



Web Service Reliability Requirements

Draft Version 0.02

Copyright (C) OASIS Open (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to OASIS, except as needed for the purpose of developing OASIS specifications, in which case the procedures for copyrights defined in the OASIS Intellectual Property Rights document must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Document history	
Version 0.1	Document created. Editor: Szabolcs Payrits, Nokia Inputs: <ul style="list-style-type: none">• Web Services Reliability Ver1.0, January 8, 2003, C. Evans, D. Chappel, D. Bunting et al.• Nokia Web Service Reliability requirements, March 2003, Sz. Payrits, M. Gerendai



Contents

Web Service Reliability Requirements	1
1. Scope	4
2. References	5
2.1. Normative references	5
2.2. Informative references	5
3. Terminology and Conventions	6
3.1. Conventions	6
3.2. Definitions	6
The Specification.....	6
3.3. Structuring of requirements	6
4. Requirements	7
4.1. Business use cases	7
4.2. Architectural requirements	7
4.3. Usage of SOAP	8
4.4. Reliability features	8
4.5. Interoperability	9
4.6. Backward compatibility	9
4.7. SOAP intermediaries	10



1. Scope

This document provides requirements for the OASIS WS-Reliability standard.

The requirements based on the submitted input papers to the OASIS WSRM Technical Committee.

The requirements specified in this document shall be adhered to by the:

- OASIS WS-Reliability specification



2. References

2.1. Normative references

[RFC2119]	"Key word for use in RFCs to Indicate Requirement Levels" S. Bradner, March 1997. http://www.ietf.org/rfc/rfc2119.txt
[SOAP11]	Simple Object Access Protocol version 1.1, W3C Note Don Box et al. , 8 May 2000
[SOAP12-1]	SOAP Version 1.2 Part 1: Messaging Framework, W3C Candidate Recommendation Martin Gudgin et al., 19 December 2002 http://www.w3.org/TR/soap12-part1/
[SOAP12-2]	SOAP Version 1.2 Part 2: Adjuncts, W3C Candidate Recommendation Martin Gudgin et al., 19 December 2002 http://www.w3.org/TR/soap12-part1/
[WS-I Basic Profile]	Web Service Interoperability Group Basic Profile

2.2. Informative references



3. Terminology and Conventions

3.1. Conventions

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY” and “OPTIONAL” in this document are to be interpreted as described in [RFC2119].

All text in all sections and appendices are normative in their scope unless they are explicitly indicated to be informative.

The figures in this document are for illustration for the normative referencing text. The figures themselves are informative.

3.2. Definitions

The Specification	Denotes the future WS-Reliability specification that is the output document of the TC's work.

3.3. Structuring of requirements

Requirements set by this document are structured in a way that all requirements can contain any number of sub-requirements that refine, clarify or specialize a general requirement for a special case.

This classification of requirements are denoted by numbering of the requirements. The number associated with a requirement consists of numbers separated by dots. The dot-separated numbers are to be read from left to right. The identifier associated with a requirements consist of a capital letter 'R' and the number above. This identifier should be in **bold**.

For example:

R1.23 and **R1.8** are two requirements defined by this specification, both being a sub-requirement of **R1**.

4. Requirements

4.1. Business use cases

Business use cases must be discussed.

R1.1	Electronic payment scenario... etc...
-------------	---------------------------------------

