Ontario

Government of Ontario

Government of Ontario IT Standard (GO-ITS)

Number 24.0

Omnibus Web Services Standard

Version # : 1.3
Status: APPROVED

Prepared for the Information Technology Standards Council (ITSC) under the delegated authority of the Management Board of Cabinet

Last Review Date: 2009-03-12
Next Review Date: 2010-03

Foreword

Government of Ontario Information Technology Standards (GO-ITS) are the official publications on the guidelines, preferred practices, standards and technical reports adopted by the Information Technology Standards Council (ITSC) under delegated authority of the Management Board of Cabinet (MBC). These publications support the responsibilities of the Ministry of Government Services (MGS) for coordinating standardization of Information & Information Technology (I&IT) in the Government of Ontario. Publications that set new or revised standards provide enterprise architecture guidance, policy guidance and administrative information for their implementation. In particular, GO-ITS describe where the application of a standard is mandatory and specify any qualifications governing the implementation of standards.

Table of Contents

1. Introduction

1.1 Background and Purpose

The Omnibus standards; 24.0, 24.1 and 24.2 provide the foundation for the realization of an open and interoperable I&IT environment in the OPS. Each of the Omnibus standards promotes the use of open standards developed by recognized standards development organizations and consortiums. They each provide an integrated set of standards designed to work together to support solution interoperability within the two supported OPS development environments (.NET and Java).

Omnibus 24.2 defines the technical environment for networking and connectivity. Omnibus 24.0 defines a web services interface environment and 24.1 defines the information, content and presentation environment. Together these standards position the OPS to take advantage of Service Oriented Architecture (SOA) frameworks in the future, which hold the greatest potential for lowering integration costs and increasing flexibility across multiple solutions in a shared (consolidated) infrastructure environment.

The Omnibus standards align with, and are designed to be implemented with, other important Government of Ontario IT Standards including **GO-ITS 54 *Application Development Standard*** which specifies application development requirements in the OPS, **GO-ITS 20.1 *Platform Software Standard*** which defines the two OPS standardized IT environments (.NET and Java), and **GO-ITS 23.1 *Government of Ontario Public Web Standard***.

1.2 Target Audience

GO-ITS 24 Omnibus Standards apply to all Government of Ontario technology solutions providers, application development and integration.

1.3 Document Scope

1.3.1 In Scope

This standard focuses on web services. Other standards that impact interoperability, *Really Simple Syndication/Rich Site Summary* (RSS), *Cascading Style Sheets* (CSS), *Atom*, etc., are out of scope and will be addressed in other GO-ITS as appropriate.

Included in the scope of this document are :

- Technical standards from recognized national and international Standards Development Organizations (SDOs) that are relevant when web services are being developed or used
- Business standards and specifications, e.g. WS-BPEL, WS-Coordination and other standards and specifications that extend service concepts
- Web Services standards and specifications – WS-I specifications serve as a foundation for basic, secure and reliable secure Web Services; these standards apply whenever a web services approach is being used

1.3.2 Out of Scope

- Mandating the use of SOA

- Architecture best practices, frameworks, governance, methodologies and principles related to Service Oriented Architecture (SOA)
- Information use, retention, disclosure and disposal policies, regulations or statutes that apply to the OPS, as well as any information asset safeguarding requirements of the OPS (e.g., Statutes, legislation, regulations or OPS-specific directives or standards regarding I&IT security and privacy)
- Monitoring and compliance mechanisms for adherence to this standards document
- Service Level Agreements (SLAs), performance metrics and quality assurance measures
- Vendor products that can form the basis of, or enable Web Services and SOA including procurement policies and strategies involving acquisition of vendor products and services

1.4 Applicability Statements

Government of Ontario IT Standards and Enterprise Products apply (are mandatory) for use by all ministries/clusters and to all former Schedule I and IV provincial government agencies under their present classification (Advisory, Regulatory, Adjudicative, Operational Service, Operational Enterprise, Trust or Crown Foundation) according to the current agency classification system.

Additionally, this applies to any other new or existing agencies designated by Management Board of Cabinet as being subject to such publications, i.e. the GO-ITS publications and enterprise products - and particularly applies to Advisory, Regulatory, and Adjudicative Agencies (see also procurement link, OPS paragraph). Further included is any agency which, under the terms of its Memorandum of Understanding with its responsible Minister, is required to satisfy the mandatory requirements set out in any of the Management Board of Cabinet Directives (*cf*. Operational Service, Operational Enterprise, Trust, or Crown Foundation Agencies).

As new GO-IT standards are approved, they are deemed mandatory on a go-forward basis (Go-forward basis means at the next available project development or procurement opportunity).

When implementing or adopting any Government of Ontario IT standards or IT standards updates, ministries and I&IT Clusters must follow their organization's pre-approved policies and practices for ensuring that adequate change control, change management and risk mitigation mechanisms are in place and employed.

For the purposes of this document, any reference to ministries or the Government includes applicable agencies.

Refer to section 2.0 *Compliance Requirements* for more information.

1.5 Requirements Levels

Within this document, certain wording conventions are followed. There are precise requirements and obligations associated with the following terms:

Must	This word, or the terms "REQUIRED" or "SHALL", means that the statement is an absolute requirement.
Should	This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular circumstances to ignore the recommendation, but the full implications (e.g., business functionality, security, cost) must be understood and carefully weighed before

1.6 Recommended Versioning and/or Change Management

Each of the three Omnibus Standards package together many inter-related standards that have been developed by several different standards development organizations and consortiums. The individual standards development organizations will each be monitored for changes (minor and substantive) to their respective standards and specifications on a regular basis, every six months. As they evolve, the Omnibus standard that references them will also be updated to reflect the changes. When the changes collectively reach a stage where they have a material impact on the implementation of the standard, then the respective Omnibus standard will be brought forward to Information Technology Standards Council (ITSC) and the Architecture Review Board (ARB) for review and approval, as per the approved governance process for all Government of Ontario IT Standards (GO-ITS).

It is important to note that, as with all GO-ITS, the Omnibus standards are meant to align with, and support other related GO-ITS such as the **GO-ITS 20.1 *Platform Software Standard***, therefore if there are updates made to related standards this may also trigger updates to the Omnibus standards.

1.7 Publication Details

All approved Government of Ontario IT Standards (GO-ITS) are published on the ITSC Intranet web site. Please indicate with a checkmark below if this standard is also to be published on the public, GO-ITS Internet Site.

Standard to be published on both the OPS Intranet and the GO-ITS Internet web site (available to the public, vendors etc.)	☑

2. Compliance Requirements

All OPS projects/solutions proceeding through the Corporate Gating Process will be required to complete the Interface Definition information. An interface definition template is found in section 4.0 of this document.

In addition, all OPS projects/solutions proceeding through the Corporate Gating Process are expected to implement a web services approach where it is reasonable and useful to do so.

Sections 5.0 and 6.0 of this document provide the mandatory implementation profiles and web services standards (respectively) that must be adhered to.

Although there is an expectation that projects/solutions will implement a web services approach, decisions regarding where it may not be reasonable and useful to do so (i.e. exemptions) will be reviewed for approval as part of the regular checkpoint review process.

Regarding the use of web services external to the OPS, planners and implementers are required to exploit web services features in an appropriately secured manner.

Also note that regardless of the degree to which web services are implemented, the template documenting the interface descriptions and implementation must be completed.

3. Interoperability

3.1 Introduction to Interoperability

Interoperability refers to the ability of software applications to create, share and effectively use information across a common network, regardless of the development language, application location, or platform.

3.1.1 Web Services

Service Oriented Architecture (SOA) is being evaluated within the OPS as an approach to architecture that would enable enterprise solutions to exchange data and participate in a process without regard for the programming languages or operating systems that underpin each service. These individual services do not have any inherent relationship to any other services, but can be orchestrated through the use of common standards in order to create a composite service.

Web services standards enable service-oriented architecture. According to the World Wide Web Consortium (W3C), a web service is a “software system designed to support interoperable Machine-to-Machine interaction over a network." Web Services (WS) constitute a body of protocols and programmatic interfaces for enabling, defining and implementing application-to-application communication for remote and distributed invocation of application services.

Three standards are fundamental in facilitating this interaction; SOAP, WSDL, and UDDI protocols, which define a self-describing way to discover and call a method in a software application, regardless of location or platform.

1. Simple Object Access Protocol (SOAP) – an XML-based, extensible message envelope format with "bindings" to underlying protocols. The primary protocols are HTTP and HTTPS
2. Web Services Description Language (WSDL) – an XML format that allows service interfaces to be described along with the details of their bindings to specific protocols, typically used to generate server and client code, and for configuration
3. Universal Description Discovery and Integration (UDDI) – a protocol for publishing and discovering metadata about Web services that enables applications to find them, either at design time or runtime

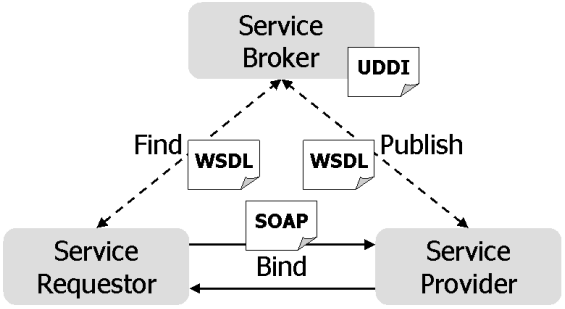

Figure 3.1.1 Web Service view

3.2 Approach to Interoperability

From the perspective of Technology Standards, the OPS’ approach towards achieving solution interoperability is centred on three primary activities:

1. Documenting service interfaces
2. Using web services profiles and their component standards
3. Adopting web services standards not mandated by the profiles

3.2.1 Documenting Service Interfaces

Interfaces are the boundary between two applications or services. Interfaces allow two separate pieces of functionality to be combined in a way that delivers value beyond the individual functions themselves, and the interface specification describes how a particular service is invoked or provided at a specific interface.

It is important to document interfaces in order to ensure that independent services can in fact be connected.

In a service-oriented architecture, Interface Specifications are used to allow independent services (functions) to be executed in a standard way, as opposed to creating highly-specialized services that require custom interfaces for each application that they support. The interface specifications in an SOA model also support interoperability by hiding the implementation of the language or platform upon which a service is based. Services written in C# running on .NET platforms and services written in Java running on Java EE platforms, for example, can both be consumed by a common composite application (or client). Applications running on either platform can also consume services running on the other as Web services, which facilitates reuse. Many COBOL legacy systems can also be wrapped by a managed environment and presented as a software service.

