October 20, 2003

RT0553

Security 20 pages

Research Report

Assurance of Web Services

Sachiko Yoshihama, Paula K. Austel, Hiroshi Maruyama

IBM Research, Tokyo Research Laboratory
IBM Japan, Ltd.
1623-14 Shimotsuruma, Yamato
Kanagawa 242-8502, Japan

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Assurance of Web Services

Sachiko Yoshihama[†], Paula K. Austel[‡], Hiroshi Maruyama[†]

[†]IBM Research, Tokyo Research Laboratory
1623-14 Shimotsuruma, Yamato,
Kanagawa 242-8502, Japan

[‡]IBM T. J. Watson Research Center
19 Skyline Drive, Hawthorne,
NY 10532, USA

Status of this Document

This document shows the authors' position on Web Services Assurance, a proposal of a new building block built on the WS-Security frameworks. The technology discussed in this document is still under development, and publication of this document does not imply endorsement by IBM.

Abstract:

WS-Assurance is a framework for communicating information that a Web Service can present as evidence of its trustworthiness so that a user (requester) can make intelligent decisions regarding their use of the service. It consists of three components: the general framework for communicating evidence, vocabulary for business assurance and mechanism for platform assurance. The framework is defined as a natural extension of the WS-Security family of specifications.

1 Introduction

The current and planned Web Services Specifications describe a composable set of functionality for providing distributed services in a heterogeneous computing environment. A service describes its functional interface in WSDL (Web Services Description Language)[4] and advertises it, for example, through a centralized directory such as a UDDI (Universal Description, Discovery and Integration) [5] registry. The messages sent between services are usually encoded using the highly interoperable XML based SOAP protocol [1][2][3] and transported over HTTP. These services are secured through the various Web Services Security Specifications defined in the Web Services Security Roadmap [8].

Before businesses can fully adopt web services to create high value transactions on the internet there needs to be a high level of assurance. Part of this assurance will be achieved through the composability of security, reliable messages, policy and transactions. [17]

At the simplest level the messages need to be protected against malicious parties who would read sensitive data or alter messages either for their benefit or to deny services to others. The WS-Security specification defines mechanisms for achieving message integrity and confidentiality. The security model builds to cover establishing trust between businesses (WS-Trust)[19], exchanging business, security and privacy policies (WS-Policy [9], WS-SecurityPolicy [19], WS-Privacy), establishing a security context that will last for a series of message exchanges (WS-SecureConversation [20]) and federating identities across businesses (WS-Federation [11]).

The businesses will now have some assurance about their trust relationship, policies and secure message exchange. The next step in assuring business transactions is reliable delivery of the messages. Networks are inherently unreliable. Packets can be lost and there can be no guarantee that a message is delivered once and only once. This is accomplished with WS-ReliableMessaging [21].

The last component that builds on this assurance is the transactions themselves. Transactions happen between several business partners and involve several exchanges of messages. WS-Transaction [23] and WS-Coordination [22] are the pieces that assure transactions are completed as expected.

In the WS-Security framework, a trust relationship is established based on the identity of each entity that is participating in a transaction. For example, the sender of a message is identified by a digital signature, which can be authenticated by the receiver using a certificate from a PKI. As a result, the decision would be binary – you may trust or not trust, depending on the authenticity of the signature and your trust relationship to the CA that issued the certificate. Finer granularity of trust is not considered, and it is assumed that this needs to be pre-negotiated using ordinary business practices.

In the real business world, an entity utilizes much finer grained information to decide whether another entity is trustworthy enough to make a deal. For example, a customer is generally interested in whether a service provider has ability to perform a certain service, with certain quality and quantity, by a certain deadline. A customer may also be interested in whether the service provider is honest, abides by the laws, respects the privacy policy, has a good history in past transactions and is conformant to the standard such as ISO9000, etc. On the other hand, a service provider is generally interested in the customer's financial credibility and his/her eligibility for receiving the service (e.g., does the customer have a license for buying controlled materials, etc.).

When the transaction takes place on the Internet, caution must be taken to secure the communication between entities. In addition to the security technologies developed to ensure integrity, authenticity, and confidentiality of the communications, it is also important to be guarantee that a computer platform, acting on behalf of an entity, behaves as expected. For example, if a client trusts an on-line shopping service and submits his credit card number – even if the service provider company is actually honest and trustworthy – the server platform might be infected with a Trojan horse that surreptitiously sends the credit card number to a malicious attacker. In another case, the server software might have a vulnerability that will be attacked in the near future, allowing the unauthorized release of customer information and credit card numbers. Therefore, it is important to make sure that the service is running on a trustworthy platform; i.e., it is running on the hardware that it claims to be, and that the OS and software are not infected by viruses or Trojan horses.

In this paper, we propose a framework for reporting assurance of business entities and the computer platforms that act on behalf of the entities. The following sections are organized as follows. Section 2 discusses the design of the WS-Assurance framework. Section 3 shows examples of business assurance technologies. Section 4 discusses the platform assurance in detail. Section 5 discusses potential use cases. Section 6 discusses related technologies. Section 7 concludes the paper. Appendix A shows bindings to WS-Security framework.

2 WS-Assurance Framework

The basic idea of WS-Assurance is that a Web service advertises its assurance information as a part of its non-functional description, and also provides a means to express and communicate a set of evidence which supports the assurance description.

