Energistics Energy Industry Web Services Interoperability Standards

VERSION 1.0

Web Services Interoperability	This document describes a set of existing published standards, along with any relevant constraints ('narrowing') on those standards, to which internally developed applications can adhere and which can be used as part of the procurement process for new applications/tools to ensure interoperability of interfaces (i.e. Web Services) between organizations.
Version:	**Web Services Interoperability Technical Standards Version 1.0**
Abstract	This document provides a basis for application and system development that will enhance the ability of those systems to interoperate with systems in other enterprises.
Prepared by:	Energistics and the Industry Services Technical Architecture Work Group.
Date published:	December 22, 2008
Document type:	Specifications
Keywords:	SOA, web services, interoperability,
Comments:	Comments and questions may be directed to Energistics at info@energistics.org.

energistics™
The Energy Standards Resource Centre

Document Information	
DOCUMENT VERSION:	1.0
DATE:	November 11, 2008
Technical	Color: R: 210 G:124, B50
Language	US English

This document was produced by
Energistics, the Industry Services SIG
Technical Architecture Work Group.

Amendment History			
Version	**Date**	**Comment**	**By**
0.1	3/27/2008	Original document as provided by BP	Tom Schultz, SAIC
0.5	5/22/2008	Revised by Web Services Interoperability working team; provided to Web Services Interoperability Forum interested party list for comments and feedback	Tom Schultz, SAIC
0.9	10/20/2008	Incorporates feedback through the interested party review period	Tom Schultz, SAIC
0.95	11/11/2008	Converted document format to Energistics specifications document; revised content where appropriate to conform to Energistics conventions	Tom Schultz, SAIC
1.0	11/11/2008	Document for membership and public review	Alan Doniger, Energistics and Tom Schultz, SAIC
1.0	12/22/2008	Final document that incorporates feedback from the membership and public review	Alan Doniger, Energistics and Tom Schultz, SAIC

Table of Contents

Executive Summary

This industry standards document is in fact a collection of references and endorsements of well established underlying standards. The focus of the Web Services Interoperability Standards is to have an agreed upon nominal technology standard suite for interoperability in the energy industry. By agreeing to a suite of foundational technology standards, a significant amount of time and resources can be saved integrating processes between enterprises, divisions and even departments inside of organizations.

There was no desire to create standards at the protocol level. This effort intended to capture the best standards fit for foundational web service interoperability. Therefore, this information is best used to guide enterprise architects, solution developers, and solution and system integrators in the basic standards that should be used to assure ease of operation between other systems.

The purpose of this document is to describe a set of existing published standards, along with any relevant constraints ('narrowing' of the specifications) on those standards, to which internally developed applications can adhere and which can be used as part of the procurement process for newly purchased applications and/or tools to ensure interoperability of interfaces (i.e. Web Services) within and among organizations. When we speak of 'narrowing' the specifications, it usually means that we are excluding a certain part of the referenced specifications from our industry standards. As an example, SOAP allows for both RPC-style and document-style services, but for the purpose of these industry standards we are including only document-style services.

This document and standards in general are also intended to be a unifying technology standards reference for commercial application and software platform developers. By providing a suite of fundamental technology standards, the energy industry will be greatly aided in its ability to electronically connect processes between separate enterprises in an agile manner.

The overall philosophy was not to create new standards, but simply to enumerate a list of related existing standards which, taken as a whole, provide an overall profile of conforming Web Services that can reasonably be expected to work well within the energy industry.

This document provides a basis for application and system development that will enhance the ability of those systems to interoperate with systems in other enterprises. The goal is to have adoption of these stable, industry standards as a foundation of enterprise interoperability in the energy industry. Through industry agreement and coordinated adoption of these standards, real business value can be derived from the agility and speed of the emerging SOA technologies from both vendors and internally developed systems.

1. Introduction

1.1 Scope

The scope of this suite of industry standards is the technical requirements for Web Services interoperability, excluding specific business process and data requirements. For the purpose of this suite of industry standards, a Web Service is defined quite narrowly to mean message-style services utilizing SOAP/XML over HTTP(S).

The technical areas in this document's scope are shown in *Figure 1* below. Each area below the dotted line is considered in scope. The areas above the dotted line, Data Semantics and Vocabularies, Process Orchestration and Quality of Service are out of scope, as is any additional service specific requirements.

Specifically, these standards do not address:

- Issues of specific messaging schemas (such as are addressed in the Energistics PRODML and WITSML Standards) or semantic interoperability. We do expect in the future that these others standards may be referenced by more specific standards as the technical context for defined services.
- Web service discovery (UDDI).
- Web service choreography.