3.2.2 Using Web Services Profiles and their Component Standards

To improve interoperability of Web Services, a number of organizations publish profiles that describe the interrelationships between a set of specifications. A specification typically supports a broad set of requirements and offers a variety of options and approaches, but these options can lead to misinterpretation and result in interoperability challenges. An interoperability profile constrains the options and makes communication easier.

A profile is a set of core specifications (SOAP, WSDL, ...) in a specific version (SOAP 1.1, UDDI 2, ...) with some additional requirements to restrict the use of the core specifications. In some cases, as with the WS-I, the organizations also publish usage scenarios and testing tools that help in deploying profile-compliant Web Services.

The OPS has elected to use the following profiles as the primary mechanism for achieving interoperability. The profiles provide additional guidance above and beyond the use of individual specifications, and therefore are more useful than the individual specifications themselves.

Where a profile specifies the use of a specification, that specification is considered to be a mandatory standard.

The current profiles that have been adopted are:

- Basic Profile (WS-I);
- Basic Security Profile (WS-I)[1];
- Simple SOAP Binding Profile (WS-I); and
- WS-Attachments Profile (IBM, Microsoft)

[1] Additional security protocols may be added to the OPS Web Services protocol stack defined in Figure 3.2.2. A security GO-ITS providing additional guidance for web service implementation should be considered for development as appropriate.

3.2.3 Adopting Web Services Standards

The OPS looks to industry recognized Standards Development Organizations (SDOs) as a primary source for technology standards. A Standards Development Organization (SDO) is an organization whose focus is on developing or coordinating the development of standards. In addition, they usually review, revise, amend and maintain standards.

Examples of SDOs that the OPS references are OASIS and W3C.

Some web services standards are also developed by *Ad Hoc* Vendor Consortiums that collectively agree on a standard, make it available (open and able to be used/licensed by anyone) and submitted it to an SDO for their endorsement. These are viable open standards supported by broad consensus, and they play an important role in achieving interoperability. In this document there are four such standards; Web Services Security Addendum, Web Services Atomic Transaction, Web Services Business Activity Framework and Web Services Meta Exchange. All the other standards included in this document have been developed by recognized SDOs. For a full list of these SDOs and *Ad Hoc* Vendor Consortiums see Appendix A.

Web Services standards define detailed specifications. Web services specifications can be assembled together to provide interoperable protocols for Security, Reliable Messaging and Transactions in loosely coupled systems. The specifications build on top of the core XML and SOAP standards. It is also acknowledged that the definitional range of web services is influenced by various SOA models and continues to expand within the industry, (e.g. see Wikipedia article http://en.wikipedia.org/wiki/Web_services#Service-oriented_architecture).

Where possible, project teams are encouraged to use the WS-I Profiles to guide the development of services. Where additional specifications are required that are not referenced in the profiles, project teams must adopt those standards that have been specified herein.

The following model presents a simplified view of the web services stack and references those standards adopted by the OPS. Please note that the profiles listed along the side of the stack describe how the individual specifications are to be assembled – the profiles are the authoritative guide to the development and deployment of web services. For this reason the individual specifications do not have versions associated with them – the profiles determine the required version.

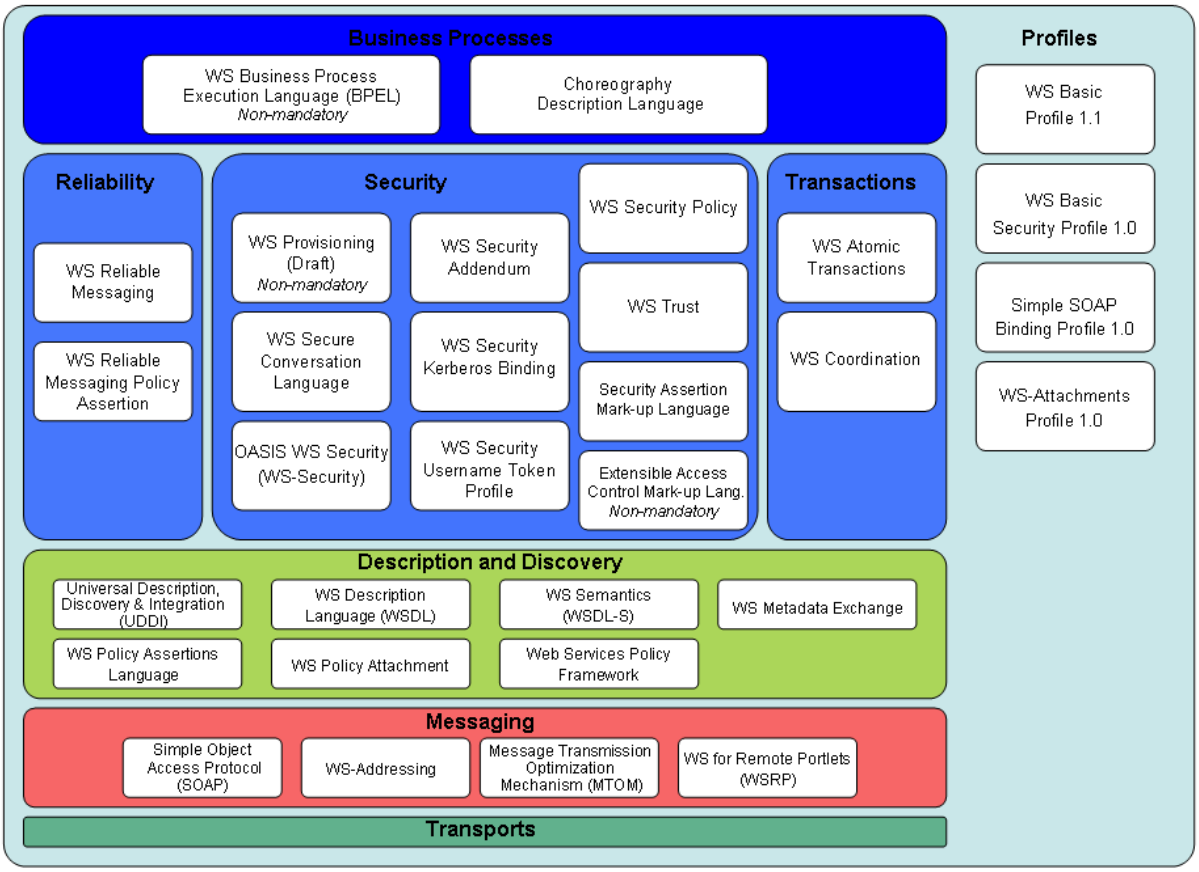

Figure 3.2.2 OPS Web Service Stack and Specifications

4. Mandatory Requirements: Documenting Interface Specifications

When OPS projects are designing automated services for re-use and on-going support within the enterprise, this standard and in particular the associated Service Model Template (SMT), are mandatory.

The SMT provided in Section 4.1 documents the specifications required to develop each interface required by the client solution, including external systems and common applications. In essence, the interface specifications gather together in one place all the information necessary for developing and unit-testing external interfaces related to the client solution.

Interface specifications, and the documentation thereof, are important:

- In large or complex projects where several teams may be developing different parts of a client's application; and
- When the specifications are written by one team and the interfaces are developed by others.

The primary purposes of completing Section 4.1 are to:

- Ensure that the programs comprising a client's application operate together as required;
- Document the specifications for each internal and external interface according to the specified standards; and
- Provide the development teams with the details to develop the interfaces.

The Interface Description document captures the interface information consistent with the Logical Design of a solution, and the interface definition must be completed. The intent of this document is to record the design decisions regarding solution and/or service interoperability, as well as for capturing the interface information consistent with the Physical Deployment of a solution.

The intent here is to capture the implementation details of the interface(s), and to build a record of interface information for future re-use.

The Interface Specifications document is also used to:

- Provide a consistent communications vehicle among the development teams that have responsibility for developing the various programs comprising the client's application;
- Stabilize and publish the system's interfaces;
- Act as documentation for the system; and
- Develop unit test cases.

There are numerous types of interfaces that exist in solution design. The purpose of this standard is to focus on those interfaces that are critical to establishing interoperability amongst solutions. The following are the types of interfaces that are considered to be within the scope of this standard and therefore must be documented using the template in Section 4.1:

- **Application Program:** The interface between an application software entity and an application platform. Several types of specifications may be required and available at this interface. In general, any specification used by a programmer to generate applications code is an API specification;

- **Application-to-Application:** An interface where an application software entity accesses services provided by other application entities (e.g. an LDAP directory service). Services vary based on applications accessed. This assumes a basic data transport paradigm, a set of messages including syntax and semantics, message sequencing, and conventions for handling exceptions that may arise;
- **Communications Service:** The interface in which an application platform accesses external entities that provides data transport services. The service provided is data transport among application platforms. Specifications include protocol states, state transitions, data syntax, and data format that are specified for interoperability among application platforms; and
- **Information Storage:** The interface across which information technology interacts with external storage media. The service provided is persistent storage of data, where the physical storage medium is often removable. Specifications at this interface include physical media and media-independent data format specifications.

Excluded from the mandatory requirements is the documentation of interfaces such as:

- **Human Technology Interface:** The interface between people and information technology, such as web browsers; and
- **Network-to-Network:** An interface where two or more possibly dissimilar communications networks exchange connectivity services, such as between an IP-based network and a PSTN-based phone network.

4.1 Mandatory Interface Description

Please use the following architectural Service Model Template (SMT) to technically describe the service interfaces that your solution requires or provides. You can copy and paste the blank template as many times as is necessary to detail each of your services. As stated in section 4, the interface specifications gather together in one place all the information necessary for developing and unit-testing external interfaces related to the client solution.

The ARB-approved Service Model Template is available at the link below or Appendix C:
http://intra.collaboration.gov.on.ca/mgs/occio/occto/our-services/occto-governance/architecture-core-team/act-presentations/act-2008-presentations/2008-10-09-arb/6-soa-artifacts-for-approval/

5. Mandatory Requirements: Web Services Profiles

5.1 WS-I Basic Profile Version 1.1

Title Web Services Interoperability Basic Profile (WSI-BP)
Version 1.1
Sponsor WS-I

Description
This specification provides interoperability guidance for core Web Services specifications such as SOAP, WSDL, and UDDI. The profile consists of a set of non-proprietary Web services specifications, along with clarifications, refinements, interpretations and amplifications of those specifications which promote interoperability. The profile uses Web Services Description Language (WSDL) to enable the description of services as sets of endpoints operating on messages.

