

**SECURING SOA AND WEB  
SERVICES FOR JUSTICE  
INFORMATION SHARING**



**IJIS Institute**

*SECURITY AND PRIVACY WHITE PAPER*

## **ACKNOWLEDGEMENTS**

---

The IJIS Institute would like to thank the following individuals and their sponsoring companies for their dedication and input to this document:

Jim Cabral, <i>Committee Chair</i>	<i>MTG Management Consultants</i>
Chuck Georgo	<i>Nowhere To Hide</i>
Alan Harbitter	<i>Nortel</i>
Jim Harris	<i>National Center for State Courts</i>
Rob Kribs	<i>Analysts International</i>
Susan Laniewski	<i>SAL Consulting</i>
Jim Pingel	<i>Wisconsin OJA</i>
Tim Skinner	<i>SRA International</i>
Bob Slaski	<i>Nlets</i>
Monique La Barre	<i>Institute for Intergovernmental Research (IIR)</i>
Scott Came	<i>SEARCH – The National Consortium for Justice Information and Statistics</i>

The IJIS Institute would also like to thank the U.S. Department of Justice (DOJ) Office of Justice Programs (OJP) Bureau of Justice Assistance (BJA) and the members of the Global Security Working Group (GSWG) for their comments and feedback.

This project was supported by Grant No. 2007-RG-CX-K021 awarded by the Bureau of Justice Assistance. The Bureau of Justice Assistance is a component of the Office of Justice Programs, which also includes the Bureau of Justice Statistics, the National Institute of Justice, the Office of Juvenile Justice and Delinquency Prevention, and the Office for Victims of Crime. Points of view or opinions in this document are those of the author and do not represent the official position or policies of the United States Department of Justice.

## TABLE OF CONTENTS

---

Acknowledgements .....	i
Table of Contents .....	ii
List of Figures .....	iv
List of Tables.....	iv
Section 1. Introduction .....	1
Section 2. Security Requirements of the JRA.....	2
JRA Security Requirements.....	2
<i>Service Authentication</i> .....	2
<i>Service Availability</i> .....	2
<i>Service Consumer Authentication</i> .....	2
<i>Service Consumer Authorization</i> .....	2
<i>Message Addressing</i> .....	2
<i>Message Confidentiality</i> .....	3
<i>Message Integrity</i> .....	3
<i>Message Non-repudiation</i> .....	3
Additional Justice-specific Security Requirements .....	3
<i>User credentialing and authorization</i> .....	3
<i>Auditing</i> .....	3
Section 3. The Current State of Standards for Web Services Security .....	4
XML Signature.....	4
XML Encryption .....	4
XML Key Management Specification (XKMS) .....	5
WS-Security .....	5
WS-Trust.....	6
WS-SecureConversation.....	6
WS-Federation .....	6
WS-Reliability, WS-ReliableMessaging, and WS-Policy .....	7
WS-I Basic Security Profile.....	7
SAML (Security Assertion Markup Language).....	8
XACML (eXtensible Access Control Markup Language) .....	8
Conclusion.....	9
Section 4. The Current State of the Industry for Web Services Security Tools.....	11
XML Security Gateways.....	11
Stand-Alone Web Service Security Services .....	11
Web Services Platforms .....	12
Web Services Management and Monitoring .....	12
Web Services Authentication.....	12
Software Development Tools .....	12
Conclusion.....	13
Section 5. Minimum Acceptable Practice in Securing Web Services .....	14
Minimum Practice to Meet Mandatory Security Requirements .....	14
<i>Service Authentication</i> .....	14
<i>Service Consumer Authentication</i> .....	14

<i>Service Consumer Authorization</i> .....	15
<i>Message Confidentiality</i> .....	15
<i>Service Availability</i> .....	15
<i>User Credentialing and Authorization</i> .....	16
<i>Auditing</i> .....	16
Minimum Practice to Meet Additional Security Requirements.....	16
<i>Message Addressing</i> .....	16
<i>Message Integrity</i> .....	16
<i>Message Non-repudiation</i> .....	16
Minimal Practice Limitations.....	17
<i>Using HTTPS, security only spans the session level</i> .....	17
<i>Relying on the application for security is not the best strategy</i> .....	17
<i>More advanced security requirements are not accommodated</i> .....	17
Section 6. A Target Security Architecture for Web Services .....	19
Shifting the Burden to the Enterprise Infrastructure.....	19
Recommended Standards and Specifications Addressing JRA Security Requirements .....	19
<i>Service Authentication</i> .....	19
<i>Service Consumer Authentication</i> .....	20
<i>Service Consumer Authorization</i> .....	20
<i>Message Confidentiality</i> .....	20
<i>Service Availability</i> .....	20
<i>User Credentialing and Authorization</i> .....	21
<i>Auditing</i> .....	21
Recommended Practices to Meet Additional Security Requirements .....	22
<i>Message Addressing</i> .....	22
<i>Message Integrity</i> .....	22
<i>Message Non-repudiation</i> .....	22
Conclusion.....	22
Section 7. Implementation Examples .....	24
Case Study: Ohio’s OLLEISN Project .....	24
<i>OLLEISN’s Web Service Security Requirements</i> .....	24
<i>OLLEISN’s Approach to Fulfilling Additional Security Requirements</i> .....	25
Case Study: Wisconsin’s WIJIS Justice Gateway Project .....	25
<i>WIJIS’ Implementation of Web Service Security Requirements</i> .....	26
<i>WIJIS’ Approach to Fulfilling Additional Security Requirements</i> .....	27
<i>WIJIS and the WS-* Standards</i> .....	27
Section 8. Summary and Conclusions .....	28
Glossary.....	31

## LIST OF FIGURES

---

Figure 1. WS-Security Specifications.....	9
Figure 2. A Minimum Acceptable Secure Web Services Configuration.....	18
Figure 3. An Orchestrated Web Service with Separate HTTPS Sessions between Service Providers .....	18
Figure 4. Example of a Target Security Architecture .....	28

## LIST OF TABLES

---

Table 1. Web Services Security Standards.....	10
Table 2. Specifications and Standards addressing JRA Requirements.....	23
Table 3. Contrasting Minimum Acceptable and Target Architectures.....	30

## SECTION 1. INTRODUCTION

---

There is increasing interest in, and use of, Web services in the justice domain. This is most prominent in projects requiring integration of loosely-coupled distributed systems and is increasingly becoming a part of business critical functions. Web services expose functionality through HTTP, circumventing typical firewall rules and other traditional forms of systems security. This exposure poses a much greater threat than was present in prior generations of distributed technologies, which typically used proprietary schemes and restrictive firewall rules to limit and control access.

This IJIS Institute white paper is intended to provide practical guidance on a standards-based approach to building *secure* Web services applications considering budget, in-place systems, and state-of-the-industry technology constraints. It also serves as an update to a previous Global Security Working Group (GSWG, or “Global”) whitepaper<sup>1</sup> on a similar theme that was also contributed to by IJIS members.

Recommendations conveyed in this paper are based on security requirements of the Justice Reference Architecture (JRA). Section 2 introduces the JRA along with security requirements for JRA service interaction profiles.

Section 3 provides an overview of current industry standards relating to Web services security. A brief explanation of each standard is included, along with a summary of the WS-Security family of specifications promoted by the Web Services Interoperability Organization (WS-I). Section 4 then covers

some of the many tools that are available to assist with implementing these standards.

Sections 5 and 6 present recommendations for minimum acceptable and target security architectures, respectively. The minimum acceptable practice focuses on a practical short-term approach given budget constraints, limitations of existing applications, and immaturity of standards. It presents basic security controls that must be in place to minimally support secure Web services while still meeting JRA security requirements. The target architecture presents an ideal approach using best practices and standards that addresses all aspects of JRA security requirements.

Section 7 presents some real-world implementation examples. In reality, these projects employ only some aspects of the recommendations. However, they are representative of early implementers that are beginning to address security requirements in a justice information sharing environment with goals that are consistent with the Justice Reference Architecture, and offer lessons for future projects on which to build.

Section 8 closes with a summary and conclusions. Our hope is that designers and architects of justice information sharing solutions will find this paper helpful as they embark on building and deploying secure Web services.

---

<sup>1</sup> “Web Services Security Issues in a Justice Environment,” Global Security Working Group, August 2003, [http://it.ojp.gov/documents/Web\\_Services.pdf](http://it.ojp.gov/documents/Web_Services.pdf).

## SECTION 2. SECURITY REQUIREMENTS OF THE JRA

---

The Global Infrastructure/Standards Working Group (GISWG) has adopted a service-oriented architecture (SOA) model for implementing justice information sharing systems, called the Justice Reference Architecture (JRA). The JRA Specification<sup>2</sup> defines the JRA as follows:

*“The JRA is a description of the important concepts in the justice domain and the relationships between those concepts. The JRA also identifies, at a high level, the kinds of “components” (software systems, hardware infrastructure, policies, practices, intersystem connections, and so on) necessary to bring those concepts to life in a particular context. The JRA is generally not specific enough to govern the implementation of any individual software system implementation. Rather, it is a framework for guiding implementations in general, with the aim of standardizing or harmonizing certain key aspects of those implementations to support reusability or interoperability.”*

---

---

<sup>2</sup>*“The Justice Reference Architecture (JRA) Specification,” Working Draft V 1.4, [http://it.ojp.gov/documents/20070214\\_jra\\_1\\_4\\_draft.pdf](http://it.ojp.gov/documents/20070214_jra_1_4_draft.pdf), Global Infrastructure/Standards Working Group, February 14, 2007.*

### **JRA Security Requirements**

The JRA Specification also defines specific functional requirements for services and non-functional requirements for service interaction profiles (SIPs) that describe frameworks for implementing the services defined in the JRA. The JRA service interaction profile requirements related to security include:

#### **Service Authentication**

The ability of a service to provide a consumer with information that demonstrates the service’s identity to the consumer’s satisfaction.

#### **Service Availability**

Service availability is a commitment of a service to be reachable at a stated percent of the time with specified conditions.

#### **Service Consumer Authentication**

Information provided with messages transmitted from service consumer to service that verify the identity of the consumer.