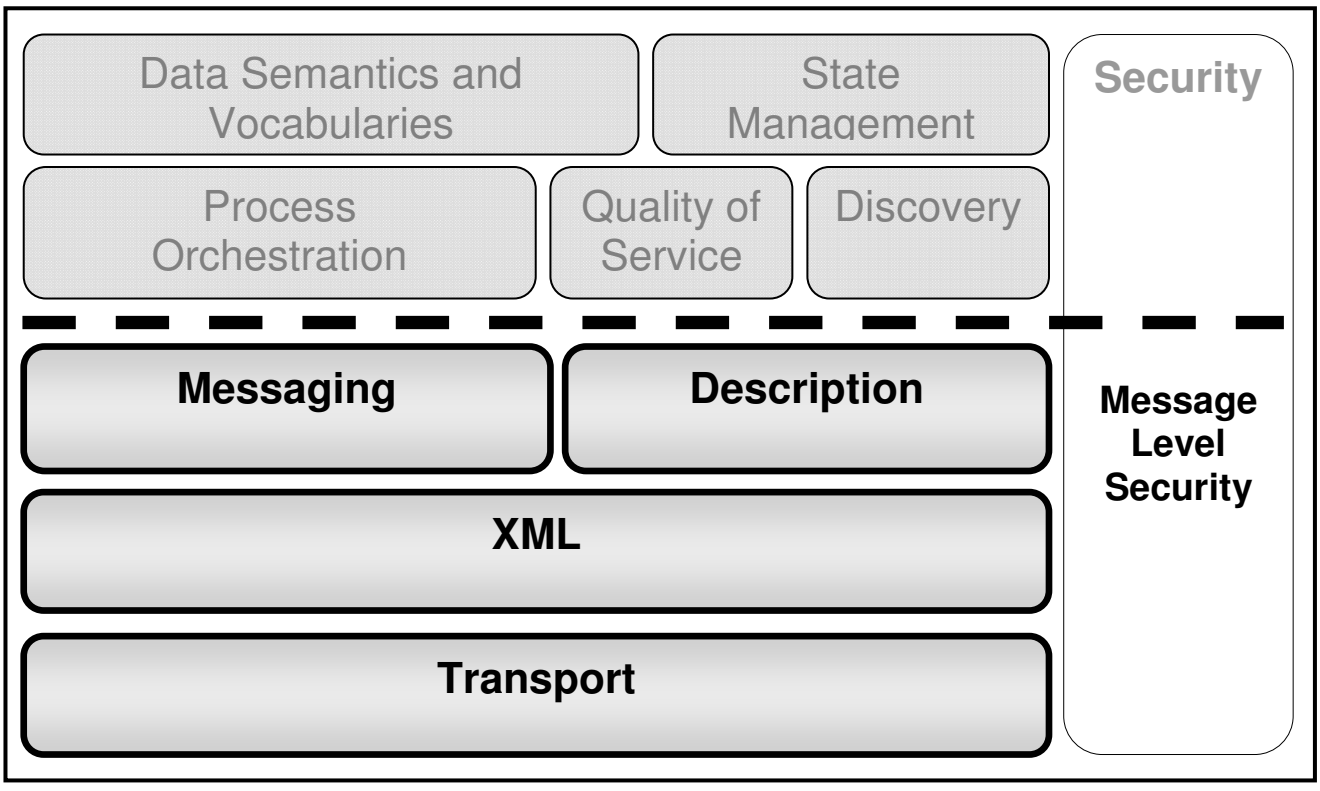

Figure 1 Web Services Interoperability Scope

Industry standards relevant to the document scope are listed below in *Table 1* along with their technical area.

Additional underlying standards may be included in future versions as the industry matures in their use of Web Services.

1.2 Document Structure

This document defines the technical standards that are required to seamlessly integrate two end points (applications, data stores, etc.) and thereby to comply with these standards. The *Technical Requirements for Interoperability* section of the document corresponds to the notions of normative and non-normative that is applied to many standards documents. This section of the document uses the convention of identifying mandatory standards with the word MUST in upper case and advisory standards by the word SHOULD in upper case (see Section 1.5 Terms and Definitions).

Section 1, the introduction, presents the context for the document as a whole. It contains descriptions of the intended audience, intellectual property and copyright statements, terms and definitions, references, and background and history of this document and its content.

Section 2 describes the technical standards that need to be complied with to seamlessly integrate two end points (applications, data stores, etc.). This section of the document corresponds to the notions of normative and non-normative that is applied in many standards documents.

Section 3 provides a point of contact to obtain additional information.

The reader of a printed copy of this document is advised to check that the printed copy is the current version by checking the Energistics web site, http://www.energistics.org.