Included Standards

- Messaging
  - SOAP 1.1
  - HTTP 1.1
  - HTTP State Management Mechanism
  - WS-Addressing 1.0
  - WSDL 1.1
- Service Description
  - XML 1.0 (Second Edition)
  - Namespaces in XML 1.0 Second Edition
  - XML Schema Part 1: Structures
  - XML Schema Part 2: Datatypes
  - WSDL 1.1
- Service Publication and Discovery
  - UDDI Version 2.04 Version Specification
  - UDDI Version 2.03 Data Structure Reference
  - UDDI Version 2 XML Schema
- Security
  - HTTP over TLS
  - TLS Protocol Version 1.0
  - SSL Protocol Version 3.0
  - Internet X.509 PKI Certificate and CRL Profile

Scope
All applications using web services.

Business Value
This specification provides resources for application developers to create interoperable web services, and provides tools to ensure that the results are compliant with WS-I guidelines. A specification typically supports a broad set of requirements and offers a variety of options and approaches, but these options can lead to misinterpretation and result in interoperability challenges. An interoperability profile constrains the options and makes communication easier.

Industry Standards http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicprofile
Tools and Support http://www.ws-i.org/Profiles/BasicProfile-1.1.html
Publication Date 2007-03-28

5.2 WS-I Basic Security Profile Version 1.0

Title Web Services Interoperability Basic Security Profile (BSP)
Version 1.0
Sponsor WS-I

Description
The WS-I Basic Security Profile is an interoperability profile that addresses transport security, SOAP messaging security and other security considerations for WS-I's Basic Profile 1.1, Simple SOAP Binding Profile 1.0 and Attachments Profile 1.0. Specifically, the BSP1.0 focuses on the interoperability characteristics of two technologies: HTTP over TLS and Web Services Security: SOAP Message Security.

The BSP1.0 also incorporates Web Services Security: Username Token Profile, Web Services Security: X.509 Certificate Token Profile, Web Services Security: Kerberos Token Profile, Web Services Security: SAML Token Profile and Web Services Security: XRML Token Profile.

Included Standards

- Transport Layer Mechanisms
  - HTTP over TLS
  - TLS Protocol Version 1.0
  - SSL Protocol Version 3.0
- SOAP
  - WS-Security: SOAP Message Security 1.0
  - WS-I Basic Profile Version 1.1
  - Simple SOAP Binding Profile Version 1.0
- Username Token
  - WS-Security: Username Token Profile 1.0
- X.509 Certificate Token
  - WS-Security: X.509 Certificate Token Profile
  - WS-Security: X.509 Token Profile 1.0
  - Internet X.509 PKI Certificate and CRL Profile
- REL Token
  - WS-Security: Rights Expression Language Token Profile 1.0
- Kerberos Token
  - WS-Security: Kerberos Token Profile 1.1
- SAML Token
  - WS-Security: SAML Token Profile 1.0
- Attachment Security
  - Attachments Profile Version 1.0 (AP1.0)
  - WS-Security: SOAP Messages with Attachments (SwA) Profile 1.1

Scope
All applications using web services.

Business Value
This specification provides resources for application developers to create secure interoperable web services, and provides tools to ensure that the results are compliant with WS-I guidelines.

Industry Standards http://www.ws-i.org/deliverables/workinggroup.aspx?wg=basicsecurity
Tools and Support http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html
Publication Date 2007-03-30

5.3 Simple SOAP Binding Profile (SSBP) Version 1.0

Title Simple SOAP Binding Profile (SSBP)
Version 1.0
Sponsor WS-I

Description
This standard defines the WS-I Simple SOAP Binding Profile 1.0, consisting of a set of non-proprietary Web services specifications, along with clarifications and amendments to those specifications which promote interoperability.

Included Standards

Messaging
- Simple Object Access Protocol (SOAP) 1.1
- XML 1.0 (Second Edition)
- Namespaces in XML 1.0
- HTTP 1.1

Description
- WSDL 1.1

Scope
All solutions using web services.

Business Value
This specification provides resources for application developers to create secure interoperable web services, and provides tools to ensure that the results are compliant with WS-I guidelines.

Industry Standards http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0.html
Tools and Support http://www.ws-i.org/Profiles/SimpleSoapBindingProfile-1.0-2004-08-24.html
Publication Date 2004-08-24

5.4 WS-Attachments Profile 1.0

Title Web Services Attachments Profile (WS-Attachments Profile)
Version 1.0
Sponsor WS-I

Description
This standard defines the WS-I Attachments Profile 1.0, consisting of a set of non-proprietary Web services specifications, along with clarifications and amendments to those specifications that are intended to promote interoperability. This profile complements the WS-I Basic Profile 1.1 to add support for interoperable SOAP Messages with Attachments-based Web services.

The WS-I Attachments Profile provides a solution to the limitations that are presented by Web Services Description Language (WSDL) 1.1. Because WSDL 1.1 attachments are not part of the XML schema type space, they can be message parts only. As message parts, the attachments cannot be arrays or properties of Java beans. The profile defines the *wsi:swa-Ref* XML schema type. Use the *wsi:swa-Ref* XML schema type to overcome the limitations of WSDL 1.1 attachments.

Included Standards

Packaging
- SOAP Messages with Attachments

Extensible Markup Language (XML) 1.0 (Second Edition)
Namespaces in XML 1.0
Description
WSDL 1.1

Scope
All solutions based on a web services model.

Business Value
This profile seeks to ensure interoperability of attachment data being exchanged between two parties.

Industry Standards	http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html
Tools and Support	http://www.ws-i.org/Profiles/AttachmentsProfile-1.0-2006-04-20.html
Publication Date	2006-04-20

6. Mandatory Requirements: Web Services

6.1 Business Processes

6.1.1 Choreography Description Language 1.0

Title Choreography Description Language
Version 1.0
Sponsor W3C

Description
The Web Services Choreography Description Language (WS-CDL) is an XML-based language that describes peer-to-peer collaborations of Web Services participants by defining, from a global viewpoint, their common and complementary observable behaviour; where ordered message exchanges result in accomplishing a common business goal.

Scope
All solutions using web services.

Business Value
The main advantage of a global definition approach is that it separates the process being followed by an individual business or system within a "domain of control" from the definition of the sequence in which each business or system exchanges information with others. This means that, as long as the "observable" sequence does not change, the rules and logic followed within the domain of control can change at will.

Industry Standards http://www.w3.org/TR/ws-cdl-10/
Tools and Support http://www.w3.org/TR/2004/WD-ws-cdl-10-20040427/#wscdlxsdschemas/
Publication Date 2004-04-27

6.2 Reliability

6.2.1 Web Services Reliable Messaging (WS-ReliableMessaging) Version 1.1

Title Web Services Reliable Messaging (WS-ReliableMessaging)
Version 1.1
Sponsor OASIS

Description
This specification describes a protocol that allows messages to be transferred reliably between nodes implementing this protocol in the presence of software component, system, or network failures. An Application Source (AS) wishes to reliably send messages to an Application Destination (AD) over an unreliable infrastructure. To accomplish this they make use of a Reliable Messaging Source (RMS) and a Reliable Messaging Destination (RMD). The AS sends a message to the RMS. The RMS uses the WS-ReliableMessaging (WS-RM) protocol to transmit the message to the RMD. The RMD delivers the message to the AD. If the RMS cannot transmit the message to the RMD for some reason, it must raise an exception or otherwise indicate to the AS that the message was not transmitted. The AS and RMS may be implemented within the same process space or they may be

separate components. Similarly, the AD and RMD may exist within the same process space or they may be separate components.

This specification integrates with and complements the WS-Security, WS-Policy, and other Web services specifications. Combined, these allow for a broad range of reliable, secure messaging options.

Scope
All solutions using service-oriented architecture.

Business Value
The protocol is described in this specification in a transport-independent manner allowing it to be implemented using different network technologies.

Industry Standards	http://www.oasis-open.org/specs/index.php#wsrx-rm1.1
Tools and Support	http://docs.oasis-open.org/ws-rx/wsrm/200702/wsrm-1.1-spec-os-01.html
Publication Date	2007-06-14

6.2.2 Web Services Reliable Messaging Policy Assertion (WS-RM Policy) Version 1.1

Title	Web Services Reliable Messaging Policy Assertion (WS-RM Policy)
Version	1,1
Sponsor	OASIS

Description
This specification describes a domain-specific policy assertion for WS-ReliableMessaging that can be specified within a policy alternative as defined in WS-Policy Framework. WS-Policy Framework and WS-Policy Attachment collectively define a framework, model and grammar for expressing the requirements, and general characteristics of entities in an XML Web services-based system. To enable an RM Destination and an RM Source to describe their requirements for a given Sequence, this specification defines a single RM policy assertion that leverages the WS-Policy framework.

The RM policy assertion indicates that the RM Source and RM Destination MUST use WS-ReliableMessaging to ensure reliable delivery of messages. Specifically, the WS-ReliableMessaging protocol determines invariants maintained by the reliable messaging endpoints and the directives used to track and manage the delivery of a Sequence of messages.

Scope
All solutions using service-oriented architecture.

Business Value
By using the XML, SOAP, and WSDL extensibility models, the WS* specifications are designed to be composed with each other to provide a rich Web services environment. This by itself does not provide a negotiation solution for Web services. This is a building block that is used in conjunction with other Web service and application-specific protocols to accommodate a wide variety of policy exchange models.

Industry Standards	http://docs.oasis-open.org/ws-rx/wsrmp/200702/wsrmp-1.1-spec-os-01.html
Tools and Support	Not Applicable
Publication Date	2007-06-14

6.3 Security

6.3.1Web Services Secure Conversation Language (WS-SecureConversation)

Title Web Services Secure Conversation Language (WS-SecureConversation)
Version 1.3
Sponsor OASIS

Description
The Web Services Secure Conversation Language (WS-SecureConversation) is built on top of the WS-Security and WS-Policy models to provide secure communication between services. This specification defines extensions to allow security context establishment and sharing, and session key derivation. This allows contexts to be established and potentially more efficient keys or new key material to be exchanged, thereby increasing the overall performance and security of the subsequent exchanges.