4.2. Architectual requirements

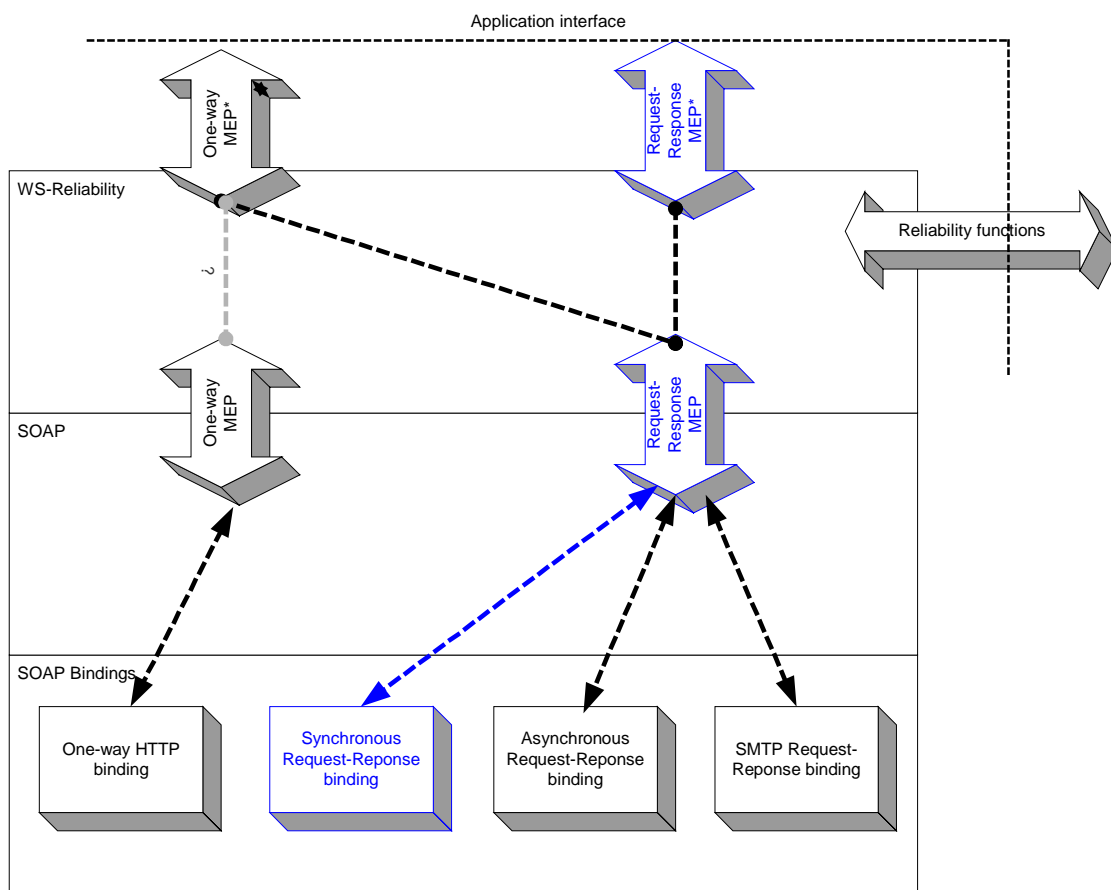


Figure 1: Web Services stack with WS-Reliability

R2.1	The implementation of the specification must fit into a layered architecture where WS-Reliability is a communication layer between the application and the SOAP layer.	
	R2.1.1	The specification of the behaviour of the communicating parties shall be described by state transitions of abstract state machines.
	R2.1.2	The communication of the WS-Reliability layer with the upper and lower layers shall be formally specified by abstract service primitives.



4.3. Usage of SOAP

R3.1	The Specification must adhere to the SOAP message construction rules. The basic messages generated by any implementation of the Specification must be compliant to the either the SOAP 1.1 or SOAP 1.2 message format.	
	R3.1.1	The Specification must prescribe the usage of the different SOAP versions in a consistent way. Therefore, it must be forbidden to mix different SOAP versions

R3.2	All information under control of the Specification must be included in well-defined SOAP header entries.
-------------	--

R3.3	The Specification must support the standard information exchange models of SOAP : the one-way messaging model and the abstract Message Exchange Patterns.	
	R3.3.1	The Specification must support a One-Way Message Exchange Pattern as defined in [WS-I Basic Profile].
	R3.3.2	The Specification must support the standard Request-Response Message Exchange Pattern defined in [SOAP11] and [SOAP12-2].

R3.3	The Specification must be SOAP binding agnostic.	
	R3.4.1	The Specifiction must support standard HTTP bindings defined in [SOAP11] and [SOAP12-2].
	R3.4.2	The Specification must able to work both with “synchronous” or “asynchronous” type of SOAP bindings.
	R3.4.3	It must be possible to change the SOAP binding without any change in the WS-Reliability implementation.

4.4. Reliability features

R4.1	The Specification must address Guaranteed Delivery as a reliability feature. The participating entities must be able to ensure that all application-level information to be sent to the party has actually been received or error reported.
-------------	--

R4.2	The Specification must address Duplicate Elimination as a reliability feature. The participating entities must be able to ensure that all duplicated application-level information is filtered out during the information exchange and is <u>not</u> received as duplicated.
-------------	---

R4.3	The Specification must address Ordering as a reliability feature. Communication between parties usually consist of several individual Message Exchange Patterns
-------------	--

R4.4	It must be possible to combine any of the functionalities in R3.1, R3.2, R3.3 in a given message exchange scenario.
-------------	--

R4.5	There are fatal cases when any of R3.1, R3.2, R3.3 cannot be absolutely guaranteed. The Specification must be able to support these cases.	
	R4.5.1	The specification must support use cases when any of R3.1, R3.2, R3.3 cannot be guaranteed. In these cases relevant parties must be informed about the fatal error occurred.

R4.5	The specification must address Crash Tolerance as a reliability feature.		
	R4.5.1	Usage of persistent storage must be supported by the Specification.	
	R4.5.2	Usage of persistent storage must not be mandated by the Specification.	
	R4.5.3	Levels of message persistency must be defined by the specification.	
		R.4.5.3.1	Involved entities must be able to communicate the supported persistency level on-the-fly.

4.5. Interoperability

R5.1	Web Service stacks with the implementations of the Specification shall be able to communicate with Web Services stacks without the implementation of the Specification.	
	R5.1.1	The decision whether a given message exchange is about to happen in reliable or non-reliable mode shall be a runtime decision.

4.6. Backward compatibility

R2.2	A Web Services stack with an implementation of the Standard must not offer less capabilities than a Web Services stack without the implementation of the standard.	
	R2.2.1	It shall be possible to create an implementation of the Standard where the interface of the a realiable message exchange is a backward-compatible extension of the interface of a non-reliable message exchange.



4.7. SOAP intermediaries

R7.1	An implementation of the Specification must be able to work in a scenarios where SOAP intermediaries are involved in the message path. [What should it exactly mean. Do we need more detailed text in this document]?
-------------	---