#### **Service Consumer Authorization**

Information provided with messages transmitted from service consumer to service that document the consumer’s authorization to perform certain actions on and/or access certain information via the service.

#### **Message Addressing**

Information provided in a message that indicates where the message originated, the ultimate destination of the message (beyond the physical end point), a specific recipient to whom the message should be delivered (this includes sophisticated metadata designed specifically to support routing), and a specific address or entity to which reply messages (if any) should be sent. While addressing is not explicitly a security requirement, the service

consumer and/or producer may require source/destination (i.e., addressing) anonymity typically implemented through cryptographic mechanisms.

### **Message Confidentiality**

Information provided in a message to prevent anyone except an authorized recipient from reading the message or parts of the message. Typically, Web services implementations address confidentiality from the standpoint of an entire message. For example, an entire SOAP XML message exchange session may be encrypted using SSL. However, justice services may require fine-grained confidentiality that encrypts subsets of a message with different keys. An example might be HIPAA-protected data that is included in an inmate record. Fine-grained confidentiality<sup>3</sup> may or may not be linked to privacy enforcement.

### **Message Integrity**

Information provided in a message to allow the recipient to verify that the message has not changed since it left the control of the sender.

### **Message Non-repudiation<sup>3</sup>**

Information provided in a message to allow the recipient to prove that a particular authorized sender in fact sent the message.

## **Additional Justice-specific Security Requirements**

In addition to the specific JRA requirements, additional justice-specific security requirements include:

### **User credentialing and authorization**

This is a means of identifying, authenticating, and determining the access privileges of the user requesting a service.

### **Auditing**

Justice services may require that a provider or the consumer track who/when/where information associated with services access and produce this information in response to an audit request after-the-fact. In the FBI Criminal Justice Information Services (CJIS), for example, this is a responsibility of the ORI subscribers. Justice Web services will require mechanisms to support auditing requirements.

---

While SIPs can be based on a number of underlying technologies including Web services, MQseries, and Java Messaging System, Web services provide the highest degree of vendor-independence and open standards content. As a result, Web services have appeal as a forward-looking implementation platform for SOA. This paper identifies best practices for securely implementing the JRA using the JRA Web services SIP<sup>4</sup>.

---

<sup>3</sup>We use the phrases "fine-grained confidentiality" and "non-repudiation" as technical terms-of-art. The reader should not associate any legal context with these phrases.

---

<sup>4</sup>"The Justice Reference Architecture (JRA) Web Services Service Interaction Profile Specification," Working Draft V 1.1, [https://it.ojp.gov/documents/WWS-SIP\\_Aug\\_31\\_version\\_1\\_1\\_FINAL\(3\).pdf](https://it.ojp.gov/documents/WWS-SIP_Aug_31_version_1_1_FINAL(3).pdf), Global Infrastructure/Standards Working Group, August 1, 2007.

## SECTION 3. THE CURRENT STATE OF STANDARDS FOR WEB SERVICES SECURITY

---

There are several security standards available today for securing Web services. Some of these standards are more mature than others, but it is obvious that Web service security has become a major issue. Over the past several years, Web services have evolved and have proven to be reliable in achieving interoperability and sharing data between applications. Successful data sharing has required standards be adopted not only for securing access to a Web service but also for securing the content of the data that is provided by a Web service.

The following is a list of Web service security standards. A brief description is given for each standard, as well as an assessment of the maturity of the standard.

---

### *XML Signature*

XML Signature is a World Wide Web Consortium (W3C) recommendation that defines XML syntax for digital signatures to provide message authentication. It is similar to the Public Key Cryptography Standard (PKCS#7) in functionality. XML Signatures can be applied to any digital content, including XML, and can be used to sign any part of the content or to sign the entire XML document.

The XML Signature standard is a mature standard, having been recommended by the W3C in 2002. It is used in various Web technologies such as SOAP and SAML. While XML Signature is an important component of secure Web services, it is not sufficient by itself to address all of the security/trust concerns and should, therefore, be used in conjunction with other Web service security standards.

The XML Signature standard directly relates to the JRA security requirement of *message non-repudiation*. An XML Signature also supports the JRA security requirement of *message integrity* by allowing the message recipient to verify that the message has not been altered since it left the sender.

For more information about the XML Signature standard, see <http://www.w3.org/TR/xmlsig-core/>.

### *XML Encryption*

The XML Encryption standard was recommended by the W3C in 2002. This standard specifies a process for encrypting data and representing the result in XML. The data encrypted may be an entire XML document, a single XML element, or even the content of an XML element. The XML Encryption standard is very flexible and supports many types of popular encryption schemes, including digital signing as specified in the XML Signature standard.

Transport Layered Security (TLS) is the accepted standard for secure communication over the Internet, and provides end-to-end encryption security. XML Encryption is not intended to replace TLS, but rather it provides a mechanism for security requirements not addressed by TLS. With XML encryption, you can encrypt parts of the data being exchanged instead of having to encrypt the entire message. This allows for secure data exchange between two or more parties. TLS allows for secure information between the two endpoints. With XML Encryption, an XML document can be securely shared among all parties who may receive the document.

The XML Encryption standard is a mature standard, having been recommended by the W3C back in 2002. It relates to the JRA requirement of *message confidentiality*, allowing only the recipient with the corresponding keys to decrypt the message. XML Encryption also relates to *fine-grained confidentiality*, since it allows you to encrypt parts of a message, or the entire message.

For more information about the XML Encryption standard, see <http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/>.

### ***XML Key Management Specification (XKMS)***

The XML Key Management Specification (XKMS) was recommended by the W3C and the Internet Engineering Task Force (IETF) in 2005. This standard specifies protocols for distributing and registering public keys suitable for use in conjunction with the XML Signature and XML Encryption standards also recommended by the W3C.

The XKMS consists of two parts: the *XML Key Information Service Specification (X-KISS)* and the *XML Key Registration Service Specification (X-KRSS)*. X-KISS allows a client to off-load part or the entire task of processing XML Signature elements to a separate XKMS service. The X-KRSS service allows for the registration and management of key pairs that are to later be used by an X-KISS service.

The XML Key Management specification is a mature standard, with version 2.0 having been recommended by the W3C in June 2005.

For more information about the XML Key Management Specification, see <http://www.w3.org/TR/xkms2/>.

### ***WS-Security***

The WS-Security standard is a set of SOAP extensions and was designed to enable applications to construct secure SOAP message exchanges. The standard, developed by Microsoft, IBM, and VeriSign, was submitted to OASIS, the international e-business open standards consortium. OASIS approved version 1.0 of WS-Security in 2004 and approved version 1.1 of this standard in February 2006.

WS-Security was designed to work with a wide variety of security models, such as TLS, PKI, and Kerberos. It provides support for multiple security tokens, trust domains, signature formats, and encryption technologies.

The WS-Security standard details a technical foundation for implementing security functions such as integrity and confidentiality in Web service messages. The standard was developed to address how message integrity and confidentiality can be enforced on Web service messaging to provide true end-to-end security. Point-to-point security can be enforced by using HTTPS in the transport layer, which does provide message integrity and confidentiality, but only between the original source and destination of a message.

It is recommended that WS-Security be used in conjunction with other Web service extensions to provide a more secure Web service application. WS-Security is a mature standard, with versions 1.0 and 1.1 having been approved as standards by OASIS in 2004 and 2006, respectively. The WS-Security standard meets many of the JRA security requirements; specifically *message addressing*, *service consumer authentication* and *authorization*, *service authentication*, and *message non-repudiation*.

For more information about the WS-Security standard, see [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbr=ev=wss](http://www.oasis-open.org/committees/tc_home.php?wg_abbr=ev=wss).

### **WS-Trust**

The WS-Trust specification was adopted by OASIS in February of 2005. Version 1.3 of WS-Trust is an OASIS committee specification as of November 2006. It specifies Web service extensions that build on the WS-Security specification to provide a framework for requesting and issuing security tokens, as well as specifying how to broker trust relationships. WS-Trust is designed to be used with the other WS-\* extensions, such as WS-Security, to provide a secure Web services environment.

In order to have a secure communication between two parties, the two parties must exchange security credentials, and each party must determine if they can “trust” the credentials of the other party. WS-Trust is a SOAP extension proposal that builds upon the WS-Security standard. It defines methods for issuing, renewing, and validating security tokens. It also defines ways to establish and broker trust relationships.

WS-Trust has been adopted by OASIS, but has not yet been ratified as a standard by that standards body. The WS-Trust standard meets many of the JRA security requirements, specifically *service consumer authentication*, *service authentication*, and *message non-repudiation*.

For more information about the WS-Trust specification, see <http://docs.oasis-open.org/ws-sx/ws-trust/200512/>.

### **WS-SecureConversation**

The WS-SecureConversation specification was submitted to OASIS in December 2005. It builds upon the existing WS-Security standard and the WS-Trust specification.

Where WS-Security is a message authentication model, WS-SecureConversation is a context authentication model, which defines how to establish a security context and exchange multiple secure messages in a secure conversation. It does this by defining a new WS-Security token type that is obtained using a WS-Trust binding.

Since WS-SecureConversation is used in conjunction with WS-Security and WS-Trust, it meets the same JRA security requirements as those standards, specifically *message addressing*, *service consumer authentication* and *authorization*, *service authentication*, and *message non-repudiation*, yet it meets these standards for a secured conversation, not just a secured message.

For more information about the WS-SecureConversation specification, see [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbr=ev=ws-sx](http://www.oasis-open.org/committees/tc_home.php?wg_abbr=ev=ws-sx).

### **WS-Federation**

Version 1.1 of the WS-Federation specification was released as a public draft in December of 2006. It is to be used for evaluation only. The specification was published by BEA Systems, Inc., BMC Software, IBM Corporation, Layer 7 Technologies, Microsoft Corporation, Novell Inc., and VeriSign Inc.

WS-Federation defines the mechanisms for different security domains to federate authorized access to resources in one domain to the trusted users in another domain. Essentially, with WS-Federation, when two domains set up a federated trust, the trusting domain is able to accept security tokens provided by the trusted domain for a user that has been authenticated in the trusted domain, and thereby able to authorize use of resources in the trusting domain to that user who was authenticated in the trusted domain.