1.3 Intended Audiences

This document is meant for anyone who is involved in the selection, creation or deployment of applications and infrastructure in the energy industry. The energy industry is defined as oil & gas upstream and downstream, as well as, electric and gas utilities.

Within the energy industry, these standards are intended to be used primarily for *inter*-organizational web service connectivity. By this we mean it is intended to facilitate communication between separate companies, across the internet, using open standards. Many companies should also find these standards useful in defining their own *intra*-organizational Web Services and/or SOA guidelines.

Typical scenarios for employing this set of standards might include:

- Within an organization, architects of a new solution can use it as a reference for developing services that must be consumed by external partners.
- Service providers and operators can use it as a reference for a minimal specification that must be adhered to for implementing a service-based process flow.
- Software vendors can use it as a guide to ensure that their service-enablement plans will meet with broad usability in the energy industry.
- Other industry groups (within Energistics and otherwise) interested in this subject might use it as part of a "reference architecture" for implementing the "services" that they define.

The intended audiences for these standards are illustrated below:

USE KEY

A= PRIMARY DOCUMENT
Key document to read

B=SECONDARY DOCUMENT
Good material to review

C= LESS USEFUL
Review as needed

DOCUMENTATION / ROLE	IT Enterprise Architects	Service Providers	Operators	Software vendors	Other Industry Groups		
Web Services Interoperability Technical Standards	A	B	B	B	B		

1.4 Usage, Intellectual Property Rights, and Copyright

The material described in this document was developed by Energistics and is the intellectual property of Energistics. Energistics develops material for open, public use in order that the material can be of maximum value to the industry as a whole.

Use of the material in this document is governed by the Energistics Intellectual Property Policy document and the Product Licensing Agreement, both of which can be found on the Energistics Web Site, http:/www.energistics.org.

The following statements characterize the intent of the Energistics use policies:

- That all material developed and published by Energistics, and used by other persons or organizations, remain under Energistics' control.
- Others should be able to comment on published Energistics materials.
- Others may copy the material to enable its use.
- Users of Energistics materials acknowledge Energistics as the source.

No one may restrict the usage and/or dissemination of Energistics published materials by attempting to copyright, trademark, license, or to use any other restrictive practice.

1.5 Terminology and Definitions

Throughout this document, terms are used with minimal description. Key terms used in this document are defined as follows:

Term / Acronym	**Definition**
DIME	Direct Internet Message Encapsulation
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
MIME	Multipurpose Internet Mail Extensions
MOM	Message Oriented Middleware
MTOM	Message Transmission Optimization Mechanism
.NET	Microsoft Windows Platform
PRODML	Production Markup Language
RPC	Remote Procedure Call
SAML	Security Assertion Markup Language
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
UDDI	Universal Description, Discovery and Integration
W3C	World Wide Web Consortium
WITSML	Wellsite Information Transfer Standard Markup Language
WS	Web Services
WS-I	Web Services Interoperability Organization
WSDL	Web Services Description Language

XML	Extensible Markup Language

1.6 References

1.6.1 Normative references

These standards are based on a collection of well established standards formed and maintained by mature standards groups. The following is the list of standards groups that were utilized in this work effort.

The World Wide Web Consortium (W3C) http://www.w3.org/

The Internet Engineering Task Force (IETF) http://www.ietf.org/

The Transport Layer Security Working Group of IETF http://www.treese.org/ietf-tls

The Web Service Interoperability Organization (WS-I) http://www.ws-i.org/

1.6.2 WS-I Basic Profile v1.1

The majority of the standards offered in this document build upon those defined in the WS-I Basic Profile v1.1, and Simple SOAP Binding Profile 1.0 owned by the Web Services Interoperability Organization.

A combined claim of conformance to both the Basic Profile 1.1 and the Simple SOAP Binding Profile 1.0 is roughly equivalent to a claim of conformance to the Basic Profile 1.0.

Specific additional guidance is provided (over and above the core WS-I recommendations) by taking into account factors such as current state of popular vendor platforms, future standard evolution, etc. It also provides additional guidance in areas outside the scope of the Basic Profile.

Obviously, the intention of this document is to tighten the WS-I basic profile specification not to loosen it. Therefore, if there is any apparent conflict between this document and the WS-I Basic Profile v1.1, the narrower standard applies.

The WS-I Basic profile promotes interoperability by defining two sets of rules covering:

- Runtime behavior of service consumers and service providers
- Design time Web Service descriptions (WSDL)

In many cases, developer tools not only generate the WSDL but also provide libraries or frameworks that determine behavior at runtime. The main way developers can intervene is by editing the WSDL.