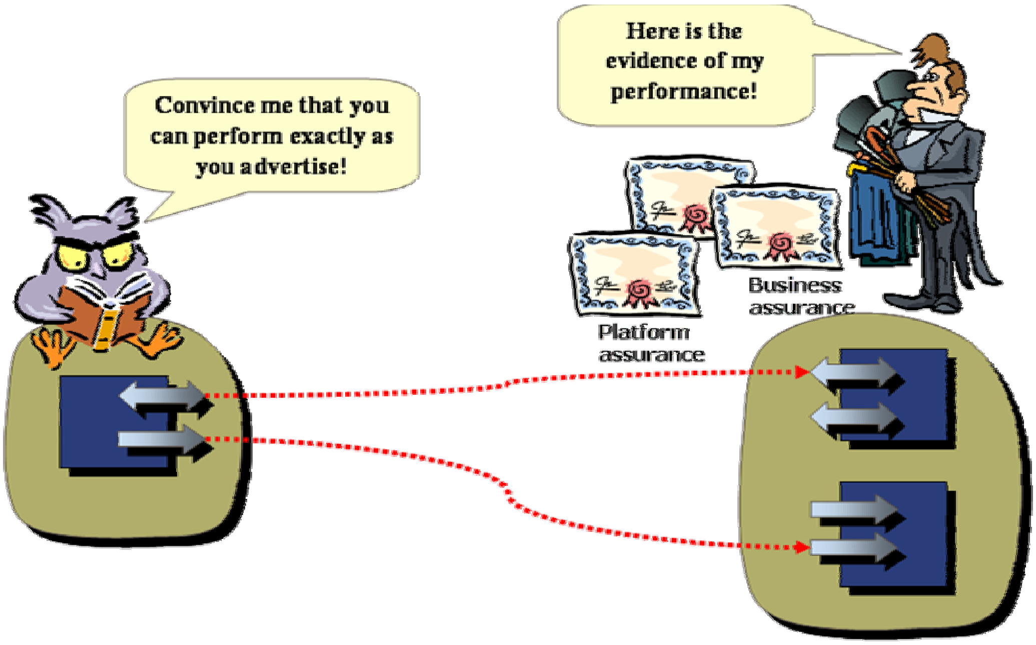

Convince me that you can perform exactly as you advertise!
Here is the evidence of my performance!
Business assurance
Platform assurance

Figure 1 WS-Assurance Framework

What kind of description and evidence is needed for a business to make intelligent decisions whether the Web service can be relied upon? First, the company who provides the service must be trustworthy. We call this *business assurance*. Business assurance represents various aspects of trustworthiness of a business entity by using ordinary business infrastructure. For example, a business assurance may be represented as:

- Business contracts
- Financial statements
- Third-party ratings, and
- Insurance coverage.

Second, the Web service must be running on trustworthy infrastructure. We call this *platform assurance*. Platform assurance represents trustworthiness of a computer platform that is acting on behalf of a business entity, by providing evidence that the service is actually running in a trustworthy environment. This information should include:

- The detailed system configuration, such as make and model of the hardware, network configuration, OS version and its configuration, and middleware versions and configuration;
- Certifications of software components (e.g., Common Criteria), if any;
- Certification of the vendor who developed the application software, if any; and

- Security policies / system management policies.

2.1 Direct Trust Model

We review two different communication models: the *direct trust model* and the *brokered trust model*, and then discuss the elements that compose the WS-Assurance framework. Figure 2 shows the Direct Trust Model of the WS-Assurance in which an entity directly provides a set of descriptions and evidence of assurance to the other party. In this model, the client evaluates the received information itself, without reference to a third-party.