There are two additional WS-Federation specifications that extend the original WS-Federation specification. The *WS-Federation Active Requestor Profile* defines mechanisms for requesting, exchanging, and issuing security tokens between active requestors such as SOAP-enabled applications. The *WS-Federation Passive Requestor Profile* defines how the WS-Federation model is applied to passive requestors such as Web browsers that support the HTTP protocol.

Though WS-Federation is not an OASIS standard, many of the above companies already have products supporting the specification. Microsoft's Windows 2003 Server R2 edition fully supports the specification and is able to provide federated identity management with other Windows 2003 R2 Servers, as well as with other platforms that support the WS-Federation Specification. WS-Federation addresses the JRA security requirements of *user credentialing and authorization*.

For more information on the WS-Federation specifications, see:

- ◆ <http://specs.xmlsoap.org/ws/2006/12/federation/ws-federation.pdf>.
- ◆ <http://www.verisign.com/wss/Active-Client-Profile.pdf>.
- ◆ <http://www.verisign.com/wss/Passive-Client-Profile.pdf>.

### ***WS-Reliability, WS-ReliableMessaging, and WS-Policy***

These three specifications, while not specific to Web service security, provide mechanisms for guaranteeing the reliability of a Web service conversation, as well as specifics for sharing Web service capability information. WS-Reliability, an OASIS standard, is a SOAP-based protocol for exchanging SOAP messages with guaranteed delivery, no duplicates, and a guaranteed message order. WS-ReliableMessaging specification is an

OASIS standard, covers much of the same ground as WS-Reliability and is recommended in the JRA Web Services SIP. It is a specification that provides an interoperable protocol used by a Reliable Messaging source and a Reliable Messaging destination to provide guaranteed delivery assurance that a message will be delivered. WS-Policy provides a way for Web services to express their capabilities, requirements, and general characteristics. These "policies" can further be attached to an access control mechanism like XACML (see page 8) for specifying which users have access to which features (policies) provided by a Web service.

For more information on these specifications, see:

- ◆ [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wsm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsm).
- ◆ <http://docs.oasis-open.org/ws-rx/wrm/200608/wrm-1.1-spec-cd-04.pdf>.
- ◆ <http://www-128.ibm.com/developerworks/library/specification/ws-polfram/>.

### ***WS-I Basic Security Profile***

The Basic Security Profile version 1.0 (BSP 1.0) was published in March 2007 by the Web Services Interoperability Organization (WS-I). The WS-I Profiles are guidelines that establish tests to ensure secure interoperability between Web services from multiple vendors. The Basic Security Profile is a superset of the WS-I Basic profile, and defines how to implement Web service security in a way that will be interoperable with other Web services that also implement the same profile.

The WS-I Basic Security Profile is an interoperability profile that addresses transport security, SOAP messaging security, and other security considerations for the WS-I Basic Profile. It defines the interoperability

requirements for implementing HTTP over TLS (a point-to-point technology), and the requirements for implementing SOAP Message Security which provide security protection for SOAP messages even across multiple intermediaries.

The WS-I Basic Profile and Basic Security Profile are guides that instruct how to implement the web service specifications to improve interoperability. On their own, the WS-I profiles do not meet any of the JRA security requirements.

For more information about the WS-I Basic Security Profile, see <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>.

### **SAML (Security Assertion Markup Language)**

SAML was developed by the Security Services Technical Committee of OASIS. Version 2.0 of the specification was approved as an OASIS standard on March 15, 2005. The SAML specification has been broadly implemented by all major Web access management vendors.

SAML is an XML-based framework for specifying authentication information about a user. It allows for assertions to be made regarding the identity, attributes, and entitlements of a user. These assertions are passed from one business entity, partner company, or application to another.

SAML provides an abstraction of the security framework, removing the details of any platform architecture or specific vendor implementation. This helps make security implementations more independent of the platform on which they are implemented. SAML enables single sign-on by allowing users to authenticate one time with an identity provider, and then re-use, or federate, that authentication across multiple services. This alleviates the burden of each service needing to perform duplicate authentication because

of the trust of the identity provider and the authentication in the provided SAML token.

SAML assertions are used to secure Web services by including them within the secured SOAP messages. By doing so, the security and identity information can be conveyed between actors in the Web service interactions. SAML assertion tokens are supported by the WS-Security and WS-Trust specifications.

SAML is the most widely implemented open standard for federated identity, and it provides the following JRA security requirements: *service consumer authentication*, *service authentication*, and *message non-repudiation*.

For more information about the SAML standard, see [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbr=ev=security](http://www.oasis-open.org/committees/tc_home.php?wg_abbr=ev=security).

### **XACML (eXtensible Access Control Markup Language)**

The XACML security specification was standardized by the OASIS standards organization. The current version 2.0 was ratified by OASIS in February 2005.

XACML is an XML-based language for specifying access control information. Similar to SAML, which provides syntax for authentication information, XACML provides syntax for authorization information. The standard describes both an access control language and a request/response language. The access control language allows you specify policies in XML for managing access to resources. The access control decision request/response language lets you query to ask whether or not an action should be allowed.

Like SAML, XACML is also one of the more mature standards identified in this document. It provides the *service consumer authorization* requirement of the JRA security requirements.

For more information about the XACML standard, see [http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbr=ev=xacml#XACML20](http://www.oasis-open.org/committees/tc_home.php?wg_abbr=ev=xacml#XACML20).

### Conclusion

Many security specifications have been developed in recent years. The standards bodies are aggressively trying to provide standards for security implementations that will be interoperable across platforms and applications. While these standards are detailed and complex, there are many tools that help simplify the implementation of these specifications by hiding the complexities.

The Web Services Interoperability Organization (WS-I), which is chartered to promote interoperability of Web services across operating systems, platforms, and programming languages, provides security specification that enable secure Web services. By adopting these specifications, you will be able to interoperate with other applications and services that have also implemented these specifications.

Figure 1 illustrates how the WS-Security specifications work together to provide a complete security model.

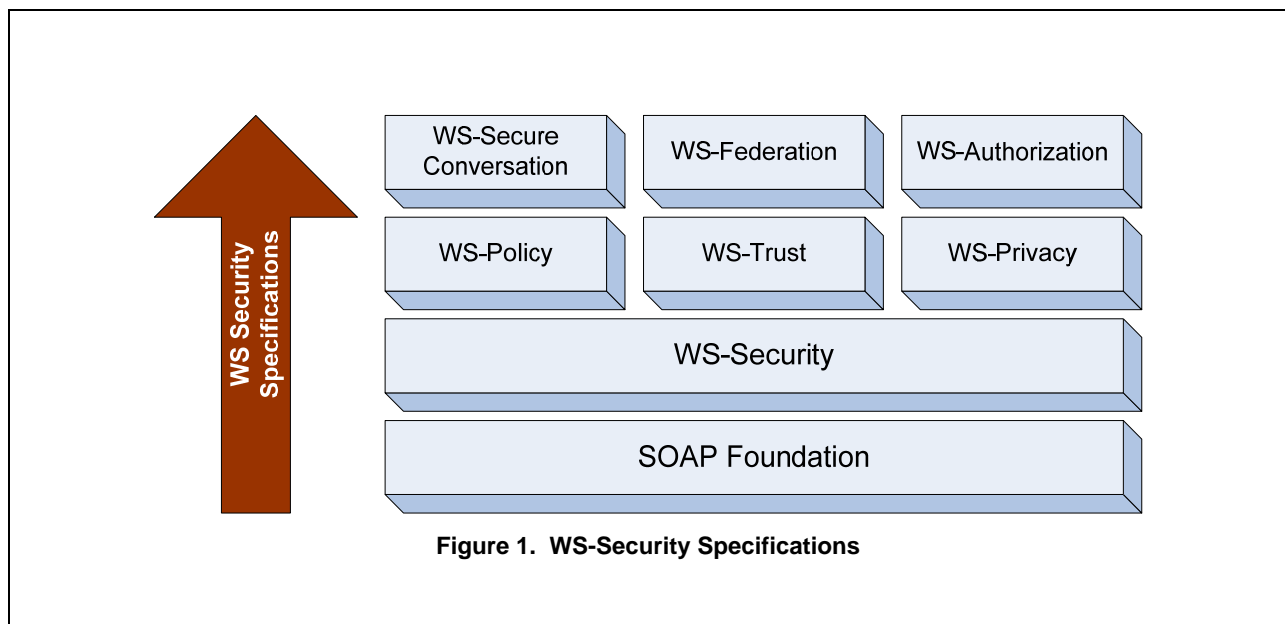


Figure 1. WS-Security Specifications

When a Web service receives a message, the SOAP envelope is processed to reveal a WS-Security container, which may contain containers for the other security-related standards. WS-Policy information will allow you to express which capabilities are provided by the service. You would include WS-Trust information when your Web service and another Web service want to share credentials and security information. If you have a long-running process between your service and the consumer, including WS-SecureConversation information provides an efficient way to pass a lot of data over a secure channel. You would include WS-Federation information, which requires WS-Policy, WS-Trust, and WS-Security information in order to exchange trusted user credentials for authenticating and authorizing remote users of your service. The WS-Authorization and WS-Privacy standards are part of the WS-Security family of standards, but have yet to be released.

Table 1 summarizes the standards detailed in this section.