It is important to recognize that registry policies that check WS-I Basic Profile compliance are only checking the WSDL not the runtime behavior.

1.6.3 Additional Included Industry Standards

Due to the nature of this maturing, but quickly changing, technology, not all industry standards are covered. In addition, the current work underway on security frameworks for SOA is expected to provide additional technical requirements that may be included in a future version of these standards.

Table 1 **- In-scope Web Services Industry Standards**

Industry Standard	***Standards Organization***	***Documentation***	***Referenced by WS-I Basic Profile***
Messaging			
SOAP 1.1	W3C	SOAP 1.1 Standard	✓
MTOM	W3C	MTOM Standard	
Description			
WSDL 1.1	W3C	WSDL 1.1 Standard	✓

Security			
TLS 1.0	IETF	TLS 1.0 Standard	✓
SSL 3.0	Transport Layer Security Working Group	SSL 3.0 Standard	✓
XML			
XML 1.0	W3C	XML 1.0 Standard	✓
XML Schema 1.1	W3C	XML Schema 1.1 Standard - part 1 XML Schema 1.1 Standard - part 2 Data types	✓
Namespaces in XML 1.1	W3C	XML 1.1 Namespaces Standard	✓
Transport			
HTTP/1.1	IETF	HTTP 1.1 Standard	✓
Interoperability Information			
Web Services Protocol Workshops			

1.6.4 Notes on Industry Standard Evolution

While it would be preferable to base industry interoperability on broadly agreed IT standards such as SOAP and WSDL, these technical standards are too widely specified to ensure interoperability. In fact, the SOAP v1.1 'standard' is not currently a formal standard. It is a note published by the W3C.

Further, this document recognizes that many of the essential standards addressing areas such as security, reliable messaging, etc. may not be available or fully supported by the commercially available development and runtime tools and/or enterprise applications in all organizations (e.g. current versions of .Net or commercial ERP packages).

1.7 Conventions

This section describes the use of key terms, notations, and abbreviations in this document.

The following terms are used in this specification. These terms extend the definitions in RFC2119 (http://www.ietf.org/rfc/rfc2119.txt).

MAY

With respect to implementations, the word "may" is to be interpreted as an optional feature that is not required in this specification but can be provided. The term "optional" has the same definition as "may".

MUST

In this specification, the word "must" is to be interpreted as a mandatory requirement on the implementation. The terms "required" or "shall" have the same definition as "must".

MUST NOT

In this specification, the words "must not" are to be interpreted as a mandatory prohibition on the implementation. The term "shall not" has the same definition as "must not".

OPTIONAL

See "May".

RECOMMENDED

See "Should".

REQUIRED

See "Must".

SHALL

See "Must".

SHALL NOT

See “Must Not”.

SHOULD

With respect to implementations, the word "should" is to be interpreted as an implementation recommendation, but not a requirement. The term “recommended” has the same definition as "should".

SHOULD NOT

With respect to implementations, the words “should not” are to be interpreted as there may exist a valid reason when the particular implementation recommendation is acceptable.

SUPPORTED

Certain facilities in this specification are optional. If a facility is supported, it behaves as specified by this specification.

UNSPECIFIED

When a value or behavior is unspecified, the specification defines no requirements for a facility on an implementation even when faced with a scenario that uses the facility.

1.8 Motivation

Typically, large organizations have a heterogeneous mix of IT applications and platforms. There are many reasons for this diversity, for example:

- engineering needs for specific systems driving product selection
- multiple applications acquired as a result of merger activity
- major product vendors selecting platforms based on their own technology evaluations
- projects buying best-of-breed solutions rather than specific platform ‘fit’

Given the levels of existing investment in these technologies, there is no reason to expect this environment will change, even in the medium-to-long term.

The result of this diverse environment is that, although it allows for strong alignment of individual solutions to specific business problems, there are significant issues experienced in integrating these ‘islands’ of functionality and allowing data to seamlessly flow around and between organizations.

There is recognizable value and a competitive advantage, if organizations can develop a capability to provide technically consistent interfaces to applications and data sources. This capability can significantly reduce the reinvention of the technical approach in every project and can increase the ability of deployed solutions to exchange information.

Many organizations have begun to embark on projects and/or segment-wide initiatives that use Web Services technologies as a method of integrating these diverse solution environments and exposing common functionality with a similar set of technologies. While this is an encouraging direction, there is a risk that individual implementations of Web Services interfaces by each organization (or worse, each project) will vary and the current diverse environment will be replicated using new technology.