The WS-Security specification focuses on the message authentication model. This approach, while useful in many situations, is subject to several forms of attack (see Security Considerations section of WS-Security specification).

Accordingly, this specification introduces a security context and its usage. The context authentication model authenticates a series of messages thereby addressing these shortcomings, but requires additional communications if authentication happens prior to normal application exchanges.

Scope
All solutions using service-oriented architecture.

Business Value
WS-Security focuses on the message authentication model but not a security context, and thus is subject several forms of security attacks. This specification defines mechanisms for establishing and sharing security contexts, and deriving keys from security contexts, to enable a secure conversation.

Industry Standards http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.3/ws-secureconversation.html
Tools and Support Not Applicable
Publication Date 2005-02-01

6.3.2 Web Services Security: SOAP Message Security (WS-Security) 1.1

Title Web Services Security: SOAP Message Security (WS-Security)
Version 1.1
Sponsor OASIS

Description
This specification describes enhancements to SOAP messaging to provide message integrity and confidentiality. The specified mechanisms can be used to accommodate a wide variety of security models and encryption technologies. This specification is flexible and is designed to be used as the basis for securing Web services within a wide variety of security models including PKI, Kerberos, and SSL. Specifically, this specification provides support for multiple security token formats, multiple trust domains, multiple signature formats, and multiple encryption technologies. The token formats and semantics for using these are defined in the associated profile documents.

This specification also provides a general-purpose mechanism for associating security tokens with message content. No specific type of security token is required, the specification is designed to be extensible (i.e.. support multiple security token formats). For example, a client might provide one format for proof of identity and provide another format for proof that they have a particular business certification.

The specification includes:

- WS-Security Core Specification 1.1
- WS-Security SOAP Message Security 1.1 Errata (only)
- WS-Security SOAP Message Security 1.1 Errata (merged)
- Username Token Profile 1.1
- SAML Token Profile 1.1
- SAML Token Profile 1.1 Errata (only)
- SAML Token Profile 1.1 Errata (merged)
- X.509 Token Profile 1.1
- X.509 Token Profile 1.1 Errata (only)
- X.509 Token Profile 1.1 Errata (merged)
- Kerberos Token Profile 1.1
- Kerberos Token Profile 1.1 Errata (only)
- Kerberos Token Profile 1.1 Errata (merged)
- Rights Expression Language (REL) Token Profile 1.1
- SOAP with Attachments (SWA) Profile 1.1
- SOAP with Attachments (SWA) Profile 1.1 Errata (only)
- SOAP with Attachments (SWA) Profile 1.1 Errata (merged)

Scope
All applications using SOAP.

Business Value
This specification supports increasing integrity and confidentiality on web services messaging.

Industry Standards http://www.oasis-open.org/specs/index.php#wssecpolv1.2
Tools and Support http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf
Publication Date 2006-02-01

6.3.3 Web Services Security Addendum

Title Web Services Security Addendum
Version Not Applicable
Sponsor *Ad hoc* Vendor Consortium (IBM, Microsoft, Verisign)

Description
This document describes clarifications, enhancements, best practices, and errata of the WS-Security specification.

Scope
All solutions using the WS-Security standard.

Business Value
Since the publication of the WS-Security specification, additional reviews and implementation experiences suggest some additions, clarifications, and corrections to the original specification.

Industry Standards http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-secureadd/ws-secureadd.pdf
Tools and Support Not Applicable
Publication Date 2002-08-18

6.3.4 Web Services Security Kerberos Token Profile Version 1.1

Title Web Services Security (WS-Security) Kerberos Token Profile
Version 1.1
Sponsor OASIS

Description
This specification describes the use of Kerberos [Kerb] tokens with respect to the WSS: SOAP Message Security specification WSS. Specifically, this document defines how to encode Kerberos tickets and attach them to SOAP messages. As well, it specifies how to add signatures and encryption to the SOAP message, in accordance with WSS: SOAP Message Security, which uses and references the Kerberos tokens. For interoperability concerns, and for some security concerns, the specification is limited to using the AP-REQ packet (service ticket and authenticator) defined by Kerberos as the Kerberos token. This allows a service to authenticate the ticket and interoperate with existing Kerberos implementations.

Scope
All solutions using Kerberos tokens in a service-oriented architecture.

Business Value
For interoperability concerns, and for some security concerns, the specification is limited to using the AP-REQ packet (service ticket and authenticator) defined by Kerberos as the Kerberos token. This allows a service to authenticate the ticket and interoperate with existing Kerberos implementations.

Industry Standards http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf
Tools and Support Not Applicable
Publication Date 2006-02-01

6.3.5 Web Services Security (WS-Security) Username Token Profile 1.1

Title Web Services Security (WS-Security) Username Token Profile
Version 1.1
Sponsor OASIS

Description
This standard describes how to use the Username Token with the Web Services Security (WSS) specification. More specifically, it describes how a web service consumer can supply a UsernameToken as a means of identifying the requestor by “username”, and optionally using a password (or shared secret, or password equivalent) to authenticate that identity to the web service producer.

Scope
All solutions using service-oriented architecture.

Business Value

The specification improves the security of web services by supporting the use of a username and optional password to identify and authenticate a service requestor.

Industry Standards http://www.oasis-open.org/committees/download.php/16782/wss-v1.1-spec-os-UsernameTokenProfile.pdf
Tools and Support Not Applicable
Publication Date 2006-02-01

6.3.6 Web Services Security Policy (WS-SecurityPolicy) Version 1.2

Title Web Services Security Policy (WS-SecurityPolicy)
Version 1.2
Sponsor OASIS

Description
WS-Policy defines a framework for allowing web services to express their constraints and requirements. Such constraints and requirements are expressed as policy assertions. This document defines a set of security policy assertions for use with the WS-Policy framework with respect to security features provided in WSS: SOAP Message Security, WS-Trust and WS-SecureConversation.

Scope
All solutions using WS-Security.

Business Value
This specification improves the security and integrity of web service solutions.

Industry Standards http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ws-securitypolicy-1.2-spec-os.html
Tools and Support http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.doc
Publication Date 2007-07-01

6.3.7 Web Services Trust (WS-Trust) Version 1.3

Title Web Services Trust (WS-Trust)
Version 1.3
Sponsor OASIS

Description
This specification defines extensions that build on WS-Security to provide a framework for requesting and issuing security tokens, and to broker trust relationships. This specification defines extensions to [WS-Security] that provide:

- Methods for issuing, renewing, and validating security tokens; and
- Ways to establish assess the presence of, and broker trust relationships.

Using these extensions, applications can engage in secure communication designed to work with the general Web services framework, including WSDL service descriptions, UDDI businessServices and bindingTemplates, and SOAP messages.

To achieve this, this specification introduces a number of elements that are used to request security tokens and broker trust relationships.

This specification defines a number of extensions; compliant services are NOT REQUIRED to implement everything defined in this specification. However, if a service implements an aspect of the specification, it MUST comply with the requirements specified (e.g. related "MUST" statements).

Scope
All solutions using WS-Security.

Business Value
This specification provides a protocol-agnostic way to issue, renew, and validate security tokens.

Industry Standards	http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html
Tools and Support	http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.doc
Publication Date	2007-03-19

6.4 SAML V1.1 Specifications (OASIS Web Services)

6.4.1 Security Assertion Markup Language V1.1 (SAML)

6.4.1.1 Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

The Security Assertion Markup Language (SAML) standard defines a framework for exchanging security information between online business partners. It was developed by the Security Services Technical Committee (SSTC) of the standards organization OASIS (the Organization for the Advancement of Structured Information Standards). This document provides a technical description of SAML V1.1.

- sstc-saml-tech-overview-1.1-cd: “Technical Overview of the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Web Services Security TC. 2004-05-11. OASIS Standard. Technical Overview. http://www.oasis-open.org/committees/download.php/6837/sstc-saml-tech-overview-1.1-cd.pdf

6.4.1.2 Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

This specification defines the syntax and semantics for XML-encoded assertions about authentication, attributes and authorization, and for the protocol that conveys this information.

- oasis-sstc-saml-core-1.1: Eve Maler, Prateek Mishra, Robert Philpott, et al. “Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. 2003-09-02. OASIS Standard. Assertions and Protocols. http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

6.4.1.3 Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

This specification defines protocol bindings and profiles for the use of SAML assertions and request-response messages in communications protocols and frameworks.

- oasis-sstc-sami-bindings-1.1: Eve Maler, Prateek Mishra, Robert Pilott, et al. “Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security

Services TC. 2003-09-02. OASIS Standard. Bindings and Profiles. http://www.oasis-open.org/committees/download.php/3405/oasis-sstc-saml-bindings-1.1.pdf

6.4.1.4 Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

This specification describes and analyzes the security and privacy properties of SAML.

- oasis-sstc-saml-sec-consider-1.1: Eve Maler, Rob Philpott, Hal Lockhart, et al. “Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. 2003-09-02. OASIS Standard. Security and Privacy Considerations. http://www.oasis-open.org/committees/download.php/3404/oasis-sstc-saml-sec-consider-1.1.pdf

6.4.1.5 Conformance Program Specification for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

This specification describes the program and technical requirements for SAML conformance.

- oasis-sstc-saml-conform-1.1: Eve Maler, Prateek Mishra, Robert Philpott, et al. “Conformance Program Specification for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. 2003-09-02. OASIS Standard. Conformance Program Specification. http://www.oasis-open.org/committees/download.php/3402/oasis-sstc-saml-conform-1.1.pdf

6.4.1.6 Glossary for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

This specification defines terms used throughout the OASIS Security Assertion Markup Language (SAML) specifications and related documents.

- oasis-sstc-saml-glossary-1.1: Eve Maler, Robert Philpott. “Glossary for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. 2003-09-02. OASIS Standard. Glossary. http://www.oasis-open.org/committees/download.php/3401/oasis-sstc-saml-glossary-1.1.pdf

6.4.1.7 Assertion Schema for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

- oasis-sstc-saml-schema-assertion-1.1: “Assertion Schema for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. OASIS Standard. Assertion Schema (oasis-sstc-saml-schema-assertion-1.1.xsd). http://www.oasis-open.org/committees/download.php/3408/oasis-sstc-saml-schema-assertion-1.1.xsd

6.4.1.8 Protocol Schema for the OASIS Security Assertion Markup Language (SAML) V1.1 – OASIS Standard

- oasis-sstc-saml-schema-protocol-1.1: “Protocol Schema for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. OASIS Standard. Protocol

Schema (oasis-sstc-saml-schema-protocol-1.1.xsd). http://www.oasis-open.org/committees/download.php/3407/oasis-sstc-saml-schema-protocol-1.1.xsd

6.4.1.9 Errata for the OASIS Security Assertion Markup Language (SAML) V1.1

This document lists the reported errata and potential errata against the OASIS SAML 1.1 Committee Specifications and their status.