Security Standard	Governing Organization / Status	Comments
XML Signature	W3C Recommendation	Widely accepted. Provides for digital signing of XML documents
XML Encryption	W3C Recommendation	Widely accepted. Provides for encrypting all or part of an XML document
XML Key Management Specification	W3C and IETF Recommendation	Newer. Used in conjunction with XML Signature and XML Encryption. Provides a standard for distributing and registering public keys
WS-Security	OASIS Standard	Widely accepted. Significant tool and vendor support. Standardizes how security is added to SOAP messages.
WS-Trust	OASIS Draft	Not yet adopted by OASIS, yet significant vendor support. Extends WS-Security to address interoperability between security tokens.
WS-SecureConversation	OASIS Draft	Builds on WS-Security and WS-Trust. Support from newer tools. Defines how to exchange multiple secure messages as part of a secure conversation.
WS-Federation	OASIS Draft	Widely accepted even though it is not yet a standard. Recent significant tool and vendor support. Provides for secure access of resources by federated identities.
WS-Reliability	OASIS Standard	Not widely adopted. Competing with WS-ReliableMessaging Standard. Model for ensuring reliable message delivery for Web services
WS-ReliableMessaging	OASIS Standard	Competing specification provided by Microsoft and IBM. Another model for ensuring reliable message delivery.
WS-Policy	W3C Recommendation	Fast-tracked at W3C. Fills gap in SOA allowing services to specify their capabilities and security policy requirements.
WS-I Basic Security Profile	WS-I Organization	Newly published by the WS-I to address interoperability of Web service security standards. Will be widely adopted because it defines how to implement a security model that will interoperate with other security implementations.
SAML	OASIS Standard	Widely Adopted. Large vendor support. Provides a way to specify authentication information about a user.
XACML	OASIS Standard	Newer, yet already widely accepted. Provides a syntax for Authorization information.

**Table 1. Web Services Security Standards**

With the needs and demands of information sharing ever increasing in the justice community, these standards can be used to secure applications, and to securely interoperate with other applications that use these same standards. The next section of this paper talks about the many types of tools that can be used to implement these standards.

## SECTION 4. THE CURRENT STATE OF THE INDUSTRY FOR WEB SERVICES SECURITY TOOLS

---

There are several categories of tools that are available to support development and implementation of Web service security. These categories range from development tools to hardware platforms and devices. Support for the Web Service Security standards mentioned in Section 3 of this paper continues to grow from tool vendors. WS-Security and SAML are the standards most widely adopted by the vendors, but support for the rest of the standards is increasing rapidly.

There are many different means to secure Web services. This section presents a list of the categories, including a description of the category and examples of some of the tools in the category. The tools listed are included to demonstrate industry support for the specifications described in this paper based on publicly available product information. The listing of a particular product in this paper should not be considered an endorsement of that product or a guarantee of standards conformance by the IJIS Institute.

---

### *XML Security Gateways*

XML Security Gateways, or XML firewalls, are hardware devices that sit on the network, similar to a firewall, and are used to offload the encryption and decryption of XML messages that conform to the WS-Security standard. These devices not only provide a security point for enforcing many of the security standards, but can also act as an XML accelerator in order to increase the number of messages that can be processed.

The following is a list of XML Security Gateways:

- ◆ Forum Sentry SOA Gateway
- ◆ IBM WebSphere DataPower XML Gateway
- ◆ Intel XML Security Gateway
- ◆ Layer 7 XML Firewall and Networking Gateway products
- ◆ Reactivity Gateway Appliance (just bought by Cisco)
- ◆ Vordel VS3000 Security Appliance

### *Stand-Alone Web Service Security Services*

This category refers to products that provide services required by some of the security standards. For example, the WS-Trust standard requires an authorized entity such as a Security Token Service (STS) to provide translation from one token type to another. Another example is stand-alone Policy Decision Point (PDP) products for the XACML standard.

- ◆ Forum Systems STS Network Appliance
- ◆ PingIdentity PingFederate (formerly PingTrust)
- ◆ RSA BSAFE Secure-WS
- ◆ Symlabs Policy Decision Point

## ***Web Services Platforms***

Most of the major software vendors (and many of the smaller ones) have platform products for supporting WS-Security, and many of the other available standards.

Examples of the Web Service Platform products that support WS-Security are:

- ◆ Server level platforms
  - BEA, IBM, Microsoft, SAP, Oracle WSM
- ◆ Enterprise service buses
  - Cape Clear, Fiorano, IONA, Tibco, Sonic, WebMethods Fabric
- ◆ Stand-alone platforms
  - Apache Axis, Systinet Server, WebMethods Glue

## ***Web Services Management and Monitoring***

There are many tools that can assist with the management and monitoring of the Web services. These tools all support the WS-Security standard, as well as several of the newer standards. Web Services Management and Monitoring tools typically allow you create a library of Web services available in your organization and assist with the policies for accessing these services, as well as logging access to these services.

- ◆ AmberPoint
- ◆ Blue Titan
- ◆ CA WSDM
- ◆ HP SOA Manager
- ◆ Oracle WSM
- ◆ Service Integrity

## ***Web Services Authentication***

The tools in this category support secure authentication across Web services, such as single sign-on, Web resource access management, and federated identities.

Products in this category include:

- ◆ CA eTrust Transaction Minder
- ◆ EnTrust Identification & Entitlement Server
- ◆ IBM Tivoli Federated Identity Manager
- ◆ Microsoft Windows 2003 Server R2
- ◆ Oracle COREid Federation
- ◆ PingIdentity PingFederate
- ◆ RSA Federated Identity Manager

## ***Software Development Tools***

Despite the trend in security being managed at the enterprise level by security administrators, application-level security is an important component of any security system and developers have a very large responsibility in securing their Web services. This category lists the tools and software development kits (SDKs) available to assist developers in complying with existing and emerging security standards.

- ◆ Apache WSS4J
- ◆ Microsoft Web Server Extensions (WSE) 3.0
- ◆ Microsoft .NET Framework 3.0
- ◆ PingIdentity SourceID
- ◆ Sun JWSDP
- ◆ VeriSign TSIK

## **Conclusion**

There are a large number of Web service security products available for use in software development. When practical, it is recommended to leverage existing products that support the appropriate security standards rather than attempting to develop custom solutions to support the standards.

At a minimum, implementers should consider installing an XML Security Gateway,

and picking a Web Service Management (WSM) product to manage your Web services. These types of tools help move some responsibility into the hands of security administrators.

However, developers still need to be aware of security issues and standards. Security standards are rather complex; using existing tools can help to simplify the development of secure Web services.

---

## SECTION 5. MINIMUM ACCEPTABLE PRACTICE IN SECURING WEB SERVICES

---

To build secure Web services, service providers and consumers must implement mechanisms that address the security requirements identified in Section 2 of this document. Ideally, Web service implementers will choose from the Web service specific standards identified in Section 3 and use advanced tools such as those identified in Section 4. However, with budget and resource constraints, it may not be practical to implement a complete, forward-looking information security infrastructure (as described in Section 6) to support Web services. In this section, we describe the basic mechanisms that must be in place to minimally support secure Web services and meet the JRA security requirements described in Section 2 of this paper.

---

### *Minimum Practice to Meet Mandatory Security Requirements*

Figure 2 on page 18 illustrates a minimum acceptable secure Web services configuration. This figure includes a service provider server<sup>5</sup>, a service consumer server, and a network connecting the two servers. Security mechanisms are provided principally from two sources: HTTPS, and the Web services application itself.

HTTPS uses the session-level protocol, Transport Layer Security, TLS<sup>6</sup>, to provide security mechanisms. In the short term, it can be expedient and less costly to have Web services developers include design mechanisms that provide for the security requirement.

A stronger, long-term solution is to use standardized Web service specific protocols and tools, moving security mechanisms into the application places fewer burdens on the enterprise infrastructure.

The following paragraphs indicate how each of the security requirements is met with this minimal configuration. We start with the mandatory requirements – those that absolutely must be addressed to provide secure Web services.

#### **Service Authentication**

Service authentication provides the Web service consumer assurance that they are talking to the service provider they are expecting and not an imposter. Service authentication can be provided by TLS. This requires that the provider obtain a digital certificate and transmit it to the consumer as a part of the standard TLS message exchange. The certificate may be obtained from a Certificate Authority (CA) that is operated by the service provider's organization or through a commercial, third party CA.

#### **Service Consumer Authentication**

Service consumer authentication assures the Web service provider that the consumer is who she/he claims to be. In the Web service context, when we talk about the consumer, we are generally referring to a computer system and not a person.

TLS can provide for symmetric authentication – in other words, the protocol can be used to authenticate both the provider and consumer. However, this typically

---

<sup>5</sup> The service provider server may consist of multiple servers providing, for example, Web services protocol support and database applications.

<sup>6</sup> TLS is an Internet standard protocol. See <http://www.ietf.org/html.charters/tls-charter.html> for more information.

requires that the consumer provide a recognized digital certificate. To reduce the complexity associated with managing consumer certificates in the minimal practice, authentication can be performed at the application level. The consumer will provide authentication information (such as a user name and password) to the Web service application in agreed upon fields of the Web service message. The Web service application will read these fields and use the authentication facilities on the service provider's enterprise (e.g., through an authentication service such as RADIUS or Microsoft NTLM) to perform the authentication.

### **Service Consumer Authorization**

Consumer authorization provides assurance to the Web service provider that the consumer, once authenticated, has sufficient privileges to access the requested service. In minimal practice, authorization is performed by the Web service application. The Web service application can use, for example, an access control list, or in more sophisticated applications, role based access control, to determine if the consumer has the necessary rights and privileges.

### **Message Confidentiality**

Message confidentiality provides assurance that the information exchange by the consumer and provider is not visible to outside, unauthorized parties. TLS provides confidentiality through symmetric key encryption. TLS establishes a session key between the provider and consumer during the initial message exchange stages of the protocol. The service provider should set TLS configuration parameters so that use of a sufficiently secure encryption algorithm, such as AES (or at a minimum, triple DES), is a requirement for access to the service.

TLS provides confidentiality by encrypting the entire session between the consumer and the provider. TLS cannot provide fine-

grained confidentiality to encrypt specific fields within a message (see Section 2 for a description of this requirement). The only way to achieve this is through the Web services application, or preferably, using an XML standard such as XML Encryption.