There is an opportunity to define and embrace a rigorous set of standards (specifically based on Web Services) which explicitly define the way that application interfaces should be implemented and which will allow for a higher degree of inter-process, inter-application and, ultimately, inter-segment interoperability than is typically possible today.

1.9 History

The Energy SOA Interoperability collaboration forum was announced in May, 2007 at the annual Energy SOA Roundtable sponsored by SAIC. This event, the 2nd annual Energy SOA Roundtable, was held on May 16 –

17 in Houston, Texas and was attended by several major Oil & Gas (BP, Chevron, ExxonMobil, Hess, Occidental and TransOcean) and Utility (Duke, CenterPoint and SEMPRA) companies.

This roundtable provided participants an opportunity to discuss how traditional architectures intersect with SOA concepts, to share real-world experiences with their peers and to discuss opportunities for collaboration.

In November 2007, SAIC announced an initiative to collaborate on service oriented architecture (SOA) / Web Services interoperability standards for the energy industry. The announcement, made via an Associated Press news release, was picked up by a number of business and industry related publications including The Wall Street Journal online, MarketWatch, Morningstar, MoneyCentral, XML Journal and SOA World Magazine.

The collaboration initiative was open to all companies in the Oil & Gas and Utility industries, including companies providing products and services to companies in the energy industry. A subsequent announcement regarding this initiative was made at a Zapthink SOA Roundtable meeting in Houston.

Representatives from over 40 companies in the energy industry expressed a level of interest in this initiative.

B J Services	Baker Corp	Baker Botts	BP
CAISO	Calpine	Chevron	CollabNet Inc.
ConocoPhillips	Devoteam	Duke	EnCana
EnergyWindow	Entergy	ERCOT	ExxonMobil
Fugro Data Solutions	Halliburton	Hess	Hunt Petroleum
IBM	ICTU	IEEE Computer Society	Information Builders
IntraPoint	Leica Geosystems	Mararthon Oil	NICO
Occidental Petroleum	SAP	Saudi Aramco	SEMPRA
Shell	Sierra Systems	Southern Company	Sun King Productions
Talisman Energy Limited	Transcendent Group	VRcontext	Wipro Technologies
Xtensible Solutions			

The kick-off for this collaboration initiative was held on February 6, 2008 with participation from industry leaders in the Oil & Gas business (BP, Chevron, ExxonMobil, Occidental, Shell and Microsoft). At that time BP shared an internal Web Services Interoperability document for consideration. Subsequently over two dozen companies expressed an interest to continue participation in the initiative. Some companies provided resources for completing the initial set of interoperability standards focused on Web Services.

Participating Companies (Steering and/or Working Team)	**Interested Companies (Receive Communications and Results)**
BP	CAISO
Chevron	Calpine
ExxonMobil	ConocoPhillips
Hess	Duke
Occidental Petroleum	EnCana
Saudi Aramco	Entergy

Shell Oil	ERCOT
Energistics	Halliburton
SAIC	IBM
	Marathon Oil
	Microsoft
	SAP
	SEMPRA
	Southern Company
	Talisman Energy

On March 4, 2008 it was agreed to migrate the Web Services Interoperability initiative into an Energistics Work Group (www.energistics.org), specifically into the Industry Services SIG's Technical Architecture Work Group. The results from the current effort will follow Energistics review processes with the membership and in public resulting in the publication of this document as an Energistics industry standards document.

1.10 Future Plans

There are some areas that were defined to be out-of-scope due to time constraints or issues of maturity, etc. These areas are expected to be addressed in future versions of these standards as the areas mature and become considered mainstream technologies. Examples of future areas include:

- Authentication and Security standards such as WS-I BSP
- Reliable Messaging
- Orchestration

Energistics plans to ask the continuing Industry Services SIG Technical Architecture Work Group to work on augmenting these standards and ensuring a consistent, future foundation for subsequent versions of the Energistics Web Services and Data Exchange Standards Families, including the WITSML Standards for drilling, completions, and well services; the PRODML Standards for production optimization and field reporting; and others.