- sstc-saml-errata-1.1-draft-16: “Errata for the OASIS Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. 2003-09-02. OASIS Standard. Errata. http://www.oasis-open.org/committees/download.php/3325/sstc-saml-errata-1.1-draft-16.pdf

6.4.1.10 Issues List for Security Assertion Markup Language (SAML) V1.1

This document catalogs issues for the Security Assertions Markup Language (SAML) V1.1, developed by the OASIS Security Services Technical Committee. It lists those issues deferred during work on the SAML V1.0 standard and any new issues raised during the SAML V1.1 effort.

- ssct-saml-1.1-issues-draft-02: “Issues List for Security Assertion Markup Language (SAML) V1.1.” OASIS Security Services TC. 2003-10-01. OASIS Standard. Issues. http://www.oasis-open.org/committees/download.php/3690/sstc-saml-1.1-issues-draft-02.pdf

6.4.1.11 Differences between OASIS Security Assertion Markup Language (SAML) V1.1 and V1.0

- ssct-saml-diff-1.1-draft-01: “Differences between OASIS Security Assertion Markup Language (SAML) V1.1 and V1.0.” OASIS Security Services TC. 2003-05-21. OASIS Standard. Differences from V1.0. http://www.oasis-open.org/committees/download.php/3412/sstc-saml-diff-1.1-draft-01.pdf

6.5 Transactions

6.5.1 Web Services Atomic Transaction (WS-Atomic Transaction) Version 1.0

Title Web Services Atomic Transaction (WS-AtomicTransaction)
Version 1.0
Sponsor *Ad hoc* Vendor Consortium (Arjuna Technologies, BEA Systems, Hitachi, IBM, IONA Technologies, Microsoft)

Description
This specification provides the definition of the atomic transaction coordination type that is to be used with the extensible coordination framework described in the WS-Coordination specification.

The WS-Coordination specification defines an extensible framework for defining coordination types. This specification provides the definition of an atomic transaction coordination type used to coordinate activities having an "all or nothing" property. Atomic transactions commonly require a high level of trust between participants and are short in duration. The Atomic Transaction specification defines protocols that enable existing transaction processing systems to wrap their proprietary protocols and interoperate across different hardware and software vendors.

Scope
All solutions using web services.

Business Value
The Atomic Transaction specification defines protocols that enable existing transaction processing systems to wrap their proprietary protocols and interoperate across different hardware and software vendors.

Industry Standards	http://ftpna2.bea.com/pub/downloads/webservices/WS-AtomicTransaction.pdf
Tools and Support	http://ftpna2.bea.com/pub/downloads/webservices/WS-Coordination.pdf
Publication Date	2005-08

6.5.2 Web Services Coordination (WS-Coordination) Version 1.1

Title	Web Services Coordination
Status	Approved
Sponsor	OASIS

Description
This specification describes an extensible framework for providing protocols that coordinate the actions of distributed applications. The coordination protocols that can be defined in this framework can accommodate a wide variety of activities, including protocols for simple short-lived operations and protocols for complex long-lived business activities. For example, WS-AtomicTransaction and WS-BusinessActivity specifications use and build upon this specification.

Scope
All solutions based on service-oriented architecture that aggregate distributed services.

Business Value
This specification enables an application service to create a context needed to propagate an activity to other services and to register for coordination protocols. The framework enables existing transaction processing, workflow, and other systems for coordination to hide their proprietary protocols and to operate in a heterogeneous environment.

Industry Standards	http://docs.oasis-open.org/ws-tx/wstx-wscoor-1.1-spec-cs-01.pdf
Tools and Support	Not Applicable
Publication Date	2006-12-04

6.6 Description and Discovery

6.6.1 Universal Description Discovery and Integration (UDDI) Version 2.0

Title	Universal Description Discovery and Integration (UDDI)
Version	2.0
Sponsor	OASIS

Description
This specification describes a platform-independent, XML-based registry for businesses worldwide to list themselves on the Internet. The focus of Universal Description Discovery & Integration (UDDI) is the definition of a set of services supporting the description and discovery of (1) businesses,

organizations, and other Web services providers, (2) the Web services they make available, and (3) the technical interfaces which may be used to access those services. UDDI is based on a common set of industry standards, including HTTP, XML, XML Schema, and SOAP.

A UDDI business registration consists of three components:

- White Pages — address, contact, and known identifiers;
- Yellow Pages — industrial categorizations based on standard taxonomies; and
- Green Pages — technical information about services exposed by the business.

Scope
All solutions using service-oriented architecture.

Business Value
UDDI provides an interoperable, foundational infrastructure for a Web services-based software environment for both publicly available services and services only exposed internally within an organization.

Industry Standards	http://www.oasis-open.org/committees/uddi-spec/tcspecs.shtml#uddiv2
Tools and Support	http://searchsoa.techtarget.com/originalContent/0,289142,sid26_gci916789,00.html
Publication Date	2002-07-19

6.6.2 Web Services Description Language (WSDL) Version 1.1

Title	Web Services Description Language (WSDL)
Version	1.1
Sponsor	W3C

Description
WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate, however, the only bindings described in this document describe how to use WSDL in conjunction with SOAP 1.1, HTTP GET/POST, and MIME.

A WSDL document uses the following elements in the definition of network services:

- Types– a container for data type definitions using some type system (such as XSD);
- Message– an abstract, typed definition of the data being communicated;
- Operation– an abstract description of an action supported by the service;
- Port Type–an abstract set of operations supported by one or more endpoints;
- Binding– a concrete protocol and data format specification for a particular port type; and
- Port– a single endpoint defined as a combination of a binding and a network address.
- Service– a collection of related endpoints.

Scope
All solutions using service-oriented architecture.

Business Value
WSDL enables one to separate the description of the abstract functionality offered by a service from concrete details of a service description such as "how" and "where" that functionality is offered.

Industry Standards http://www.w3.org/TR/wsdl
Tools and Support http://www.w3.org/TR/wsdl#A4.1
Publication Date 2001-03-15

6.6.3 Web Services Semantics (WSDL-S) Version 1.0

Title Web Services Semantics (WSDL-S)
Version 1.0
Sponsor W3C

Description
The Web Service Semantics document defines a mechanism to associate semantic annotations with Web services that are described using Web Services Description Language (WSDL).

Scope
All solutions using web services.

Business Value
This specification allows Web service developers to annotate their Web services with their choice of ontology language (such as UML or OWL). This is significant because the ability to reuse existing domain models expressed in modeling languages like UML can greatly alleviate the need to separately model semantics.

Industry Standards http://www.w3.org/Submission/WSDL-S/
Tools and Support http://www.w3.org/Submission/2005/SUBM-WSDL-S-20051107/
Publication Date 2005-11-07

6.6.4 Web Services Metadata Exchange (WS-MetadataExchange) Version 1.1

Title Web Services Metadata Exhange (WS-MetadataExchange)
Version 1.1
Sponsor *Ad hoc* Vendor Consortium (BEA, CA, IBM, Microsoft, SAP, webMethods)

Description
Web services use metadata to describe what other endpoints need to know to interact with them. Specifically, WS-Policy describes the capabilities, requirements, and general characteristics of Web services; WSDL describes abstract message operations, concrete network protocols, and endpoint addresses used by Web services; XML Schema describes the structure and contents of XML-based messages received by and sent by Web services.

This specification defines how metadata associated with a Web service endpoint can be represented as WS-Transfer resources, how metadata can be embedded in WS-Addressing endpoint references, and how metadata could be retrieved from a Web service endpoint.

Scope
All solutions using web services.

Business Value
This specification facilitates the retrieval of metadata from complex web services.

Industry Standards http://specs.xmlsoap.org/ws/2004/09/mex/WS-MetadataExchange.pdf
Tools and Support Not Applicable
Publication Date 2006-08-01

6.6.5 Web Services Policy Assertions (WS-PolicyAssertions) Language Version 1.1

Title Web Services Policy Assertions (WS-PolicyAssertions) Language
Version 1.1
Sponsor *Ad hoc* Vendor Consortium (BEA, IBM, Microsoft, SAP)

Description
This specification defines general messaging-related assertions for use with WS-Policy. The goal of WS-PolicyAssertions is to provide an initial set of assertions to address some common needs of Web Services applications.

- Policy – A policy is a set of domain-specific policy assertions; and
- Policy Assertion – A policy assertion represents an individual preference, requirement, capability or other property.

Scope
All solutions using web services.

Business Value
This specification is a building block that is used in conjunction with other Web services and application-specific protocols to accommodate a wide variety of policy exchange models.

Industry Standards http://xml.coverpages.org/ws-policyassertionsV11.pdf
Tools and Support Not Applicable
Publication Date 2003-05-28

6.6.6 Web Services Policy 1.5 – Attachment (WS-PolicyAttachment)

Title Web Services Policy 1.5 - Attachment
Version 1.5
Sponsor W3C

Description
This specification, Web Services Policy 1.5 - Attachment, defines two general-purpose mechanisms for associating policies with the subjects to which they apply; the policies may be defined as part of existing metadata about the subject or the policies may be defined independently and associated through an external binding to the subject.

To enable Web Services Policy to be used with existing Web service technologies, this specification describes the use of these general-purpose mechanisms with WSDL definitions and UDDI.

Scope
All solutions based on service-oriented architecture.

Business Value
Not Applicable

Industry Standards	http://www.w3.org/TR/ws-policy-attach
Tools and Support	http://www.w3.org/TR/2007/REC-ws-policy-attach-20070904
Publication Date	2007-09-04

6.6.7 Web Services Policy Framework (WS-Policy) 1.2

Title	Web Services Policy Framework (WS-Policy)
Version	1.2
Sponsor	W3C

Description
The Web Services Policy Framework (WS-Policy) provides a general purpose model and corresponding syntax to describe the policies of a Web Service.

