trust Language (WS-Trust), http://www.ibm.com/developerworks/library/ws-trust/index.html

[DIMI03] Dimitrakos T. A Service-Oriented Trust Management Framework. In *Trust, Reputation, and Security: Theories and Practice*, Vol. 2631, p. 53-72. Rino Falcone, Suzanne Barber, Larry Korba and Munindar Singh, Lecture Notes in Computer Science, Springer-Verlag, 2003.

[GAMB88] Gambetta D.. *Can We Trust Trust?*, chapter 13, pages 213-237. Basil Blackwell, 1988. Reprinted in electronic edition from Department of Sociology, University of Oxford.

[GOTT02] K. Gottschalk, S. Graham, H.Kreger and J.Snell, Introduction to Web services architecture, IBM Systems Journal, Volume 41, Number 2, 2002

[GRAN03] Grandison T. W. A., Trust Management for Internet Applications, PhD Thesis, Imperial College of Science, Technology and Medicine, University of London, Department of Computing, 2003

[HALL02] Hallam-Baker P., Hodges J., Maler E., McLaren C., Irving R., SAML 1.0 Specification, 2003, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[JASP99] Jasper R. and Uschold M. A Framework for Understanding and Classifying Ontology Applications, in Proceedings of the IJCAI99 Workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, (August 1999).

[JØSA96] Jøsang A. The right type of trust for distributed systems. In *New Security Paradigms Workshop*, 1996.

[MARS94] Marsh, S., Formalising Trust as a Computational Concept, PhD Thesis, University of Stirling, UK, 1994

[MODI02] Modi T., WSIL: Do we need another Web Services Specification?, www.webservicearchitect.com

[RATN01] Ratnasingam P. P., Interorganizational trust in Business to business e-commerce, PhD thesis, Erasmus University Rotterdam, 2001

[REMY04] Remy D., Rosenberg J., Securing Web Services with WS-Security, Sams publishing, Indiana, USA, 2004

[RIVE96] Rivest R. and Lampson B., “SDSI - A Simple Distributed Security Infrastructure,” October 1996.

This document was created with Win2PDF available at http://www.daneprairie.com.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.