2. Technical Requirements for Interoperability

2.1 WS-I basic profile 1.1

Standard	In addition to complying with other rules stated in this standard, services MUST comply with the WS-I basic profile version 1.1 available at: WS-I Basic Profile 1.1 and the Simple SOAP Binding Profile Version 1.0 available at: Simple SOAP Binding Profile - Version 1.0 Tools are available from vendors and WS-I to validate WSDL and message schema for WS-I basic profile v1.1 compliance. Developers of Web Services MUST use these tools to ensure WS-I compliance.
Justification	This profile defines rules to aid interoperability between vendor implementations, particularly between .NET and J2EE implementations. In many cases it narrows features permitted in underlying standards. Web Services Basic Profile Version 1.1 from the Web Services Interoperability Organization consists of implementation guidelines and recommendations for implementing Web Services using a core set of specifications that can be used together to develop interoperable Web services.
Advice	WS-I Basic Profile version 1.1.is currently widely adopted by a number of vendors and can be easily implemented in most development environments. A number of additions to the WS-I v1.1 standard are being worked upon such as V 1.2, which builds on Basic Profile 1.1 by incorporating Basic Profile 1.1 errata, requirements from Simple SOAP Binding Profile 1.0, and adding support for WS-Addressing and MTOM. There is also early work ongoing within WS-I on future versions/releases (v2.0 and beyond) which will incorporate and clarify areas such as Reliable Secure messaging.

2.2 Adherence to Standards

Standard	SOAP headers MUST be reserved for headers defined by industry standards such as those adopted by WS-I Basic Security Profile v1.1 Proprietary SOAP headers MUST NOT be used.
Justification	Avoid creation of proprietary mechanisms where standards either exist or are emerging because such mechanisms will eventually be provided in a standard way by vendor products. SOAP headers are generally processed by platform environments (including intermediaries) and are not always accessible to application code.
Advice	SOAP headers are intended to control generic processing of messages by SOAP protocol stacks. Service specific processing should be based on the contents of the message body.

2.3 Attachments

Standard	Where services need to transfer binary documents as attachments to SOAP

	messages, MTOM ('SOAP Message Transmission Optimization Mechanism' W3C Recommendation 25 January 2005) MUST be used as described in WS-I V1.2. Services MUST NOT implement SOAP with Attachments (SwA) as defined by WS-I Attachments Profile Version 1.0.
Justification	Some platforms support attachments over SOAP Web Services using MIME (as recommended by the WS-I.org Attachments Profile v1.0). Carrying attachments as HTTP MIME parts outside the SOAP envelope is efficient and builds on the existing MIME standard. However WS-Security digital signatures cannot be used to bind attachments to the SOAP envelope. There is now broad industry support for MTOM/XOP (SOAP Message Transmission Optimization Mechanism). This is claimed to combine the best aspects of DIME and MIME. On the wire it looks very similar to MIME. But when processed by the SOAP stack it appears as if attachments are embedded in the SOAP envelope, so it is compatible with WS-Security.
Advice	In some cases attachments to Web Services can be avoided by returning a URL from where the consumer can retrieve the file using traditional HTTP or FTP. In this case and especially if the original WS invocation used Transport Security, a separate security mechanism (e.g., HTTPS, SFTP) MUST be used to protect the transfer of information from the HTTP or FTP URL.

2.4 Service Description

Standard	All services MUST be fully described in WSDL documents according to WSDL v1.1 described in a W3C Note, March 2001 'Web Services Description Language' at Web Services Descriptive Language (WSDL) 1.1 W3C Note 15. *WSDL* MUST be available to the authorized consumers. WSDL definitions MUST follow WS-I Basic Profile to ensure interoperability of implementations in different environments. WSDL v2.0 W3C is a W3C Recommendation 26 June 2007; however, it MUST NOT be used. Services SHOULD be defined by explicit, implementation-independent, stateless interfaces and a clearly defined data schema directly associated with the service.
Justification	WSDL is defined in XML and is therefore strictly implementation independent. WSDL v1.1 is currently supported by various tools on different platforms and referenced by WS-I Basic Profile v1.1, whereas WSDL v2.0 is not. *Stateless* – See the discussion of State below. *Explicit* – The WSDL MUST clearly document the functionality of the service to reduce the requirement for a consumer of a service to have prior knowledge of the implementation. *Schema association* – The data format for transfer between interfaces MUST be clearly defined and associated directly with the interface to provide both the 'transport' and 'data' information in a single location. • Reduces effort to use new services by standardizing service description. • Eliminates misunderstandings and ambiguities in traditional written-document based specifications. Enables use of tools which can accelerate development by importing WSDL and building necessary constructs.

Advice	• Many development and test tools generate code from WSDL thereby reducing errors. WSDL interoperability often amounts to tools interoperability (i.e. be able to publish WSDL from one tool and consume that WSDL in another tool). • Formal contract between endpoints place long-term obligations on both the provider and consumer • Interface definitions may require to be supplemented by other documentation that define semantics; such documentation SHOULD be pointed to from the WSDL.

2.5 Namespaces

Standard	Namespaces MUST be used to qualify all payload elements.
Justification	Namespaces prevent collision of names when combining standards and message definitions developed by different parties.
Advice	There are no formal guidelines on the form that namespaces should take. By convention, including the fully-qualified domain of the organization publishing the server (i.e. chevron.com, energistics.org) as a part of the namespace will go a long way to eliminating collisions. Enterprises SHOULD manage namespaces under their control in a manner avoiding ambiguity.