### **Service Availability**

There are many threats to service availability. Some of them are derived from the additional exposure that Web service protocols may enable and the associated increased risk of denial of service attacks. A good way to thwart such attacks is through the use of application-aware and XML-aware firewalls. These devices are discussed in more detail in the next section of this document. If application-aware firewalls are not available, the service provider should take extra measures to ensure that the servers that host the Web service applications are protected against attacks that would threaten availability. These may include:

*Server hardening, careful administration, and maintenance*

When a justice organization implements Web services, it is often transforming its use of Web servers from the role of publishing general interest information to one of supporting a mission critical, law enforcement function. As a result, the security management rigor applied to these servers must be increased. Server hardening includes the removal or deactivation of all but critical software programs from the server, frequent application of software patches and updates, and more stringent physical security protection.

*Traditional firewall protection*

While most outward facing Web servers accept requests from all sources, firewall tables should limit access to Web services servers to the IP address of known and trusted service consumers.

#### *Virus protection and intrusion detection*

In a resource-constrained environment, the use of virus scanning and intrusion detection sensors are limited to those servers that support mission critical functions and handle sensitive data. All of the servers involved in implementing Web services (e.g., Web, database, and applications servers) should be considered to be mission critical and configured with virus protection and intrusion detection.

#### **User Credentialing and Authorization**

When we discussed consumer authentication and authorization, we specifically noted that in a Web services context, the consumer is considered to be a computer system and not necessarily a person (i.e., a user). However, in many justice applications, the identity and credentials of the individual that will eventually receive the information provided by a Web service is very important. In the next section of this document, we will describe advanced mechanisms and standards for credentialing and authorizing users. In the minimum practice configuration, the Web service application is responsible for user credentialing and authorization.

#### **Auditing**

Maintaining log information to support auditing is, in general, a good security practice. In justice applications, it is often a requirement of the service provider that records of who accessed what and when be maintained. In the minimum practice, the production of audit logs is primarily the responsibility of the Web service application. However, auditing can be augmented by the standard logs that are produced by Web server system software.

#### **Minimum Practice to Meet Additional Security Requirements**

We categorize three of the security requirements as “additional” requirements. By additional, we mean that the decision to include them is based on the functions provided by the Web service and/or policies of the provider and consumer. The minimum practice to meet these requirements is described in the following paragraphs.

#### **Message Addressing**

Addressing confidentiality is sometimes required to disguise the fact that specific services are being requested by specific consumers. Neither TLS nor the Web services confidentiality can be used to provide for this requirement. The Web service provider will have to configure mechanisms that operate at the network level of the protocol stack, such as IPSEC or Internet Protocol version 6 (IPv6) in tunneling mode, to provide for this security requirement.

#### **Message Integrity**

Message integrity assures both parties, the Web service consumer and provider, that the content of their exchanged messages have not been inappropriately modified in transit. Confidentiality mechanisms can provide lightweight message integrity, so the use of additional message integrity mechanisms is sometimes considered unnecessary. However, TLS can be configured to include hashing and message authenticity code (MAC) algorithms that provide a high level of assurance that exchanged messages retain their integrity.

#### **Message Non-repudiation**

Non-repudiation provides assurance to the service provider and/or consumer that the other party sent a specific message at a specific time. Non-repudiation may not be needed in all applications. There are two common ways of providing message non-

repudiation at a reasonable assurance level: (1) by requiring that the originator digitally sign a message or a document contained in a message, and (2) by engaging a third party, a mutually trusted digital notary, to store information about the subject message or document for the purposes of later proof. Both digital signature and digital notary are advanced security mechanisms. Ideally, if these mechanisms are required, they should be applied using Web service-specific standards. The standard for implementing digital signature in Web services is XML-Signature. This mechanism is described in the next section of this document. For the purposes of the minimum practice configuration, we place the burden of provision of non-repudiation on the Web service application.

### ***Minimal Practice Limitations***

From the previous paragraphs, it appears that we can meet most of the security requirements with HTTPS and the Web service application. Why would we consider a more complex and potentially more costly approach? While minimum practice provides for the basic requirements, there are several significant limitations:

#### **Using HTTPS, security only spans the session level**

We can see the limitations of reliance on HTTPS by looking at the complexity added with an orchestrated Web service. Figure 2 on page 18 presents a very simple Web service application. There is only one service provider involved. When the real power of the Web service technology is exploited, the service may be “orchestrated” and multiple service providers will be involved. This situation is shown in Figure 3 on page 18. Note that in Figure 3, two HTTPS (i.e., TLS) sessions are required: one between the service requestor and provider, and one between the two orchestrated providers. As

information is passed from the consumer to multiple providers, it must be unencrypted and re-encrypted. This is inefficient and presents security vulnerability. As we will see in the next section, when Web service-specific message level confidentiality mechanisms are used, encrypted fields can stay encrypted throughout the entire orchestrated service.

#### **Relying on the application for security is not the best strategy**

Under the minimal security practices, we relied heavily on the Web services application to safeguard the service. With this approach, it is hard to avoid proprietary security mechanisms that are unique to the application. Further, the security of the Web service depends heavily on the security technology expertise of the Web services designer. In our proposed target secure Web services architecture, we will, instead, rely on standards-based security mechanisms that are provided at an enterprise level. All Web services built by the provider, independent of who develops them, will use these enterprise security mechanisms. If Web services are secured through consistent standardized mechanisms, a higher level of assurance can be achieved.

#### **More advanced security requirements are not accommodated**

Many of the “additional” security requirements identified in this section are difficult to accommodate though TLS and Web services application layer security mechanisms alone. For example, in previous paragraphs, we pointed out that non-repudiation and digital signature required mechanisms such as XML Signature. Digital signature is one of the additional requirements that are likely to appear in justice Web services. So, relying exclusively on HTTPS and application-level security will limit the functions that can be provided through Web services.

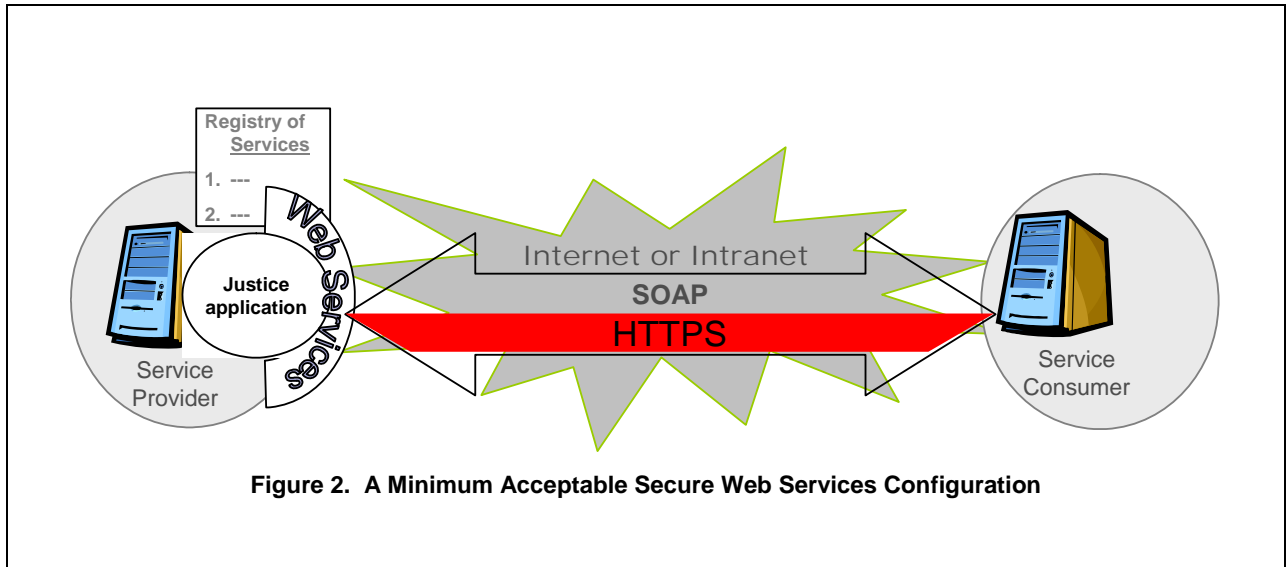


Figure 2. A Minimum Acceptable Secure Web Services Configuration

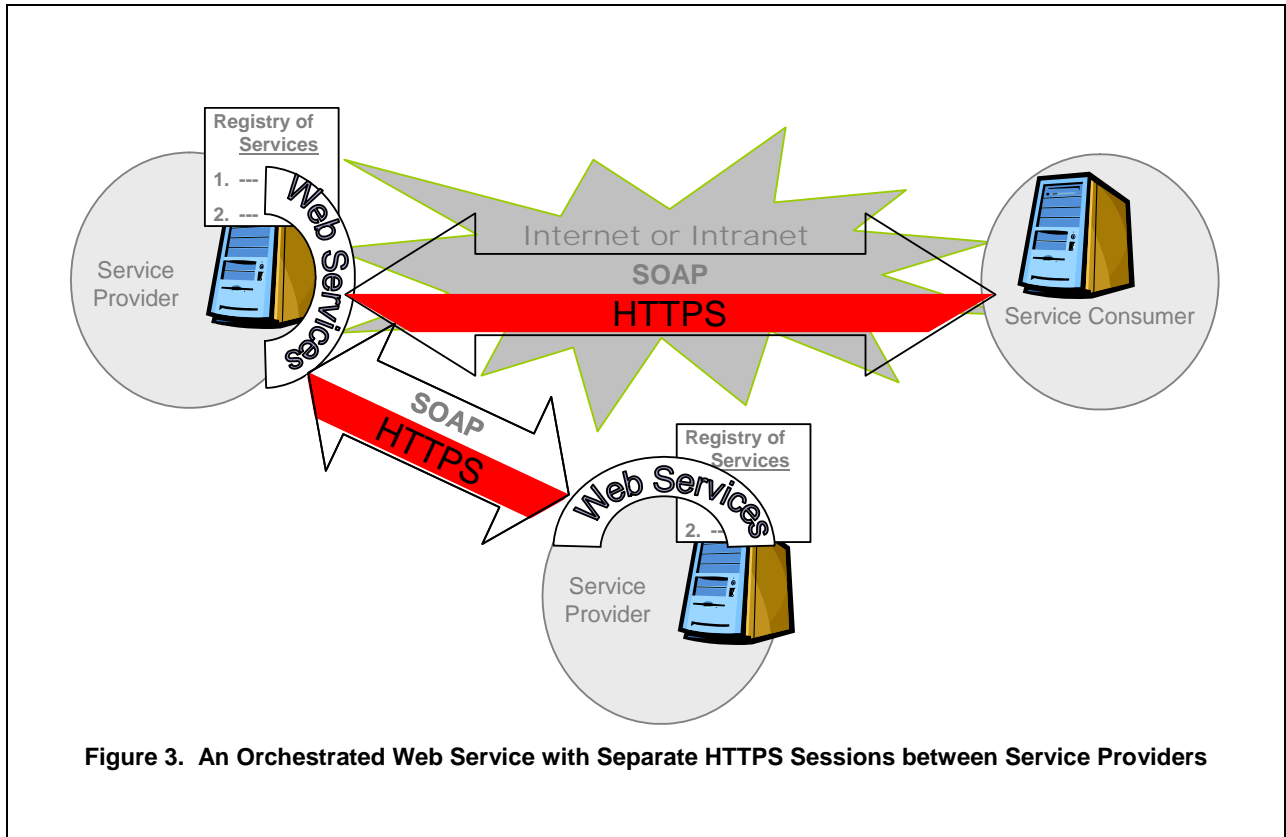


Figure 3. An Orchestrated Web Service with Separate HTTPS Sessions between Service Providers