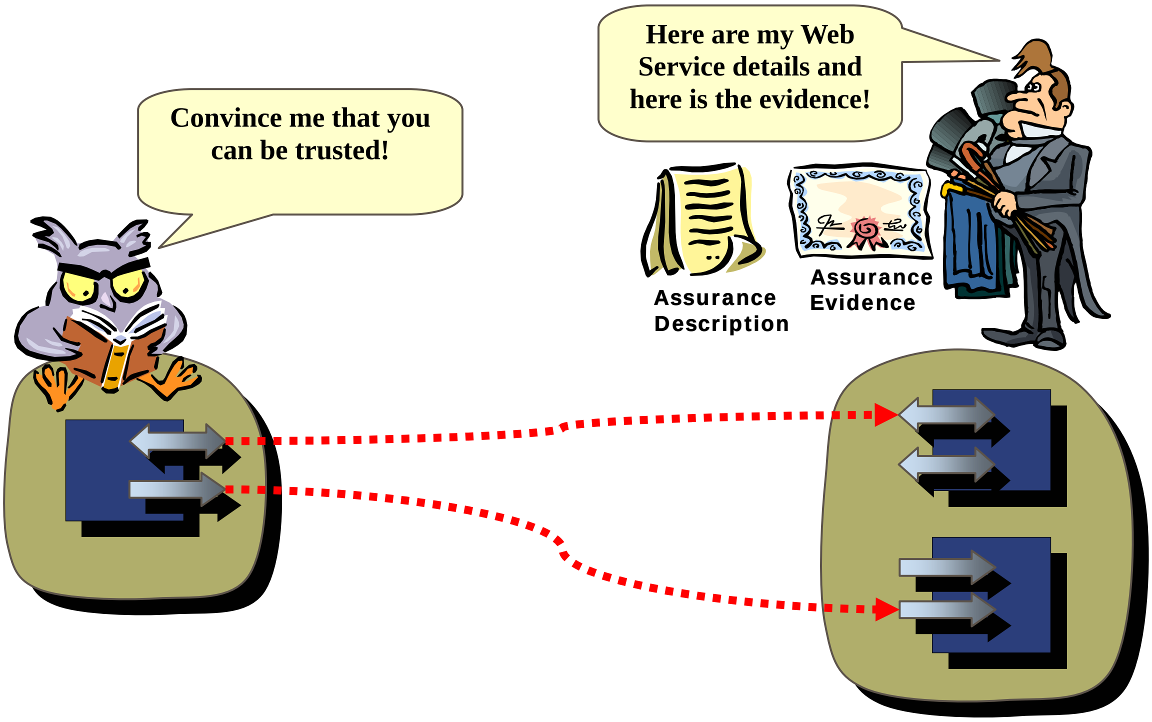

Convince me that you can be trusted!
Here are my Web Service details and here is the evidence!
Assurance Description
Assurance Evidence

Figure 2 Direct Trust Model

The communication consists of two important elements: the *description* and the *evidence*.

2.2 Brokered Trust Model

It may be too burdensome for each entity to evaluate adequacy of the assurance information received from the service. For example, a client may not be able to make a proper judgment on, for example, whether a financial statement or a particular version of the OS can be trusted. Instead, we envision that there will be third party services which provide a much simpler index that represents the level of trust (such as “Green”, “Yellow”, “Orange”, and “Red”). For example, an IT security service provider might assert that a particular combination of OS and middleware versions, with a particular set of configurations, has no known vulnerabilities. Thus the target platform is in the state of “Green.” The client may therefore delegate inspection of assurance information to a third party. We call this *Brokered Trust Model,* whereby a trusted third party makes such decisions. The Brokered Trust Model works either in the *push* or *pull* mode as discussed below.

2.2.1 The Push Mode of the Brokered Trust Model

Figure 3 shows the brokered trust model in the push mode. In this model, a Web Service sends its assurance description to a trusted third party who is responsible for evaluating the description and returns a response in a form of a certificate. The client then validates the certificate.

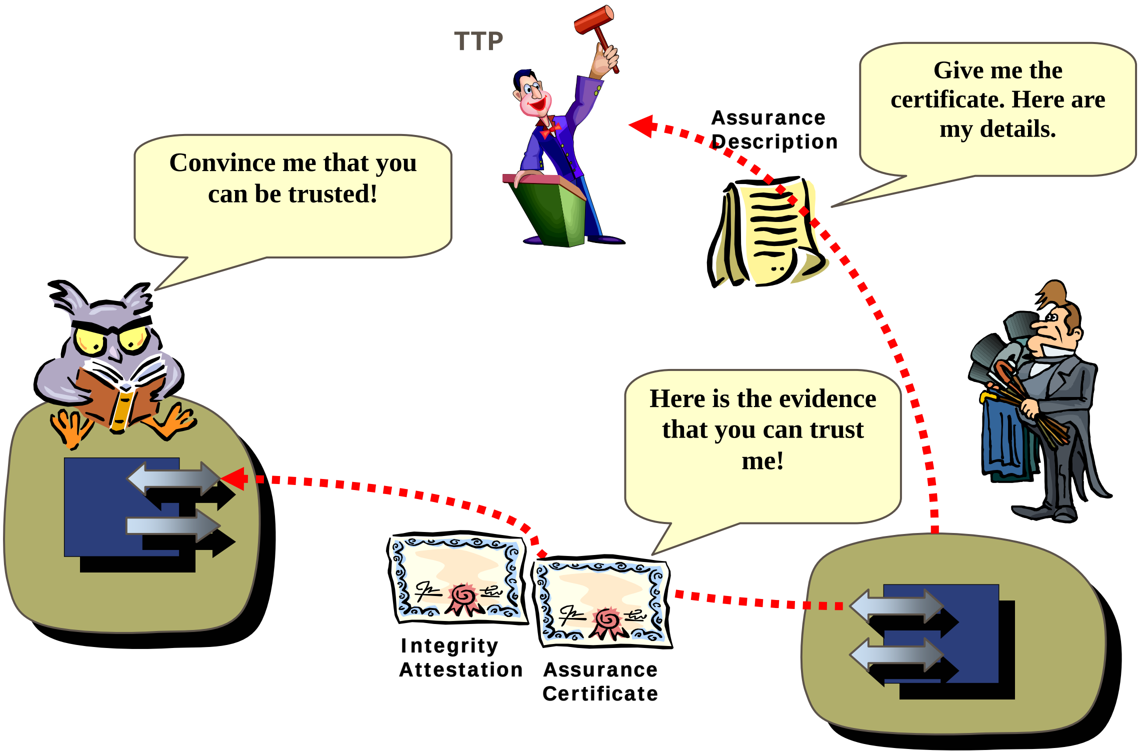

TTP
Convince me that you can be trusted!
Assurance Description
Give me the certificate. Here are my details.
Here is the evidence that you can trust me!
Integrity Attestation
Assurance Certificate

Figure 3 Brokered Trust Model in the Push Mode

Note that if the certificate regards the platform configuration, it needs to be accompanied by the attestation signature when it is sent to the client, so malicious software cannot replay an old certificate after the platform is compromised (See Section 4 for the details). The protocol may also include a challenge-and-response to avoid potential replay attacks; i.e., the client sends a nonce to the platform when requesting the assurance, which is returned along with the platform configuration certificate and signed by the attestation signature.