2.6 Transport Protocol

Standard	WS-I compliant services MUST use HTTP (/HTTPS) transport.
Justification	WS-I v1.1 states that messages MUST be sent using either HTTP/1.1 or HTTP/1.0 and that messages SHOULD be sent using HTTP/1.1.
Advice	HTTP Version 1.1: Hypertext Transfer Protocol from IETF based on RFC2616 . Notes on use of the HTTP header In principle, application information required to process a SOAP message SHOULD NOT be placed in HTTP headers. HTTP is a lower level protocol used only for Web Services transport. In particular, assuming the presence of the SOAPAction HTTP header or the value of the SOAPAction HTTP header could result in interoperability problems and therefore MUST NOT be relied upon. This is used inconsistently by SOAP implementations and is often set to an empty string “”. WS-I Basic Profile provides the following rules for WSDL: *R2744 - A HTTP request MESSAGE MUST contain a SOAPAction HTTP header field with a quoted value equal to the value of the soapAction attribute of soapbind:operation, if present in the corresponding WSDL description.* *R2745 - A HTTP request MESSAGE MUST contain a SOAPAction HTTP header field with a quoted empty string value, if in the corresponding WSDL description, the soapAction of soapbind:operation is either not present, or present with an empty string as its value.*

2.7 Message Schemas

Standard	WSDL documents MUST reference or include complete message schemas that provide an explicit template for request and response message bodies in accordance

	with WS-I v1.1.
Justification	WSDL and XML Schema are the standard format for defining an interface. The major development environments natively consume WSDL definitions and create the appropriate code 'stubs' to enable simple consumptions. This reduces adoption effort for both vendors and operators. The WSDL MUST fully define the service interface including the elements required in the request message and the elements that may be expected in the response message. Message schema MAY be embedded in the WSDL or referenced by import or include statements. Schema references MUST be namespace qualified and MAY optionally carry a location attribute as a hint to the recipient parser as to where the schema can be retrieved from. Note that locations are likely to differ between deployment environments.
Advice	XML Schema has replaced the earlier DTD standard. The current version of XML is defined by XML Schema Parts 0, 1 and 2, first published by W3C in 2 May 2001, revised 28 October 2004.

2.8 Messaging Standards

2.8.1 SOAP message style

Standard	Services MUST use a literal wrapped document style.
Justification	The SOAP specification allows messages to be 'document' or 'RPC' style and also 'literal' or 'encoded'. Document literal MAY be further qualified (by convention) as wrapped or unwrapped. RPC style services are allowed in WS-I 1.1 but are not natively supported by many major platforms, including .Net. (i.e. automatic code generation and schema validation is not possible). RPC-based services also break the concept of 'loose coupling' and require a number of out-of-band interactions between the requester and the provider. Use of document style services allows services to be closer to the vision of 'plug-and-play'.
Advice	The only problem with SOAP document literal style is that the WSDL operation is not automatically contained in the message body making dispatch difficult or impossible. The industry has adopted the 'document literal wrapped' convention to solve this problem. The operation is defined by the root element of the request message schema. This can be viewed as a 'wrapper' to the data portion. It is important to note that this is a convention not a standard.

2.8.2 SOAP Operations

Standard	Service operations MUST be specified by the root element ('wrapper element') of the SOAP message body. Web Service implementations MUST use the root element of the SOAP message body to determine the service operation. Web Service implementations MUST NOT rely on the SOAPAction header to indicate the service operation. For secure services, Web Service Descriptions (WSDL) MUST NOT include a

	SOAPAction header or MUST define it as an empty string.
Justification	To identify the required operation in a SOAP message, the operation SHOULD be a 'wrapper element' that appears as the root element of the message body containing the message payload. This is a technique natively supported by Microsoft .NET implementations. The SOAPAction Header MUST NOT be relied on because; • it is not mandatory to specify it in the WSDL • it is an HTTP header and is not forwarded over other transports and • it is not encrypted when the message is sent over HTTPS • it is prohibited by the WS-I Basic Security profile v1.0 rule C2010. WS-I Basic Profile v1.1 section 3.4.3 provides the rules for runtime behavior Although not yet formally adopted by all organizations, the WS-I Basic Security Profile v1.0 outlaws the use of the SOAPAction header for secure Web Services.
Advice	WS-I compliant WSDL allows the SOAPAction Header to be omitted from the WSDL or included. One way to ensure that service providers do not rely on the SOAPAction header is to omit it from the WSDL. There could be an argument that including the SOAPAction header assists debugging by ensuring that SOAP operations appear in HTTP Web logs. If a SOAPAction is included in the WSDL, the service provider SHOULD be tested to ensure that the service works even when consumers omit the SOAPAction at runtime. However, because of the security considerations, the SOAPAction header SHOULD be removed from the WSDL for services in production.