WS-Policy defines a policy to be a collection of policy alternatives, where each policy alternative is a collection of policy assertions. Some policy assertions specify traditional requirements and capabilities that will ultimately manifest on the wire (e.g., authentication scheme, transport protocol selection). Other policy assertions have no wire manifestation yet are critical to proper service selection and usage (e.g., privacy policy, QoS characteristics). WS-Policy provides a single policy grammar to allow both kinds of assertions to be reasoned about in a consistent manner.

WS-Policy defines a base set of constructs that can be used and extended by other Web services specifications to describe a broad range of service requirements and capabilities.

Scope
All solutions using web services.

Business Value
WS-Policy provides a flexible and extensible grammar for expressing the capabilities, requirements, and general characteristics of entities in an XML Web services-based system. WS-Policy defines a framework and a model for the expression of these properties as policies.

Industry Standards	http://www.w3.org/Submission/WS-Policy/
Tools and Support	http://www.w3.org/Submission/2006/SUBM-WS-Policy-20060425/
Publication Date	2006-04-25

6.7 Messaging

6.7.1 Simple Object Access Protocol (SOAP) Version 1.1

Title Simple Object Access Protocol (SOAP)
Version 1.1
Sponsor W3C

Description
SOAP provides a simple and lightweight mechanism for exchanging structured and typed information between peers in a decentralized, distributed environment using XML. SOAP does not itself define any application semantics such as a programming model or implementation specific semantics; rather it defines a simple mechanism for expressing application semantics by providing a modular packaging model and encoding mechanisms for encoding data within modules. This allows SOAP to be used in a large variety of systems ranging from messaging systems to RPC.

SOAP consists of three parts:

1. The SOAP envelope construct defines an overall framework for expressing what is in a message; who should deal with it, and whether it is optional or mandatory;
2. The SOAP encoding rules defines a serialization mechanism that can be used to exchange instances of application-defined datatypes; and
3. The SOAP RPC representation defines a convention that can be used to represent remote procedure calls and responses.

In addition to the SOAP envelope, the SOAP encoding rules and the SOAP RPC conventions, this specification defines two protocol bindings that describe how a SOAP message can be carried in HTTP messages either with or without the HTTP Extension Framework.

Scope
All solutions based on service-oriented architecture.

Business Value
The framework has been designed to be independent of any particular programming model and other implementation specific semantics.

Industry Standards http://www.w3.org/TR/SOAP
Tools and Support http://www.w3.org/TR/2000/NOTE-SOAP-20000508
Publication Date 2000-05-28

6.7.2 Web Services Addressing (WS-Addressing)

Title Web Services Addressing (WS-Addressing)
Version Not Applicable
Sponsor W3C

Description
WS-Addressing provides transport-neutral mechanisms to address Web services and messages. Specifically, this specification defines XML elements to identify Web service endpoints and to secure end-to-end endpoint identification in messages. This specification enables messaging systems to support message transmission through networks that include processing nodes such as endpoint managers, firewalls, and gateways in a transport-neutral manner.

Scope
All solutions based on service-oriented architecture.

Business Value
WS-Addressing provides transport-neutral mechanisms to address Web services and messages.

Industry Standards	http://www.w3.org/Submission/ws-addressing/
Tools and Support	http://www.w3.org/Submission/2004/SUBM-ws-addressing-20040810/
Publication Date	2004-08-10

6.7.3 Web Services Message Transmission Optimization Mechanism (WS-MTOM)

Title	Web Services Message Transmission Optimization Mechanism (WS-MTOM)
Version	Not Applicable
Sponsor	W3C

Description
The first part of this specification describes an abstract feature for optimizing the transmission and/or wire format of a SOAP message by selectively encoding portions of the message, while still presenting an XML Infoset to the SOAP application.

The second part describes an Optimized MIME Multipart/Related Serialization of SOAP Messages implementing the Abstract SOAP Transmission Optimization Feature in a binding independent way. This implementation relies on the XML-binary Optimized Packaging format.

The third part uses this Optimized MIME Multipart/Related Serialization of SOAP Messages for describing an implementation of the Abstract Transmission Optimization Feature for the SOAP 1.2 HTTP binding.

Scope
All solutions based on service-oriented architecture.

Business Value
The use of WS-MTOM can increase the speed and efficiency of web services-based solutions.

Industry Standards	http://www.w3.org/TR/soap12-mtom/
Tools and Support	http://www.w3.org/TR/2005/REC-soap12-mtom-20050125/
Publication Date	2005-01-25

6.7.4 Web Services for Remote Portlets (WSRP)

Title	Web Services for Remote Portlets (WSRP)
Version	1.0
Sponsor	W3C

Description
This specification defines a web-service interface for interacting with presentation-oriented web services. The intent of this specification is to enable an application designer or administrator to pick from a rich choice of compliant remote content and application providers, and integrate them with just a few mouse clicks and no programming effort.

This joint standard layers on top of the existing web services stack, utilizing existing web services standards and will leverage emerging web service standards (such as security) as they become available. The interfaces are defined using the Web Services Description Language (WSDL).

Scope
All applications that use web services to integrate content from remote sources.

Business Value
This specification provides a standardized method for aggregating content from remote sources.

Industry Standards http://www.oasis-open.org/specs/index.php#wsrpv1.0
Tools and Support http://www.oasis-open.org/committees/download.php/3343/oasis-200304-wsrp-specification-1.0.pdf
Publication Date 2003-09-03

7. Non-Mandatory

7.1 Non-Mandatory XACML Specifications (OASIS Web Services)

7.1.1 Extensible Access Control Markup Language (XACML)

7.1.1.1 Extensible Access Control Markup Language (XACML) Version 1.1 – OASIS Standard

This specification defines an XML schema for an extensible access-control policy language.

- cs-xacml-specification-1.1: “Extensible Access Control Markup Language (XACML) Version 1.1.” OASIS Security Services TC. OASIS Standard. Specification Document. http://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf

7.1.1.2 Policy Schema for the Extensible Access Control Markup Language (XACML) Version 1.1 – OASIS Standard

- cs-xacml-schema-policy-01: “Policy Schema for the Extensible Access Control Markup Language (XACML) Version 1.1.” OASIS Security Services TC. OASIS Standard. Policy Schema (cs-xacml-schema-policy-01.xsd). http://www.oasis-open.org/committees/download.php/915/cs-xacml-schema-policy-01.xsd

7.1.1.3 Context Schema for the Extensible Access Control Markup Language (XACML) Version 1.1 – OASIS Standard

- cs-xacml-schema-context-01: “Context Schema for the Extensible Access Control Markup Language (XACML) Version 1.1.” OASIS Security Services TC. OASIS Standard. Context Schema (cs-xacml-schema-context-01.xsd). http://www.oasis-open.org/committees/download.php/919/cs-xacml-schema-context-01.xsd

7.1.1.4 Extensible Access Control Markup Language (XACML) Version 2.0 – OASIS Standard

This specification defines version 2.0 of the extensible access-control markup language.

- access_control-xacml-2.0-core-spec-cd-04: “Extensible Access Control Markup Language (XACML) Version 2.0.” OASIS Security Services TC. 2004-12-06. OASIS Standard. Specification Document. http://docs.oasis-open.org/xacml/access_control-xacml-2_0-core-spec-cd-04.pdf

7.1.1.5 Policy Schema for the Extensible Access Control Markup Language (XACML) Version 2.0 – OASIS Standard

- access_control-xacml-2.0-policy-schema-cd-04: “Policy Schema for the Extensible Access Control Markup Language (XACML) Version 2.0.” OASIS Security Services TC. OASIS

Standard. Policy Schema (access_control-xacml-2.0-policy-schema-cd-04.xsd). http://www.oasis-open.org/committees/download.php/915/cs-xacml-schema-policy-01.xsd

7.1.1.6 Context Schema for the Extensible Access Control Markup Language (XACML) Version 2.0 – OASIS Standard

- access_control-xacml-2.0-context-schema-cd-04: "Context Schema for the Extensible Access Control Markup Language (XACML) Version 2.0." OASIS Security Services TC. OASIS Standard. Context Schema (access_control-xacml-2.0-context-schema-cd-04.xsd). http://docs.oasis-open.org/xacml/access_control-xacml-2.0-context-schema-cd-04.xsd

7.1.1.7 SAML 2.0 Profile of XACML – OASIS Standard

This specification defines a profile for the use of the OASIS Security Assertion Markup Language (SAML) Version 2.0 to carry XACML 2.0 policies, policy queries and responses, authorization decisions, and authorization decision queries and responses. It also describes the use of SAML 2.0 Attribute Assertions with XACML.

- access_control-xacml-2.0-saml_profile-spec-cd-02: "SAML 2.0 Profile of XACML." OASIS Security Services TC. 2004-11-11. OASIS Standard. Specification Document. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml_profile-spec-cd-02.pdf

7.1.1.8 SAML 2.0 Assertion Extension Schema – OASIS Standard

- access_control-xacml-2.0-saml-assertion-schema-cd-01: "SAML 2.0 Assertion Extension Schema." OASIS Security Services TC. OASIS Standard. Assertion Extension Schema (access_control-xacml-2.0-saml-assertion-schema-cd-01.xsd). http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml-assertion-schema-cd-01.xsd

7.1.1.9 SAML 2.0 Protocol Extension Schema – OASIS Standard

- access_control-xacml-2.0-saml-protocol-schema-cd-01: "SAML 2.0 Protocol Extension Schema." OASIS Security Services TC. OASIS Standard. Protocol Extension Schema (access_control-xacml-2.0-saml-protocol-schema-cd-01.xsd). http://docs.oasis-open.org/xacml/access_control-xacml-2.0-saml-protocol-schema-cd-01.xsd

7.1.1.10 XML Digital Signature Profile of XACML – OASIS Standard

This specification profiles use of the W3C XML-Signature Syntax and Processing Standard in providing authentication and integrity protection for XACML schema instances.

- access_control-xacml-2.0-dsig_profile-spec-cd-01: "XML Digital Signature Profile of XACML." OASIS Security Services TC. OASIS Standard. XML Digital Signature Profile. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-dsig_profile-spec-cd-01.pdf

7.1.1.11 Privacy Policy Profile of XACML – OASIS Standard

This working draft describes a profile of XACML for expressing privacy policies.