2.2.2 The Pull Mode of the Brokered Trust Model

The pull mode of the brokered trust model works in a similar way as the direct trust model, except that the client defers to a trusted third party's trust decision. (See Figure 4).

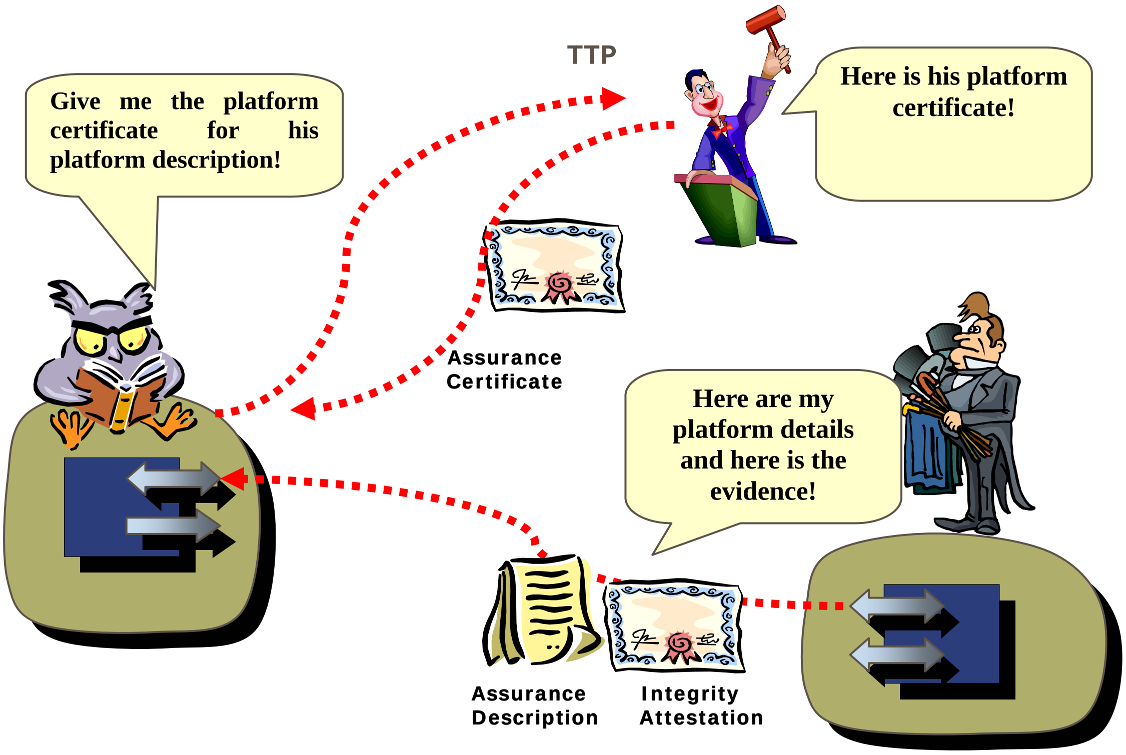

TTP
Give me the platform certificate for his platform description!
Here is his platform certificate!
Assurance Certificate
Here are my platform details and here is the evidence!
Assurance Description
Integrity Attestation

Figure 4 Brokered Trust Model in the Pull Mode

Compared to the push model, this model mitigates the problem of outdated certificates (i.e., the service's status described in the certificate changes before the certificate expires, thus the trustworthiness of the contents is no longer certain) by using the online verification of the assurance description. On the other hand, the pull model may suffer from the performance bottleneck of the trusted third party because for every transaction the trusted third party needs to be contacted. Techniques such as distributed third party implementations and intelligent caching mechanisms may be used to reduce the performance bottleneck.

2.3 Relationship to the WS-Security Model

The above model is essentially a direct application of the WS-Security model as described in the WS-Security Roadmap paper[8]. Assurance descriptions and/or assurance certificates should be treated as security tokens, which can have various different syntaxes. Appendix A provides a sample platform description and attestation and mapping to WS-Security specifications.

3 Business Assurance

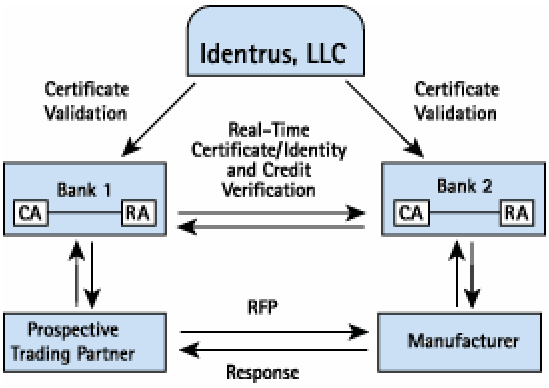
Business assurance consists of policies or rules that will govern the business operations. Sometimes the set of rules is well defined and accepted by a large collection of businesses, as in 3.1. Sometimes, however, the set of rules will need to be negotiated before a transaction can occur. Understanding and abiding by the rules is necessary for building trust. A trusted third party is sometimes established to attest to the trustworthiness of a business. The policies often refer to identity assurance, liability, and privacy practices. Technology assurance alone cannot create the necessary levels of trust needed for business transactions.