## SECTION 6. A TARGET SECURITY ARCHITECTURE FOR WEB SERVICES

---

This section discusses a recommended architecture for Web services security in the context of JRA security requirements described earlier in this paper. It also explores recommendations from the *Guide to Secure Web Services*<sup>7</sup> released by the National Institute of Standards and Technology (NIST) and how these, along with other technologies, fit into a target Web services security architecture. These recommendations are based on the consensus of the members of the IJIS Security and Privacy Committee based on a mapping of security requirements for justice information sharing to available and emerging standards for Web services security. However, to date, the specifications recommended in the target architecture have not yet been implemented in this combination. While there are no clear inconsistencies among these specifications, implementers will need to adapt and test the target architecture in the specific environment being protected.

---

### *Shifting the Burden to the Enterprise Infrastructure*

One of the fundamental objectives in implementing an ideal architecture for Web services security involves shifting responsibility for security away from applications and into the infrastructure of the enterprise. The best practices and approaches described in this section allow us to do this so that security is not dependent upon the skills and awareness of application developers. This goes beyond the usual firewalls and intrusion detection mechanisms to incorporate technologies that specifically deal

with the openness and autonomy of Web services.

### *Recommended Standards and Specifications Addressing JRA Security Requirements*

The actual technologies and techniques employed when designing Web services security architecture depend upon the needs of the operational environment and will vary significantly for different industries and applications. The JRA and justice-specific security requirements for JRA service interaction profiles, described previously, are the basic premise for defining best practices and approaches presented in this section.

The following paragraphs indicate how each of the security requirements is met with this recommended configuration.

#### **Service Authentication**

A service consumer can be assured of the identity of a service provider through the use of digital certificates. In minimal configurations, certificates may be exchanged through TLS. However, as a session-level protocol, TLS security is limited to an open connection. Each time a service consumer repeatedly connects and disconnects from a service provider, the identity of the service provider must be revalidated.

A more efficient solution is to use the WS-SecureConversation language to establish a security context, an authenticated state between a service consumer and service provider that may be used by multiple connections between the service consumer and provider. Therefore, the recommended configuration should support service authentication through the WS-SecureConversation specification and the related WS-Security and WS-Trust specifications.

---

<sup>7</sup> *Guide to Secure Web Services, Special Publication 800-95, National Institute of Standards and Technology, August 2007*

### **Service Consumer Authentication**

Similarly, service providers need to be assured of the identity of service consumers. In minimal configurations, service consumer authentication may be performed at the application level. However, this approach forces the service consumer to authenticate to each application independently, which is inefficient. Even worse, each application may implement authentication differently which may force the service consumer to support multiple tokens and authentication mechanisms which limits the scalability and manageability of the information sharing system.

The recommended solution for service consumer authentication is the exchange of Security Assertion Markup Language (SAML) messages and tokens using the WS-Security and WS-I Basic Security Profile specifications. SAML enhances security for service providers by supporting federated identity and a variety of different tokens and authentication mechanisms. SAML simplifies security for service consumers by supporting single sign-on to multiple service providers.

### **Service Consumer Authorization**

Service providers also need to control access to information based on the identity and privileges of both the requesting service consumers and the end users. In minimal configurations, service consumer authorization may be performed by each application. However, this approach makes it very difficult to ensure consistent security, privacy policies, and data access rules across applications.

The recommended solution for service consumer authorization is the use of XACML for describing security and privacy policies. By defining authorizations through a set of XACML rules, access to information can be controlled consistently across applications.

### **Message Confidentiality**

Message confidentiality can be implemented using various types and levels of encryption. In a minimal configuration, TLS provides confidentiality at the session level by encrypting every message in a session with the same symmetric key. In some use cases, however, public key encryption is preferred to symmetric key encryption because it eliminates the need for a shared secret key (and thus a mechanism for secure key exchange) between each combination of sender and recipient. In addition, some messages may be more sensitive than other messages, requiring the use of a different encryption mechanism and/or keys for each message. Furthermore, higher levels of encryption have higher impacts on performance, so applying the highest-level of encryption to every message is generally inadvisable. In addition to the need for message-level confidentiality, some applications may require different levels of encryption at the data element level.

In a recommended configuration, each message should be protected at the appropriate level of encryption, according to the sensitivity of the message, using the XML Encryption and the WS-Security and WS-I Basic Security Profile specifications. This approach grants the implementer a great deal of flexibility, through a broader range of encryption options, and supports better scalability in the number of participants than the use of TLS alone.

### **Service Availability**

In a minimal configuration, availability is ensured through multiple layers of defenses including network firewalls, XML-aware firewalls, hardened servers, virus protection, and intrusion detection/prevention. However, despite these protections, access to services may still be compromised by a failure of any single part of the architecture.

Therefore, in addition to providing multiple defenses against availability, the recommended configuration would implement the WS-ReliableMessaging specification to provide a capability for identifying, managing, and recovering from failures in any part of the architecture, including the network, service providers, and service consumers.

### **User Credentialing and Authorization**

In most applications, the identity and credentials of the individual that will eventually receive the information provided by a Web service is important. This is particularly true in justice information sharing. In the minimal configuration, the Web service application is responsible for user credentialing and authorization. As with service consumer authorization, this approach makes it difficult to ensure consistent security and privacy policies and data access rules across applications.

It is recommended that implementations support the SAML extensions defined by the Global Federated Identity and Privilege Management (GFIPM)<sup>8</sup> project. The GFIPM token includes attributes specific to justice information sharing, including the privileges associated with certain law enforcement roles. In the future, GFIPM will be extended to also support other justice roles beyond law enforcement.

### **Auditing**

In justice applications, the service provider is often required to record certain audit information for each data access including who, what and when. In the minimum practice, the production of audit logs is primarily the responsibility of the Web service application. However, this decentralization limits the visibility into how the information is accessed and used across applications as well as the ability to consistently implement audit rules.

It is recommended that each justice organization define standards for log management to be used by all applications as defined in NIST Special Publication 800-92, *Guide to Computer Security Log Management*. This approach will enable the organization to improve the visibility and consistency of audit rules across the enterprise.

---

<sup>8</sup> GFIPM is the Global Federated Identity and Privilege Management framework supported through the Office of Justice Programs (OJP), Bureau of Justice Assistance (BJA); National Institute of Justice (NIJ); and the U.S. Department of Homeland Security (DHS). The Global Security Working Group (GSWG) provides oversight. The framework provides the justice community and partner organizations with a standards-based approach for implementing federated identity. Additional information may be found on OJP's Website at <http://it.ojp.gov/GFIPM>.

## ***Recommended Practices to Meet Additional Security Requirements***

The recommended practices to meet the additional requirements of Web service, dependent on the requirements of each application, are described in the following paragraphs.

### **Message Addressing**

Some applications may require anonymity in message addressing. As in the minimal configuration, it is recommended that the Web service provider configure mechanisms that operate at the network level of the protocol stack to ensure confidentiality.

### **Message Integrity**

Message integrity ensures that messages have not been modified in transit. In the minimum configuration, TLS provides basic level integrity checks.

In a recommended configuration, message integrity can be protected at multiple levels. The integrity of all messages should be protected using the WS-ReliableMessaging specification. In addition, XML Signatures should be applied as described above to any messages that are particularly sensitive to manipulation.

### **Message Non-repudiation**

Non-repudiation provides assurance to the service provider and/or consumer that the other party sent specific messages at specific times. Not every application will require non-repudiation. In a minimum configuration, the Web service application should provide non-repudiation as needed.

In a recommended configuration, if the application requires non-repudiation, this feature should be provided through the use of XML Signatures applied to the message as

defined by the WS-Security and WS-I Basic Security Profile specifications. The XML signatures provide a mechanism to validate the message provider as well as the time the message was signed at a reasonable assurance level.

## ***Conclusion***

It is recommended that justice organizations implementing Web services and service-oriented architectures implement WS-Security and the related specifications defined in this section and summarized in Table 2 on page 23. These provide a flexible, standards-based, interoperable framework for Web services security. However, new Web services security specifications will continue to emerge, so implementers are encouraged to educate themselves on the latest standards.