2.8.3 SOAP Headers

Standard	Application data MUST NOT be carried in SOAP headers because headers are generally not accessible by the business application code. All required and optional SOAP headers MUST be defined in the service WSDL binding.
Justification	It SHOULD NOT be necessary for a Web Service intermediary to parse the message body in order to route or authorize a SOAP request. Parsing is a computationally expensive operation. Also message bodies MAY be encrypted with keys not available to intermediaries. Defining SOAP headers in the WSDL ensures that consumer developers are aware of the headers required without having to download and apply information in other documents. In the future, service registries could be able to attach WS-Policy statements to service definitions in a way that UDDI aware development environments can access them. When such support is available, it could make sense to move some header definitions into WS-Policy statements.
Advice	Implementations SHOULD NOT define custom SOAP headers. In most cases there will either be a standard header that does the job or the information belongs in the message body. The example below illustrates correct usage: <wsdl:binding name="StockServiceSoapBinding" type="intf:StockService"> <wsdlsoap:binding style="document" transport= "http://schemas.xmlsoap.org/soap/http"/> <wsdl:operation name="getLastSellPrice"> <wsdlsoap:operation soapAction=""/> <wsdl:input name="getLastSellPriceRequest"> <wsdlsoap:header message="intf:getLastSellPriceRequest" part=

	"request_header" use="literal"/> <wsdlsoap:body parts="parameters" use="literal"/> </wsdl:input> <wsdl:output name="getLastSellPriceResponse"> <wsdlsoap:body use="literal"/> </wsdl:output> </wsdl:operation> </wsdl:binding>

2.9 Transport Security

Standard	To transact secure communications between applications either the TLS 1.0 (preferred; as found in http://wp.netscape.com/eng/ssl3/draft302.txt) or the SSL 3.0 standard (as found in http://www.ietf.org/rfc/rfc2246.txt) MUST be used. These standards are protocols that allow applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. The Version 1.0 of the Transport Layer Security (TLS) and its predecessor SSL 3.0 provides this communications privacy facility over the Internet.
	• When service and/or client authentication is required, HTTPS must be used.
	• For service authentication, server-side cert with a well-known root authority will be used.
	• For client-side authentication, one of two options MUST be used:
	  ○ A userid and password issued by the service, and used in a HTTP Basic Authentication inside the TLS/SSL tunnel.
	  ○ A client-side cert with a well-known root authority.

Justification	This profile defines the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) are very similar. They are cryptographic protocols that provide secure communications on the Internet for such things as web browsing, e-mail, Internet faxing, instant messaging and other data transfers. TLS protocol is based on the SSL 3.0 protocol specification as published by Netscape, they both are very similar.
Advice	There are enough differences between SSL 3.0 and TLS 1.0 that they cannot interoperate; however, TLS 1.0 does incorporate a mechanism by which a TLS implementation can back down to SSL 3.0. It MAY be necessary for interactions to use SSL to accommodate legacy systems.

2.10 Stateless

Statement	WS interfaces SHOULD NOT be designed such that the validity or semantics of a WS invocation is dependant on the history of prior WS invocations by that client.

Rationale	• Maintenance of session state adds complexity to service providers:
	  ▪ limit scalability,
	  ▪ dramatically increases the cost of deploying in a HA / DR context
	  ▪ introduces new failure modes
	  ▪ limits the use of routing / load balancing
	• Provider resources (e.g. memory) used to maintain session state for one user are no longer available to be shared by all
	• Alternatively, if state must be maintained (e.g., for a Shopping Cart app), the service infrastructure should make it globally accessible to all service instances, so as to allow load-balancing and failover.

	• The design of service interfaces MUST NOT depend on implicit, shared knowledge created through a sequence of interactions between a specific requester and provider.
Implications	• Tools and performance may be affected. Therefore delegate the state management to the layer using the services • If state is required to be maintained, it SHOULD be done at the application-level, so it is not dependent on the transport used (for future transports, although the standard today only specified HTTP transport)

3. Further Information

Readers are asked to contact any of the following persons with any questions or requests for further information

Mr. Alan Doniger (Energistics, Chief Technology Officer)
e-mail: Alan.Doniger@energistics.org Tel+ +1 713 267 5124

<End of Document>