3.1 Identrus and Eleanor – Identity and Transaction Assurance

Identrus (www.identrus.com) is a Limited Liability Company that provides identity assurance to financial institutions. Identrus is comprised of 60 international member banks and provides a framework for trust based on the trust relationship between banks and their customers. Identrus establishes a set of operational rules that defines the framework for creating legally binding contracts.

Eleanor is a payment initiation scheme that builds on top of Identrus' identity services. Along with the technical specifications for transactions there is also a set of Operational Rules that define services levels, rights, obligations and business practices required of all participants. These rules govern the legality of the transactions and can in some cases guarantee payment.

The chart shows a sample message flow for an Eleanor payment initiation in which a manufacturer contracts with a prospective subcontractor, when each have the backing of separate Identrus affiliated financial institutions.



During this transaction, each party advertises its capability and willingness to do the business. The operational rules of Identrus and Eleanor can be seen as the business assurance component of WS-Assurance.

3.2 Insurance

Insurance is one of the most common strategies to mitigate business risks. Even business processes provided as Web services should be subjects of insurance. Businesses need to manage risks by first performing a risk assessment. Then they need to see which risks can be reduced by applying technologies to avoid the risk. They also need to understand which risks they are willing to assume. For the risks that can not be covered by the previous choices insurance can be a solution. e-business insurance products are being provided by firms such as Insuretrust (http://www.insuretrust.com). A description plus proof of insurance coverage constitute a level of assurance for the Web service.

3.3 Business Ratings

Less quantitative, but still important information on judging trustworthiness of a business is a third party rating. There are commercial business rating services, such as OpenRatings (http://www.openratings.com/).

4 Platform Assurance

This section discusses three elements that constitute platform assurance; the description of the platform, the proof that the platform is in fact running with the described configuration (which we call *attestation* after the TCG terminology), and the platform configuration certificate that vouches for trustworthiness of a particular platform configuration. Note that a Web service is hosted by a complex site in the real world. Thus, the “platform” here may consist of multiple hardware boxes such as routers, load balancers, multiple application servers, multiple backend servers, a directory server, an administration console, and so on.

4.1 Platform Description

A platform description is a set of attributes that describe a configuration of a particular computer platform instance. The following list shows a typical structure of a platform description.

- Platform Hardware - Make, model, serial No.
- Operating System- name, provider, version, build No.
  - Modules – kernel, device drivers
  - Configuration – OS configuration details, device driver configuration details
- Runtime Software – name, provider, version, build No.
  - Configuration – Runtime configuration details
- Network Configuration
  - Routers, firewalls, IDS, …
- Operating policies

4.2 Integrity Attestation

In order to ensure that a platform description is accurate, the software stack that composed the platform description needs to be trustworthy. This section discusses various technologies for proving the integrity of the platform.

4.2.1 TCG-Based Attestation

The Trusted Computing Group (TCG) [6] defines secure hardware architecture for ensuring integrity of the platform.

The TCG specification [7] defines a tamper-resistant hardware module called a Trusted Platform Module (TPM) that includes Platform Configuration Registers (PCRs). These are special registers used for measuring platform integrity. In the TCG architecture, the platform measures the hash value of each software stack while it is booting. During the boot sequence, each component measures the next component (e.g., BIOS measures the boot loader, the boot loader measures the OS), and *extends* the PCR with the hash value of the next component; i.e., after extended with a new hash value, the new value in a PCR will be the hash of the concatenation of the old PCR value and the new hash value. The operations on PCRs are protected by the hardware, thus the values in PCRs cannot be altered arbitrarily. Therefore, a particular PCR value (i.e., integrity measurement) proves that a certain set of software components is running on the platform.

The TPM also defines an operation called *quote*, by which the TPM signs a concatenation of the PCR values and an arbitrary value using a special key called the Attestation Identity Key (AIK). An AIK is a signature key that is generated and securely stored in a TPM, which is bound to a particular instance of the TPM through a trusted third party. Therefore, an attestation signature performed by the *quote* operation proves that the signature is done by a particular instance of the platform that is running a particular set of software components.

Platform assurance takes advantage of the integrity attestation mechanism of the TCG by utilizing an attestation signature on the platform description. In addition, a platform description may also describe which components were measured in what order, that gives some degree of flexibility to the recipient while verifying the integrity measurement value.

4.2.2 Other Attestation Techniques

The integrity management solution of Tripwire monitors changes to systems and configuration files by comparing them with a recorded snapshot, and may then notify an administrator when changes are detected. Since the solution is intended for centralized management of computer platforms in a trusted enterprise network environment, the design does not support platform attestation between two parties which has no established trust relationships.