SOA / JRA Security Requirement	Recommended Standards / Specifications
Service Authentication	<ul style="list-style-type: none"> <li>• WS-SecureConversation</li> <li>• WS-Security</li> <li>• WS-Trust</li> <li>• X.509</li> </ul>
Service Availability	<ul style="list-style-type: none"> <li>• WS-ReliableMessaging</li> </ul>
Service Consumer Authentication	<ul style="list-style-type: none"> <li>• WS-Security + WS-I Basic Security Profile</li> <li>• SAML</li> <li>• X.509</li> </ul>
Service Consumer Authorization	<ul style="list-style-type: none"> <li>• XACML</li> </ul>
Message Confidentiality	<ul style="list-style-type: none"> <li>• XML Encryption</li> <li>• WS-Security + WS-I Basic Security Profile</li> </ul>
User Credentialing and Authorization	<ul style="list-style-type: none"> <li>• GFIPM</li> </ul>
Auditing	<ul style="list-style-type: none"> <li>• NIST SP 800-92</li> </ul>
Message Addressing	<ul style="list-style-type: none"> <li>• Network-specific</li> </ul>
Message Integrity	<ul style="list-style-type: none"> <li>• WS-ReliableMessaging</li> <li>• XML Signature</li> </ul>
Message Non-repudiation	<ul style="list-style-type: none"> <li>• XML Signature</li> <li>• WS-Security + WS-I Basic Security Profile</li> </ul>

**Table 2. Specifications and Standards addressing JRA Requirements**

## SECTION 7. IMPLEMENTATION EXAMPLES

---

Both of the case studies in this section began at a time when the standards discussed above were in development or still being reviewed. Therefore, both Wisconsin and Ohio have had to look carefully at the emerging standards, many in draft form at the time, and make decisions about what approaches best meet their needs for security.

In both cases, these states at the very least meet the minimal requirements. Each is moving toward what is described as the target architecture.

Part of the challenge in securing justice information sharing is that the local agencies vary widely in technical expertise and resources. It, therefore, becomes important to craft security policies and procedures that:

- ◆ are sufficiently complete for the legacy systems and those locals with less technical capacity, but
- ◆ do not obscure the key points required by those with greater capacity.

---

### *Case Study: Ohio's OLLEISN Project*

OLLEISN (Ohio Local Law Enforcement Information Sharing Network) is a secure, Internet-based system designed to enable Ohio local law enforcement agencies to share law enforcement information with each other.

Information is shared through a centralized repository that is currently accessed by over 700 agencies. The repository contains a full representation of each participating agency's data.

#### **OLLEISN's Web Service Security Requirements**

OLLEISN's Web services are provided to exchange data including both upload services for sending data to the repository and search services for retrieving data from the repository. All agencies are connected to OLLEISN via a hardware-based Virtual Private Network (VPN) that is tunneled over a TLS.

#### *Service Authentication*

The authenticity of the OLLEISN service is trusted, based on the use of a known IP address for the OLLEISN application server.

#### *Service Consumer Authentication*

Web service extensions are utilized to authenticate all consumer requests. These extensions were developed in house to integrate OLLEISN with an LDAP-based security directory maintained by the Ohio Attorney General. By utilizing Web service extensions, the security mechanism is separated from the application ensuring that all SOAP requests are authenticated regardless of the application.

Consumers are authenticated by taking credentials passed as SOAP header extensions and passing them on to the LDAP server for authentication. If the credentials are valid, the request is allowed to pass on to the application.

#### *Service Consumer Authorization*

In addition to authenticating users, the LDAP security server maintained by the Ohio Attorney General also contains user authorization rights. The Web service extensions that authenticate OLLEISN consumers also verify that each consumer is authorized for OLLEISN access. This mechanism is used to enforce OLLEISN's

“give to receive” policy to block requests from non-participating agencies.

#### *Message Confidentiality*

All communications between consumers and OLLEISN’s Web services are secured by the hardware-based VPN that is tunneled over a TLS.

#### *Service Availability*

Access to OLLEISN’s Web services is protected by traditional firewalls and is limited to known and trusted consumer systems. Access is only allowed over established hardware-based VPNs.

The servers that host the OLLEISN Web services are dedicated to this purpose and are not shared by any outward-facing applications. These servers are maintained by the Ohio Attorney General in accordance with their system maintenance policies and procedures.

#### *User Credentialing and Authorization*

All consumer applications that call OLLEISN Web services, are required to pass the end

user’s credentials for the purpose of user credentialing, authorization, and auditing as described in the above Service Consumer Authentication section.

#### *Auditing*

A Web service extension that was developed in house provides a general purpose logging system for audit purposes. The OLLEISN system has the capability to record each use of the system, including the identity of the individual accessing the system, the time of the access to the system, and the information queried.

### **OLLEISN’s Approach to Fulfilling Additional Security Requirements**

#### *Message Non-Repudiation*

None at this time.

#### *Message Integrity*

None at this time.

#### *Message Addressing*

None at this time.

---

### ***Case Study: Wisconsin’s WIJIS Justice Gateway Project***

Wisconsin's WIJIS Gateway is a Web application for the purpose of sharing access to justice community information. The system is intended for use by a wide range of members of the justice community (police, courts, district attorneys, corrections, and so forth). The Gateway accepts data from a wide range of state and local justice data sources, referred to as *Submitters*. The Gateway acts in all cases as an intermediary between the Users and the provided data. The Gateway is a *pointer system*, meaning submitters upload a shallow representation of each incident or arrest record to searchable database. But each submitter then hosts a "record retrieval" service that will provide more detail for a

given record when a Gateway user gets a “hit” and wants to retrieve that detail.

The Gateway’s security architecture falls closer to the “minimally acceptable” solution than the “target architecture” presented above.

Provision of data to the Gateway by Submitters via an SOA solution involves some specific challenges:

- ◆ Every Submitter agency is both a Web service provider and consumer – consuming services to upload pointers and providing a Web service for accepting a “record-retrieval” request, from the Gateway on behalf of a user wanting additional detail about a specific record. Whether acting as a service provider or consumer, each approach has

independent requirements for secure communication and messaging.

### **WIJIS' Implementation of Web Service Security Requirements**

WIJIS ensures that its connection with the information system at each Submitter is secured through:

- ◆ Implementation of technical measures as outlined in the Minimum Acceptable Practice in Securing Web Services (Section 5), details of which are found immediately below.
- ◆ Documentation of procedure.
- ◆ Agreement of responsibility (i.e., memoranda of understanding).

#### *Service Authentication*

WIJIS employs TLS for service authentication when it acts as a Web service *client* of Submitter-hosted services. The Gateway's services, and all data submitters, are required to obtain x.509 digital certificates from the WIJIS certificate authority (CA). [Note: WIJIS currently uses certificates self-signed by a state entity but has plans to migrate to a VeriSign solution] The WIJIS CA permits the Submitters' certificate signing requests (CSRs) to be multi-purposed so that they may be used for both service authentication and service consumer authentication.

#### *Service Consumer Authentication*

WIJIS employs TLS for service consumer authentication when it acts as a Web service *provider* for Submitters' client requests. All services and submitters are required to obtain digital certificates from the WIJIS CA. The WIJIS CA permits the Submitters' certificate signing requests (CSRs) to be multi-purposed so that they may be used for both service authentication and service consumer authentication.

---

Since many SSL/TLS library and utility implementations do not fully support WIJIS-required authentication steps of Subject (a.k.a. the common name or CN) corroboration with the service's hostname and of certificate revocation list (CRL) checking, WIJIS also employs its own developed, open-source software that conducts both *service authentication* and *service consumer authentication*.

---

#### *Service Consumer Authorization*

WIJIS' Web service application performs service consumer authorization in order to assure that the consumer engages in service requests on behalf of only its Submitter jurisdiction and is denied service requests it may make on behalf of any other agency or jurisdiction.

#### *Message Confidentiality*

WIJIS employs TLS, algorithm configuration parameters, and a minimum of 128-bit key length for sufficiently secure encryption.

It is WIJIS' goal to achieve fine-grained confidentiality through use of XACML and a set of tools to improve the ease of XACML authoring, implementation, and maintenance. This toolset, currently under development, is known as Cascading Disclosure Control Language, or CDCL. CDCL will allow business owners to author rules that control disclosure of specific pieces of information at the data node level, not only at the document level. Taken together, XACML and CDCL (or CDCL standing alone) will allow ease-of-use in authoring structured, conditional disclosure rules that are dynamically responsive to different messages, senders, recipients, and contexts; a mechanism for enacting multiple rules at run-time; and a publishing mechanism that allows a high degree of public transparency as to the actual disclosure rules that the system enacts.

#### *Service Availability*

WIJIS employs customized message tracking information to ensure reliable messaging. An eventual migration to WS-ReliableMessaging is likely.

- ◆ **Server hardening, careful administration, and maintenance:** WIJIS employs customary and best-practice host security procedures, software utilities, and restrictions on physical access to its servers.
- ◆ **Traditional firewall protection:** WIJIS employs customary and best-practice network security procedures and devices. However, WIJIS does not limit access to its servers dependent on service consumer IP address due to
  - the relative ease of IP address spoofing
  - many service consumers being configured for dynamic IP address assignment
  - WIJIS does accommodate Submitters that employ IP address limitations for requests of Submitter-hosted services.
- ◆ **Virus protection and intrusion detection:** WIJIS employs customary and best-practice host security procedures, software utilities, and restrictions on physical access to its servers.

#### *User Credentialing and Authorization*

Users are authenticated through Active Directory or other LDAP services. The WIJIS Gateway receives credentialing information from the Active Directory. WIJIS is in the process of using the SAML standard to implement a Federated Identity Management solution, so that end users will authenticate through their “home” identity provider

directories. The home directory will then communicate with the Gateway through the open-source Shibboleth product the user’s credentials. With the publication and maturation of the SAML 2.0 standard, WIJIS anticipates moving to a strong authentication solution using one-time password hardware tokens as a second factor.

#### *Auditing*

WIJIS’ Web service application produces audit logs, and these are augmented by standard system logs.

#### **WIJIS’ Approach to Fulfilling Additional Security Requirements**

##### *Message Non-Repudiation*

None at this time, other than the assurances that come as a byproduct of WIJIS’ solutions for authentication and encryption.

##### *Message Integrity*

None at this time.

##### *Message Addressing*

None at this time.