- access_control-xacml-2.0-privacy_profile-spec-cd-01: “Privacy Policy Profile of XACML.” OASIS Security Services TC. OASIS Standard. Privacy Profile. http://docs.oasis-open.org/xacml/access_control-xacml-2_0-privacy_profile-spec-cd-01.pdf

7.1.1.12 Hierarchical Resource Profile of XACML – OASIS Standard

This document provides a profile for the use XACML with resources that are structured as hierarchies. The profile addresses resources represented as nodes in XML documents or represented in some non-XML way. The profile covers identifying nodes in a hierarchy, requesting access to nodes in a hierarchy, and specifying policies that apply to nodes in a hierarchy.

- access_control-xacml-2.0-hier_profile-spec-cd-01: “Hierarchical Resource Profile of XACML.” OASIS Security Services TC. OASIS Standard. Hierarchical Resource Profile. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-hier_profile-spec-cd-01.pdf

7.1.1.13 Multiple Resource Profile of XACML – OASIS Standard

This document provides a profile for requesting access to more than one resource in a single XACML Request Context, or for requesting a single response to a request for an entire hierarchy.

- access_control-xacml-2.0-mult_profile-spec-cd-01: “Multiple Resource Profile of XACML.” OASIS Security Services TC. OASIS Standard. Multiple Resource Profile. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-mult_profile-spec-cd-01.pdf

7.1.1.14 Core and Hierarchical Role Based Access Control (RBAC) Profile of XACML – OASIS Standard

This specification defines a profile for the use of XACML in expressing policies that use role based access control (RBAC). It extends the XACML Profile for RBAC Version 1.0 to include recommended Attribute for roles, but reduces the scope to address only “core” and “hierarchical” RBAC. This specification has also been updated to apply to XACML 2.0.

- access_control-xacml-2.0-rbac_profile1-spec-cd-01: “Core and Hierarchical Role Based Access Control (RBAC) Profile of XACML.” OASIS Security Services TC. OASIS Standard. Core and Hierarchical Role Based Access Control (RBAC) Profile. http://docs.oasis-open.org/xacml/access_control-xacml-2.0-rbac_profile1-spec-cd-01.pdf

7.3 Non-Mandatory Business Processes Specifications

7.3.1 Web Services Business Process Execution Language (WS-BPEL) Ver. 2

Title Web Services Business Process Execution Language
Version 2.0
Sponsor OASIS

Description
This specification describes a business process modeling language that is executable, serialized in XML. WS-BPEL defines a model and a grammar for describing the behavior of a business process based on interactions between the process and its partners. The interaction with each partner occurs through Web Service interfaces, and the structure of the relationship at the interface level is encapsulated in what is called a partnerLink. WS-BPEL utilizes several XML specifications: WSDL 1.1, XML Schema 1.0, XPath 1.0 and XSLT 1.0. WSDL messages and XML Schema type definitions provide the data model used by WS-BPEL processes. XPath and XSLT provide support for data manipulation.

Scope
All solutions that use service-oriented architecture to enable business process automation.

Business Value
This specification supports the interoperability of applications by providing a framework that makes business processes executable in an application environment,

Industry Standards http://www.oasis-open.org/specs/index.php#wsbpelv2.0
Tools and Support Not Applicable
Publication Date 2007-04-11

7.4 Non-Mandatory Web Services Provisioning Specifications

7.4.1 Web Services Provisioning Specifications (WS-Provisioning – draft)

Title Web Services Provisioning (WS-Provisioning)
Version N/A
Sponsor OASIS

Description
This document describes the APIs and Schemas necessary to facilitate interoperability between provisioning systems in a consistent manner using Web services.

Scope
All solutions using service-oriented architecture.

Business Value
WS-Provisioning describes the APIs and schemas necessary to facilitate interoperability between provisioning systems and to allow software vendors to provide provisioning facilities in a consistent way. The specification addresses many of the problems faced by provisioning vendors in their use of existing protocols, commonly based on directory concepts, and confronts the challenges involved in provisioning Web Services described using WSDL and XML Schema.

Industry Standards	http://www.oasis-open.org/committees/provision/charter.php
Tools and Support	http://download.boulder.ibm.com/ibmdl/pub/software/dw/library/WSP-20031007.zip
Publication Date	2003-10-01

8. Related Standards

8.1 Impacts to Existing Standards

GO-ITS Number	Describe Impact	Recommended Action (alternatively provide a page number where details can be found)
GO-ITS 28 Web Services	GO-ITS 28 (Previous versions 1 & 2): Previous version 1.0 mandated against the use of web services across the GO-Net firewall. Version 2.0 did not mandate against the use of web services external to the OPS, but regarding their use beyond the GO-Net, planners and implementers are required to "exploit Web Services features demonstrably in an SOA-compliant, appropriately secured manner."	Revised Omnibus: All of the standards have been updated. GO-ITS 28 is replaced by the revised Omnibus as it continues to provide context for product and solution deployments. Regarding use of web services external to the OPS, planners and implementers are required to exploit web services features in an appropriately secured manner. All documents referencing GO-ITS 28 should be updated to reference the revised version, namely, GO-ITS 24.0 Omnibus Web Services Standard.

8.2 Impacts to Existing Environment

Impacted Infrastructure (includes Common Components and other applications)	Describe Impact	Recommended Action (alternatively provide a page number where details can be found)
IT Components and Services	Promotes Web Services as the chosen technology for all future Web Services and SOA implementations (not REST, RPC, DCOM, etc.)	Apply this standards document on a net new basis with respect to future Web Services and SOA implementations
Legacy IT Assets	Promotes Web Services as the chosen technology for all future legacy integration work	Apply this standards document on a retroactive basis only if specified by legacy renewal and transformation projects

9. Contact Information

	Administrative Contact	Technical Contact
Full Name:	Doretta Ojeda	Brian Bisaillon
Job Title:	Standards Program Coordinator	Technical Coordinator
Organization:	Ministry of Government Services (MGS)	Ministry of Government Services (MGS)
Division:	Office of the Corporate Chief Technology Officer (OCCTO)	Office of the Corporate Chief Technology Officer (OCCTO)
Branch:	Technology Adoption Branch	Technology Adoption Branch
Office Phone:	416-327-2094	416-212-0940
E-mail Address:	doretta.ojeda@ontario.ca	brian.bisaillon@ontario.ca

10. Acknowledgements

10.1 Editors

Full Name	Cluster, Ministry and/or Area
Brian Bisaillon, Technical Coordinator	OCCTO
Paul Daly, Standards Coordinator	OCCTO

10.2 Contributors

Full Name	Cluster, Ministry and/or Area
Richard Budel	IBM Canada
Brian Bisaillon	OCCTO
Asim Masoodi	OCCTO
Paul Daly	OCCTO

10.3 Consultations

The following groups were consulted:

Corporate Architecture Branch - OCCTO
Technology Adoption Branch - OCCTO
Common Components, Applications and Services (CCAS) - OCCS
OPS .NET Center of Excellence
Application Architecture Domain Working Group (AADWG)
Technology Architecture Domain Working Group (TADWG)
Architecture Core Team (ACT)

The following individuals were consulted:

Tim Dafoe, MGS Corporate Security Branch
Brady Thompson, MGS OCIPO

11. Document History

Created: 2007-03-15

Approved: 2008-01-16
- Approved by the IT Standards Council as an update to previous Omnibus Technical Standard dated January 2005.

Updated: 2008-06-04
- Cover page numbering set to Version 1.1
- Guidance concerning SAML and XACML made explicit in keeping with the original text of retired GO-ITS 28 *Web Services* standard, i.e. SAML V1.1 set as mandatory with SAML V2.0 and XACML both set as non-mandatory (see new section # 7: *Non-Mandatory*).

Updated: 2008-07-23
- Combined the two interface definition templates into one in order to simplify and remove redundant fields.

Updated: 2008-11-03
- Cover page draft number set to Version 1.1, Draft Revision 8.
- Page 11: Added footnote showing where development of a new security GO-ITS providing guidance for web service implementation should be considered.
- Page 12: Inserted Wikipedia link for highlighting the wide range of web service definition.
- Page 15: Replaced the interface definition template with links and references to the ARB-approved Service Model Template (SMT).

Updated: 2008-11-14
- Revised Figure 3.2.2 *OPS Web Service Stack and Specifications*:
  - Changed BPEL to non-mandatory
  - Added - MTOM; XACML
  - Removed - WS-BusinessActivity 1.0; WS-Notification; WS Resource Framework
- Draft revision number set to Rev. 9

Updated: 2008-11-19
- Distribution to ITSC members. Draft set to Version 1.2

Updated: 2009-01-20
- Removed section 7.2 (non-mandatory reference to SAML 2.0). SAML 1.1 is the mandatory version but SAML 2.0 is deployable, as required, through the usual GO-ITS Exemption Process and I&IT Project Gateway Process
- Distribution to ITSC members. Draft set to Version 1.3

Endorsed: 2009-01-21
- ITSC endorsement to proceed to ARB

Approved: 2009-03-12
- Architecture Review Board approval

12. Copyright Information

Appendix A: Glossary

Standard
Standards, in the technology usage of the term, are generally-accepted definitions that describe how a technology component (e.g. hardware, software, protocol, etc) is defined and implemented.

Standards can be of two types:

- *De jure*: a standard that is formally endorsed by a Standards Development Organization (SDO); or
- *De facto*: a standard that has gained wide-spread support but which as not been ratified by any official standards body

Technology standards are adopted by organizations because they help to ensure seamless interoperability amongst various technology services or solutions.

Open Standard
An open standard, in the technology usage of the term, are standards that are publicly available and allow for free usage. In the OPS, a standard is considered to be open when it complies with all these elements:

- Cannot be controlled by any single person or entity with any vested interests;
- Evolved and managed in a transparent process open to all interested parties;
- Platform-independent, vendor-neutral and usable for multiple implementations;
- Openly published (including availability of specifications and supporting material);
- Available royalty free or at minimal cost, with other restrictions offered on reasonable and non-discriminatory terms; and
- Approved through due process by rough consensus among participants.

Standards Development Organization (SDO)
A Standards Development Organization (SDO) is an organization whose focus is on developing or coordinating the development of standards. In addition, they usually review, revise, amend and maintain standards.

This document makes specific reference to a number of SDOs, including the World Wide Web Consortium (W3C), The Organization for the Advancement of Structured Information Standards (OASIS).