Antivirus software is the most commonly used tools for detecting compromised software platforms. Most of the currently available antivirus systems use virus-pattern database and are capable of detecting known viruses only. Other type of antivirus systems work better by additionally monitoring suspicious behavior, but such software depends upon users to make security decisions whether it is a virus or a false alarm. It is difficult to detect intrusion if the virus-pattern, database or the antivirus system itself is compromised.

Signed-code technologies, such as signed Java archive (JAR) files or Windows Authenticode, are used to detect compromised software package before installation. However, the technology is not capable of detecting compromised software after installation. It also puts the platform on the risk if vulnerable code is signed, since the once trusted code is allowed to perform security sensitive operations.

Various standards have been defined to provide criteria for security of computer software and hardware. The Common Criteria Certification is a set of internationally recognized criteria for evaluating the security of IT products, and is adopted by 16 countries as the security standard. Capability Maturity Model for Software (SW-CMM) is a de facto standard model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes. BS7799, or the British Standard 7799 Information Security Management System Certificate, is a set of internationally recognized security evaluation criteria. Platform assurance can leverage these standards by utilizing them as evidence of trustworthiness of each component. In addition, it provides two advantages. First, the TCG mechanism can verify integrity of software components to assure that they are not compromised. Second, platform assurance can aggregate assurance of each component to provide assurance of an entire platform.

4.2.3 Attestation of Distributed System

A Web service platform usually consists of multiple hardware boxes, such as routers, firewalls, load balancers, application servers, and so on. The client may not be interested in the details of each one of them; rather, the client should be interested in the integrity of the Web service's infrastructure as a whole. It is desirable that we can provide a simple quantitative measure that represents the level of integrity of the infrastructure. At this moment, there is not such a measure; this is an important technical challenge for the Research.

4.3 Platform Configuration Certificate

A Platform Configuration Certificate represents a trusted third party's belief in the trustworthiness of a particular platform description. The certificate may also include the expected PCR values that are intended to match a particular configuration described by the platform description.

A Platform Configuration Certificate shall include attributes that express degree of trustworthiness. For example, trustworthiness may be expressed in a binary decision of yes or no, or multiple levels such as low, mid, and high. For example, if the PCR values match the platform description, and all the components in the current configuration are known to be trusted, then trustworthiness could be set as "high". If the PCR values do not match the platform description at all, then platform might be compromised by viruses, and the trustworthiness could then be set as “low”. If the PCR values match the platform configuration, but some components in the current configuration have known vulnerabilities, it is possible that the platform will be compromised sometime in the near future, and the trustworthiness could be “mid”.

Different levels of trustworthiness may be given to different functions of the platform. For example, if only a particular middleware on the platform is compromised, all the services that use the middleware cannot be trusted, while others might remain secure.

5 Use Cases

This section briefly discusses potential applications of the WS-Assurance.

Web Services Security Token Services (STS)

The WS-Trust specification [18] defines a service for exchanging issuing and validating security tokens for web services. This service is known as a Security Token Service (STS). The Security Token Service can be part of your trusted domain or be run by a trusted party. The security of the STS is critical to the overall security of Web Services. If this service is compromised then the security of every web service that relies on the STS is compromised. Business assurance is needed to establish an STS as a trusted party . Platform assurance can assure that the resources needed for securely handling security tokens has not been compromised.

Aggregated Trust Decision

A collection of evidence in business assurance can be used to judge the trustworthiness of an entity as an aggregation of evidence, thus allowing the decision less susceptible to the erroneous judgment of a trusted third party. One of the problems in the identity based trust establishment is that it is quite sensitive to vulnerabilities of PKI, since it relies on a single decision point. For instance, in January 2001, VeriSign Inc. issued two digital certificates to an individual who fraudulently claimed to be a representative of Microsoft Corporation [1616]. Until the incident was discovered – two months later – and the certificates revoked, the fraud digital certificate could have allowed the attacker to write malicious code and distribute it signed by “Microsoft Corporation”. In the current PKI, timeliness of certificate revocation is another challenge. By weighting trust decisions from each trusted third party, and by accumulating all the decisions in business assurance, there is more scope to avoid such attacks.

Remote Platform Management

The platform assurance can be integrated with a remote platform management mechanism to detect and update an old version of software that has known vulnerabilities. Arbaugh et al [14] proposed modeling the vulnerability life cycle, and pointed out that the rate of intrusion increases once a vulnerability is disclosed, and keeps increasing until a satisfactory security patch is applied. Integrity measurement technology provided by the platform assurance is crucial in detecting vulnerable or compromised software components at the earliest opportunity, especially in a distributed environment.

Service Level Adaptation

A client may advertise its business/platform assurance to a service provider, so that the service provider may differentiate the level of service depending on the level of assurance. For example, a service provider may limit the amount of the transaction depending on the third-party rating in the business assurance.

Grid Computing

In the Grid computing, trustworthiness of each computational node is increasingly important. In the case of the SETI@home project [1515], the efficiency of parallel computation was spoiled by verbose computations, because there are malicious nodes that return incorrect computation results. In the case of commercial Grid computing service in which a single node provides services to multiple customers, it is important to ensure that an application for a customer is not affected by other applications.

On-Demand Computing

Trustworthiness of Web-Services has been becoming more important as it is widely adopted for integrating out-sourced business processes in the On-Demand computing model. Each Web-Service's trustworthiness can be judged by the trustworthiness of the service provider and the platforms that provide the service; i.e., business and platform assurance.