#### **WIJIS and the WS-\* Standards**

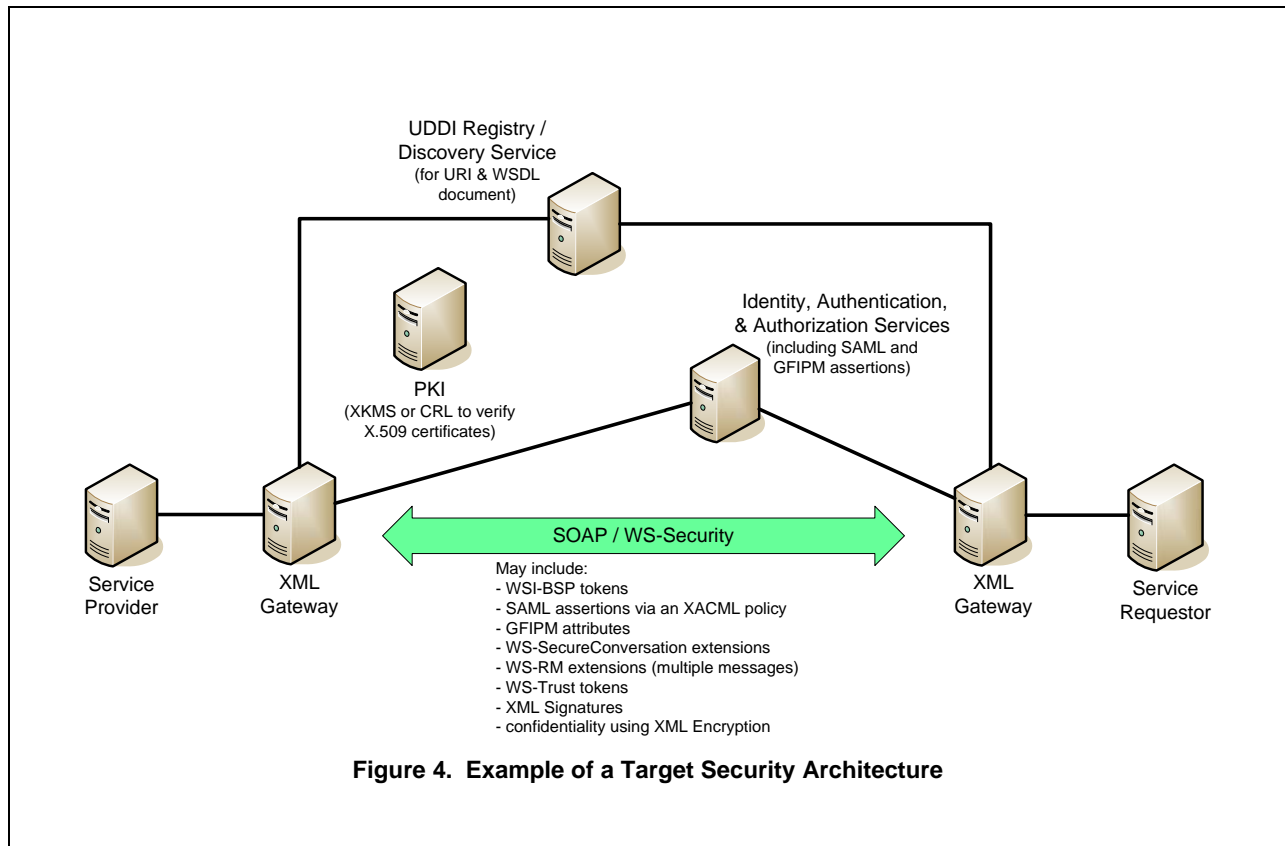
WIJIS developed the Gateway security infrastructure in early 2006. Because of the heterogeneous nature of local law enforcement technology environments, WIJIS desired a uniform means of securing both FTP (for our legacy partners) and Web Services (for the more technologically capable). In addition, WIJIS anticipated using protocols other than SOAP and had concerns about adopting WS-Security or other WS-\* standards at the time. Finally, toolsets and expertise/experience in implementing WS-Security were not available to the state of Wisconsin at the time. WIJIS continues to evaluate emerging SOA security standards.

## SECTION 8. SUMMARY AND CONCLUSIONS

In this paper, we have explored standards and tools for designing and implementing Web services security. Using the Justice Reference Architecture (JRA) security requirements as a premise, we have specified a minimally acceptable approach (using HTTPS, TLS, and application level security) that should be viable for most justice organizations embarking on information sharing projects.

In time, standards relating to Web services security will mature with more widespread adoption, and justice organizations will expect an environment that meets the goals of the target architecture we have specified. Specifically, target architecture shifts the burden of securing Web services to the enterprise infrastructure and addresses each

of the JRA security requirements with more robust and secure standards. Table 3 on page 30 provides a summary contrasting for each JRA security requirement the minimum acceptable practices with the standards and specifications recommended in the target architecture. Figure 4 below shows a diagram (similar to scenarios from the aforementioned NIST *Guide to Secure Web Services*) illustrating components of a target security architecture and where the various security technologies and protocols might be used in a typical operational environment. Many of the standards employed in the target approach result in extensions to the SOAP message in the form of tokens or additional messages.



Though we have confidence in capabilities of the recommended architecture and standards, there are a few notes of caution. First, the landscape of Web services security technologies is always changing. Keeping up with these changes is a constant challenge that requires a dynamic strategy. Also, understanding the benefits and drawbacks of each of the standards and technologies involved is essential. There is no single approach that serves as a universal remedy to securing Web services. Implementers with

technical questions relating to how they might apply the JRA should contact Global and the IJIS Institute, who may be able to make technical assistance resources available. Finally, decisions regarding your security architecture should consider not only the general JRA security requirements but also the specific business needs and risks associated with applications in your environment. We hope this paper provides some guidance for those decisions.

---

SOA / JRA Security Requirement	Minimum Acceptable Practice	Target Architecture (recommended standards / specifications)
Service Authentication	<ul style="list-style-type: none"> <li>Exchange digital certificates through TLS</li> </ul>	<ul style="list-style-type: none"> <li>WS-SecureConversation</li> <li>WS-Security</li> <li>WS-Trust</li> <li>X.509</li> </ul>
Service Availability	<ul style="list-style-type: none"> <li>Network firewalls</li> <li>XML-aware firewalls</li> <li>Hardened servers</li> <li>Virus protection</li> <li>Intrusion detection</li> </ul>	<ul style="list-style-type: none"> <li>WS-ReliableMessaging</li> </ul>
Service Consumer Authentication	<ul style="list-style-type: none"> <li>Performed at application level</li> </ul>	<ul style="list-style-type: none"> <li>WS-Security + WS-I Basic Security Profile</li> <li>SAML</li> <li>X.509</li> </ul>
Service Consumer Authorization	<ul style="list-style-type: none"> <li>Performed by the Web service application</li> </ul>	<ul style="list-style-type: none"> <li>XACML</li> </ul>
Message Addressing	<ul style="list-style-type: none"> <li>Network-specific</li> </ul>	<ul style="list-style-type: none"> <li>Network-specific</li> </ul>
Message Confidentiality (and fine-grained confidentiality)	<ul style="list-style-type: none"> <li>TLS (using AES or DES symmetric key encryption)</li> </ul>	<ul style="list-style-type: none"> <li>XML Encryption</li> <li>WS-Security + WS-I Basic Security Profile</li> </ul>
Message Integrity	<ul style="list-style-type: none"> <li>TLS (using hashing and MAC)</li> </ul>	<ul style="list-style-type: none"> <li>WS-ReliableMessaging</li> <li>XML Signature</li> </ul>
Message Non-repudiation	<ul style="list-style-type: none"> <li>Provided by Web service application as needed</li> </ul>	<ul style="list-style-type: none"> <li>XML Signature</li> <li>WS-Security + WS-I Basic Security Profile</li> </ul>
User credentialing and authorization	<ul style="list-style-type: none"> <li>Responsibility of Web service application</li> </ul>	<ul style="list-style-type: none"> <li>GFIPM</li> </ul>
Auditing	<ul style="list-style-type: none"> <li>Responsibility of Web service application</li> <li>Web server system logs</li> </ul>	<ul style="list-style-type: none"> <li>NIST SP 800-92</li> </ul>

**Table 3. Contrasting Minimum Acceptable and Target Architectures**

## GLOSSARY

---

This glossary provides definitions for many of the acronyms used in this white paper.

Term	Definition
AES	Advanced Encryption Standard
BJA	Bureau of Justice Assistance (U.S. Department of Justice)
CA	Certificate Authority
CDCL	Cascading Disclosure Control Language
CJIS	Criminal Justice Information System
CN	Common Name
CRL	Certificate Revocation List
CSR	Certificate Signing Request
DES	Data Encryption Standard
DHS	U.S. Department of Homeland Security
DOJ	U.S. Department of Justice
GFIPM	Global Federated Identity and Privilege Management
GISWG	Global Infrastructure/Standards Working Group
GSWG	Global Security Working Group
HIPAA	Health Insurance Portability and Accountability Act
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP over SSL
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSEC	IP Security
JRA	Justice Reference Architecture
LDAP	Lightweight Directory Access Protocol
MAC	Messaging Authenticity Code
NIST	National Institute of Standards and Technology
OASIS	Organization for the Advancement of Structured Information Standards
OJP	Office of Justice Programs (U.S. Department of Justice)
OLLEISN	Ohio Local Law Enforcement Information Sharing Network
ORI	ORiginating agency Identifier.
PDP	Policy Decision Point
PKCS	Public Key Cryptography Standard

<b>Term</b>	<b>Definition</b>
PKI	Public Key Infrastructure
SAML	Security Assertion Markup Language
SDK	Software Development Kit
SIPs	Service Interaction Profiles
SOA	Service-oriented Architecture
SOAP	Service-oriented Architecture Protocol
SSL	Secure Socket Layer
STS	Security Token Service
TLS	Transport Layered Security
UDDI	Universal Description, Discovery, and Integration
VPN	Virtual Private Network
W3C	World Wide Web Consortium
WIJIS	Wisconsin Justice Information Sharing
WS-I	Web Services Interoperability Organization
WSM	Web Services Management
XACML	eXtensible Access Control Markup Language
X-KISS	XML Key Information Service Specification
XKMS	XML Key Management Specification
X-KRSS	XML Key Registration Service Specification
XML	eXtensible Markup Language
XMLENC	XML Encoding Library for Java