Client Solution
A client solution, in the technology usage of the term, is a combination of hardware, software, services, and applications that are coupled in order to satisfy business requirements. In the SOA world, a solution might be a service, that is to say a simple and unique process, or a composite of multiple services that are linked together.

Appendix B: List of Web Services Standards Development Organizations & Vendor Consortiums

This is a list of the SDOs and *Ad Hoc* Vendor Consortiums referenced in this standard:

SDO	**Description**
Organization for the Advancement of Structured Information Standards (OASIS)	Develops specifications based on XML and SGML
Web Services Interoperability Organization (WS-I)	Promotes Web services interoperability across platforms, operating systems and programming languages
World Wide Web Consortium (W3C)	Develops common protocols to enable the evolution and interoperability of the Web

***Ad Hoc* Vendor Consortium**	**Web Services Area**
IBM, Microsoft, Verisign	Web Services Security Addendum
Arjuna Technologies, BEA Systems, Hitachi, IBM, IONA Technologies, Microsoft	Transactions
BEA, CA, IBM, Microsoft, SAP, webMethods	Web Services Metadata Exchange
BEA, IBM, Microsoft, SAP	Web Services Policy Assertions
Akamai, The Globus Alliance, HP, IBM, SAP, Sonic Software, TIBCO	Web Services Notification

Appendix C: Service Model Template

<Project Name>
Service Model Template

Version <n.n>

[Note: The following template is provided only for reference with the Checklist Guidebook. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document.

This document is a basic guide, and may be customized by the project team or by the user. New sections can be included and reason should be provided when existing sections are removed]

Revision History

Date	Version	Description	Author
<yyyy-mm-dd>	<x.x>	<details>	<name>

Table of Contents

Document Guidelines

- *This template must be completed to ensure "Services" are created in a consistent, standard way, more clients or consumers of these services can look up and apply them as they exist or are adapted, according to their business requirements and technology environments.*

- *A Service is a discreet automated business functionality used by a consumer and accessed through well-defined standardized interface. Services can reside remotely and be discoverable over a network.*

- *Characteristics of a Service include:*
    - *Discrete functionality*
    - *Reusable logic*
        - *Composable (can be assembled into composite applications)*
    - *Share a formal interface*
        - *Abstracts or hides underlying logic*
    - *Loosely coupled and stateless*
    - *Discoverable over a network*
    - *Technology-neutral, standards-based protocols*

- *A service may refer to one or more business processes or functions.*

- *In an architectural context, a service supports a process or processes and provides output(s) as requested or invoked by a client, which may be any consumer of the service (e.g., an end-user, an application or a system). A service provider (which may also be an application or a system) accounts for creating and delivering a service to the client or service consumer.*

- *As the boundary of activities expands beyond one business area (e.g., project, program, ministry, etc.) and across the OPS or other enterprises (i.e., inter-jurisdictional partners), services that are created and delivered can be collected in a common container (e.g., a repository or registry) for retrieval and reuse.*

- *Use this template to document interfaces that are language-independent and are written in a way that can be called from several programming languages (Java, .NET). This is a desired feature for a service-style interface which is not bound to a particular process or system. An example of this is Web Service interface (WSDL).*

- *At the logical level, the Service Model Template will capture the high-level design that represents the transformation of the functional service requirements into a service architecture/design. There may be elements in the template that cannot be completed at this point in time.*

- *At a physical level, project will build upon the logical design SMT and illustrate, in much greater detail, how the service is physically implemented using class/object design, component design, and service interface design. At this point, the SMT is expected to be fully completed.*

- *Do not use this template to document:*

- *Interfaces that are language dependent and call a set of code functions, procedures or library, which are part of an internal computer program.*
- *Proprietary data sharing interfaces*
- *Hardware connectivity interface*

Service Model

Service Definition – Technical Attributes

The service is further specified in terms of the following technical attributes:

Service Property	Description
Service Name	*[Provide the service name used in the Interface Definition (example: that is, in the WSDL for a Web Service).]*
Release Number	*[Provide the version number and revision number.]*
Current State of Release	*[Identify the current state of release.* *The service may be in one of the following states:* ▪ *In Development / Being Provisioned* *A service that is being provisioned. It is in development or being supplied to the enterprise by a third-party provider.* ▪ *Awaiting Certification* *A service for which a specification and system-tested software exist, and which is ready to be certified.* ▪ *Published* *A service which is certified and advertised as being available for consumption in production systems – immediately or at some future date. The service is now subject to change control, production operations and monitoring, and possible asset management.* ▪ *Active* *A service which has been deployed into the production environment. Its behaviour and usage is monitored and alerts acted upon.* ▪ *Retired* *A service which:* ▪ *has become obsolete and is no longer called by any consumer* ▪ *has been superseded by newer releases of the service,* ▪ *has been replaced by a different service* ▪ *is supplying functionality that the enterprise no longer requires*

Service Property	Description
	▪ *is no longer supplied by the third-party provider* ▪ *is no longer approved for use in the enterprise]*
Service Grouping	*[Identify the service grouping that this service is part of.* *The service grouping that this service is part of. The service may belong to one of the following categories:* ▪ *Application Service* *Application Services support program area business functions.* ▪ *Enterprise Application Services* *Enterprise Application Services comprise of services that can be used by the enterprise as a whole, in support of business function.]*
Service Layer	*[Identify the type of service.* *The type of service as specified below:* ▪ *Process Service* *The Process Service implements orchestration and can also be classified as a form of business service. It is very much "business-centric", as it resides at the top of the service layer hierarchy and is responsible for composing business services according to the rules specified in the orchestration workflow logic.* ▪ *Application Service* *Business Services compose IT services to execute business logic.* ▪ *Utility Service* *An IT Service that contains reusable application logic. It may be called by Business Services.* ▪ *Infrastructure Service* *An IT Service that is not directly used by the business. Infrastructure services may include directory services, naming services, or communication services.]*
Transport Binding	*[Identify the transport binding for the service such as HTTP/S, JMS, TCP, etc.]*
Protocol	*[Identify the protocol for the service such as SOAP.]*
Key Standards	*[Identify and list the standards (Industry or OPS) this service is in compliance with. E.g. GO-ITS 24]*

Service Interface Definition

Interface Definition specifies the operations provided by the service.

[Operation 1]

Each operation will have the following attributes:

Interface Definition	Description
Operation Name	*[Provide the name of the operation as in WSDL.]*
Operation Description	*[Provide a brief description of the service operation.]*
Input Message Definition	*[Provide the Input message content using element names from WSDL.]*
Output Message Definition	*[Provide the output message content using element names from WSDL.]*
Fault Message Definition	*[Provide the fault message content using element names from WSDL.]*
Properties	*[Describe the properties of the operation in terms of the following.* ▪ *Atomic or long-running* *Atomic transactions have a short duration and they have an "all or nothing" property. Long-running transactions are designed specifically for those business interactions that occur over a long period of time.* ▪ *Updating or read-only* *Updating service operations change the state of business objects whereas read-only service operations are immutable.*
WSDL Location (if available)	*[Provide the location of the Web Service's Web Services Description Language (WSDL) service description file. WSDL describes the public interface to the Web Service.]*

Type of Interface	□ Application Program		□ Communications Service
	□ Application to Application		□ Information Storage
Characteristic	□ Synchronous		□ Asynchronous
Access Type	□ Web Service	□ Generic Adapter	□ Custom API

Visibility of Service	Description
Public	Service consumable by the general public
Inter-enterprise	Service consumable by partners or other OPS ministries
Intra-enterprise	Service consumable within program area (application)
Intra-domain	Service consumable within a business domain (within multiple program area)

Behavioral Model

The Behavioral Model characterizes the actions invoked against the service and the process aspects of interacting with the service. Service behavior is modeled in the Action Model and the Service Interaction Model.

a. Action Model

Action Model Format	When used
Narrative Description	Simple data retrieval operations
Pre- and Post-Conditions	Complex operations
Machine-readable Description (future)	Sophisticated usage of Web services to allow a service consumer to autonomously choose which provider to use

b. Service Interaction Model

The Service Interaction Model characterizes the temporal relationships and temporal properties of actions and events associated with interacting with the service. It may be expressed as an UML interaction or sequence diagram

[Describe the Service Interaction Model for the service.]

Service Data Usage

[Document any significant information of this design for alignment and integration with the information architectures]

[Describe information needs of the service and depict the service data usage in a diagram].

The Service data usage always includes a class that represents the service itself, which identifies all its operations. This class may also include attributes, but typically does not. Instead, it has associations to other types, which allow an arbitrarily complex definition of its attributes. In addition, invariants are used to express rules that cannot be expressed by the properties of the associations and attributes.]

Component Model

[Provide the component model as described above.]

[It is recommended to use graphic images that clearly show the horizontal partitioning of the service and the overall system functionality in terms of components and their interactions.

This section describes the overall structure of the system, the decomposition of the system into layers and subsystems, and any architecturally significant components.

It describes an overview of the component model and its organization in terms of the components in subsystems and layers, as well as the allocation of packages and classes to the implementation subsystems and components.

This subsection names and defines the various layers and their contents, the rules that govern the inclusion to a given layer, and the boundaries between layers. Include a component diagram that shows the relations between layers.

For each layer, include a subsection with the following information:

- *Its name.*
- *An enumeration of the subsystems located in the layer. For each subsystem, give its name, abbreviation or nickname, and a brief description.*
- *A component diagram shows the subsystems and their import dependencies.]*

The component model identifies and specifies the lower-level components that will be responsible for realizing the functionality and maintaining the Quality of Service (QoS) of the exposed services.

This layer of components typically uses container-based technologies such as application servers to implement the components, workload management, high-availability, and load balancing.

[The component model should contain specification of components that implement the Business or IT Services, including:

- *Interface*
- *Business logic*
- *Access to domain objects*
- *Configurable profile if any]*

State Management Strategy

[This section of the Service Model should express:

- *whether the service is stateful;*
- *the state(s) that need to be managed in the service, and*
- *a state management strategy.*

Quality of Service (QoS) Specification

[For each discreet service, describe the quality-of-service requirements by completing the relevant items in the QLM (Quality Level Metrics) document.]

Standards

[Describe the standards to which the service conforms. This may be a reference to other specifications such as WS-I profiles, or to the default services standards adopted by this service.]