6 Related Technologies

A series of Composite Capability/Preference Profiles (CC/PP) specifications published as W3C working draft specifications define a mechanism for describing and negotiating device capabilities. A W3C draft document [1112] defines the structure and vocabularies for describing device capabilities using Resource Description Framework (RDF). Ohto and Hjelm [13] define a protocol for sending CC/PP information using HTTP extension mechanism. Several additional profiles had been defined by extending CC/PP vocabularies to describe properties such as the hardware model and software versions. However, since CC/PP is rather intended for use in content adaptation, little security issues are explicitly covered; e.g., a malicious user may forge a capability description, which is difficult to detect by the other party.

TCG defines an architecture for assuring the platform identity and integrity as discussed in Section 4. The platform assurance proposal is built on top of the TCG specifications to assure integrity of the platform descriptions, while attempting to provide much finer grained information, so that it is easier for humans and machine to analyze the semantics of the platform configuration.

7 Conclusion

As need for trust increases in the Web-Services infrastructure, a flexible, fine-grained, and decentralized solution for assuring capability and reliability of services is increasingly important. The WS-Assurance framework, built around existing TCG and Web-Services Security technologies, proposes a mechanism to integrate various evidence and descriptions to assure trustworthiness of business entities and platforms.

Acknowledgements

The authors wish to thank Frank Seliger, Seiji Munetoh, Tim Ebringer, Mimi Zohar, Ray Valdez and other people of IBM Corporation for comments and insights on an earlier version of this paper.

References

1. W3C Recommendation 24 June 2003, SOAP Version 1.2 Part 0: Primer, SOAP Version 1.2 Part 0: Primer. URL: http://www.w3.org/TR/soap12-part0/
2. W3C Recommendation 24 June 2003, SOAP Version 1.2 Part 1: Messaging Framework. URL: http://www.w3.org/TR/soap12-part1/
3. W3C Recommendation 24 June 2003, SOAP Version 1.2 Part 2: Adjuncts, http://www.w3.org/TR/soap12-part2/
4. Web Services Description Language (WSDL) Version 1.2 Part 1: Core Language,

W3C Working Draft 11 June 2003. URL: http://www.w3.org/TR/wsdl12/.

5. UDDI Version 2 Specifications. URL: http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm#uddiv2.
6. Trusted Computing Group (TCG) URL: http://www.trustedcomputinggroup.org/.
7. Trusted Computing Group (TCG) Main Specification Version1.1b. URL: https://www.trustedcomputinggroup.org/downloads/.
8. IBM, Microsoft, Security in a Web Services World: A Proposed Architecture and Roadmap, http://www-106.ibm.com/developerworks/library/ws-secmap/, April 2002.
9. Web Services Policy Framework (WS-Policy), 28 May 2003. URL: http://www.ibm.com/developerworks/library/ws-policy
10. Web Services Policy Attachment (WSPolicyAttachment), 28 May 2003. URL: http://www-106.ibm.com/developerworks/webservices/library/ws-polatt/.
11. Web Services Federation Language (WS-Federation), 08 July 2003. URL: http://www.ibm.com/developerworks/library/ws-fed/.
12. Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies, W3C Working Draft 25 March 2003. URL: http://www.w3.org/TR/2003/WD-CCPP-struct-vocab-20030325/.
13. Ohto H., Hjelm J. CC/PP Exchange Protocol Based on HTTP Extension Framework, W3C Note 24 June 1999. URL: http://www.w3.org/TR/NOTE-CCPPexchange.
14. Arbaugh, W. A., Fithen, W. L., McHugh, J., Windows of Vulnerability: A Case Study Analysis. Pp.52-59, IEEE Computer, 2000.
15. SETI@Home. URL' http://setiathome.ssl.berkeley.edu/.
16. VeriSign Security Alert Fraud Detected in Authenticode Code Signing Certificates March 22, 2001.URL: http://www.verisign.com/developer/notice/authenticode/.
17. IBM, Microsoft, Secure Reliable Transacted Web Services: Architecture and Composition, Sept 2003. URL: www-3.ibm.com/software/solutions/webservices/pdf/SecureReliableTransactedWSAction.pdf
18. WS-Trust Specification, Draft 18 December 2002. URL: http://www-106.ibm.com/developerworks/library/ws-trust/
19. Web Services Security Policy (WS-SecurityPolicy), Draft 18 December 2002. URL: http://www.ibm.com/developerworks/library/ws-secpol/.
20. Web Services Secure Conversation (WS-SecureConversation), Draft 18 December 2002. URL: http://www.ibm.com/developerworks/library/ws-secon/.
21. Web Services Reliable Messaging Protocol (WS-ReliableMessaging), 13 March 2003. URL: http://www.ibm.com/developerworks/library/ws-rm/.

22. Web Services Coordination (WS-Coordination) , 9 August 2002. URL: http://www-106.ibm.com/developerworks/webservices/library/ws-coor/.

23. Web Services Transaction (WS-Transaction), 9 August 2002. URL: http://www.ibm.com/developerworks/library/ws-transpec/

24. Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1 4, Committee Specification, 27 May 2003. URL: http://www.oasis-open.org/committees/documents.php?wg_abbrev=security

Appendix A: WS-Security Bindings for Platform Assurance

This section shows one possible representation of the platform assurance based on the WS-Security framework. The syntax of the platform description is nothing more than a sample; it is not our intention to propose a specific syntax at this moment.

Platform Description

A platform description could be represented in an XML element (List 1) and communicated using the standard techniques for meta data exchange, such as attaching to an entity via WS-PolicyAttachement [10]. It could also be exchanged by the Attribute Service mechanism defined in the WS-Federation specification [11].

List 1 One Possible Syntax of Platform Description

```
<wsp:Policy>
  <PlatformDescription>
    <Platform make="IBM" model="8688-4RX" SerinalNo="xxxx-xxxx">
      <PlatformOptions> … Platform options </PlatformOptions>
    </Platform>
    <OperatingSystem Name="Windows XP" Version="5.1"
      Build="2600xpsp2.030422-1633">
      <Modules> … Kernel modules / device drivers </Modules>
      <OSConfig> … OS Config details </OSConfig>
    </OperatingSystem>
    <Runtimes>
      <Runtime Name="WebSphere" Version="5.02" Build="..">
        <RTConfig> … Runtime Configuration details </RTConfig>
      </Runtime>
       … Other runtime libraries / daemons /  …
    </Runtimes>
    <ManagementPolicies> … </ManagementPolicies>
  <PlatformDescription>
</wsp:Policy>
```

Attestation Signature

An attestation signature can be integrated into the WS-Security framework by adding a new signature algorithm that uses the PCR value, and using that signature algorithm with the XML Digital Signature. The new signature algorithm can be derived from the TPM_Quote operation of the TCG specification. In this new algorithm, the signature value is defined as a concatenation of the TCG_QUOTE_INFO and a signature output from the TPM_Quote operation. The TCG_QUOTE_INFO is a structure defined in the TCG specification which includes a hash of PCR values and an arbitrary 160 bit data. The TPM_Quote operation generates a signature against a TCG_QUOTE_INFO using an AIK. List 2 shows an example of the attestation signature in a SOAP message.

List 2 Attestation Signature

```
<S:Envelope ...>
<S:Header>
 <wss:Security>
  <ds:Signature>
   <SignedInfo>
    <CanonicalizationMethod Algorithm="..."/>
     <SignatueMethod
            Algorithm="http://trustedcomputinggroup.org/2002/08/rsa_pcr#"/>
   </SignedInfo>
   <SignatureValue>..signature value & hash of PCRs..</SignatureValue>
  <ds:Signature>
 </wss:Security>
</S:Header>
 …
</S:Envelope>
```

Platform Configuration Certificate

A platform configuration certificate is a kind of security token and can be represented as any of generic assertion languages, such as X.509 attribute certificates and SAML assertions. A certificate in a SAML [24] assertion is exemplified in List 3. The certificate contains a list of PCR values that can be trusted for a given platform description.

List 3 Platform Configuration Certificate

```
<saml:Assertion …
  AssertionID="128.9.167.32.12345678"
  Issuer= " IBM Corporation"
  IssueInstant="2002-12-03T10:02:00Z">
  <saml:Conditions
   NotBefore="2002-12-03T10:00:00Z" NotAfter="2002-12-03T10:05:00Z">
    <saml:AudienceRestrictionCondition>
      <saml:Audience>http://…/<saml:Audience>
    </saml:AudienceRestrictionCondition>
  </saml:Conditions>
  <saml:AttributeStatement>
    <saml:Attribute AttributeName="PlatformDescriptionURI" AttributeNamespace="">
      <saml:AttirbuteValue>
         <PlatformDiscription URI= " http://www.fabrikam123.com/platformdesc " />
         <TrustedPCRValues> … List of possible PCR values of the trusted configuration
         </TrustedPCRValues>
      </saml:AttributeValue>
    </saml:Attribute>
  </saml:AttributeStatement>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    … Digital Signature by IBM
  </ds:Signature>
</saml:Assertion>
```

SOAP Message with the Platform Assurance

List 4 shows an example SOAP message that includes a certificate for the AIK, an integrity attestation, a platform configuration certificate, and a reference to a platform description. Note that the certificates are security tokens and thus they can be carried in the <wsse:Security> header block.

List 4 A SOAP Message with the Platform Assurance

```
<S:Envelope>
 <S:Header>
  <wsse:Security>
   <wsse:BinarySecurityToken>
     Certificate for the AIK, issued by a CA
   </wsse:BinarySecurityToken>
   <ds:Signature>
    <ds:SignedInfo>
     <ds:Reference> … both the platformCert and the body must be signed </ds:Reference>
     <SignatueMethod Algorithm="http://trustedcomputinggroup.org/2002/08/rsa_pcr#"/>
    </ds:SignedInfo>
    <ds:SignatureValue> … Contains PCR …</ds:SignatureValue>
   </ds:Signature>
   <saml:Assertion AssertionId="platformCert">
     Platform configuration certificate, signed by platform vendor
     Contains pointer to platform desc + PCR
   </saml:Assertion>
 </S:Header>
 <S:Body Id="body"> … Payload of the message, signed and encrypted … </S:Body>
</S:Envelope>
```