IT- og Telestyrelsen
Ministeriet for Videnskab
Teknologi og Udvikling

OWSA Profile *Reliable Messaging* version 0.8

OIO **W**eb **S**ervice **A**rchitecture Profile RM

Draft

for public review

National IT and Telecom Agency, Copenhagen, March 8, 2006

OWSA model RM

Colophon:

OIO Web Service Architecture model Reliable Messaging

The OWSA reference models are recommended by the National OIO Enterprise Architecture Committee and developed by National IT and Telecom Agency, IT Strategic Office as secretariat for the National OIO Enterprise Architecture Committee. …

Contact:

IT architect Mikkel Hippe Brun

E-mail: mhb@itst.dk

Phone +45 3337 xxxx (direct)

Authors:

IT Architect René Løhde, IT Strategic Office

IT Architect Nils Bundgaard, Ramboll Management

IT Architect Claus Andreasen, Ramboll Management

IT Architect Tommy Pedersen, Simmark

The Danish Ministry of Science, Technology and Innovation
National IT and Telecom Agency
IT Strategic Office
Holsteinsgade 63
DK-2100 Copenhagen
Tel. +45 35 45 00 00
Fax. +45 35 45 00 10
http://www.itst.dk
itst@itst.dk

1 Introduction

This document describes the **O**IO **W**eb **S**ervice **A**rchitecture model **R**eliable **M**essaging abbreviated OSWA model RM. This reference model represents an operational step towards providing a service-oriented architecture. OWSA model RM is of more reference models, which will be prepared by the Ministry of Science, Technology and Innovation (MTVU) in collaboration with authorities and the sector of business. Each model will sort out a range of quality needs, required by the authorities regarding interoperability, and each model will be tested in an actual implementation.

The models have been designed to:

- Ensure uniformity in realisation of the principles in OIO's *White Paper on Enterprise Architecture* covering interoperability, security, transparency, flexibility and scalability through use of standards for web services
- Make it easier for vendors to develop web services that work well in practice across public organisations/offices and IT platforms.

OWSA model RM is one of several profiles, which can be used as building blocks for Web Services with well defined properties. As the name indicates this profile defines the properties required for implementing reliability in messages exchanges. OWSA model RM requires OWSA model Basic Profile (BP).

1.1 Objective

The objective of the Model RM is to provide guidelines to enable the sender and receiver (service provider and service consumer) to ensure a high level of reliability in the use of web services which are otherwise often based on unreliable transport systems (such as HTTP).

Through use of Model RM, reliability on the message level is achieved without loss of flexibility regarding platform and transport protocols. This profile provides quality properties of OWSA Basic Profile architecture, as described in this document.

1.2 Target groups

- Solution-designers such as
  - IT architects
  - System consultants
  - Developers with design responsibilities (the reference model does not include platform-specific guidelines for implementation
- Project managers.

1.3 Goal

The goal of the OWSA model RM is to:

- Create a common basis for reliability implemented at the message level
- Ensure that use does not require specific systems or transport protocols
- Establish a technological consensus regarding the kind of technologies used for reliability and methods for use.

The model RM is based on the following elements:

- OWSA Model Basic Profile 1.0
- WS Reliable Messaging

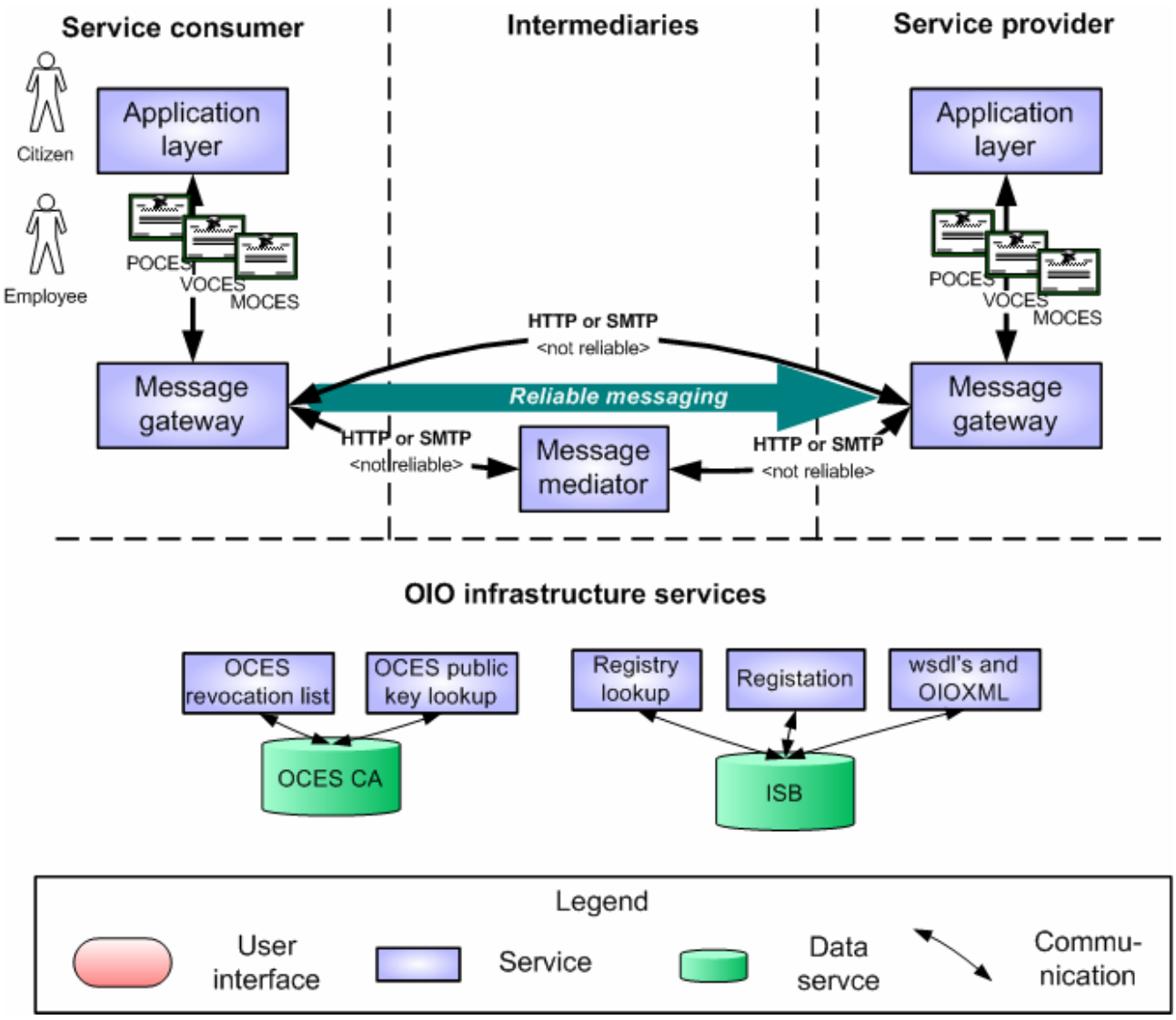
1.4 OWSA model RM and existing services

OWSA models should be used in connection with existing services when there is a strategic reason to modernize these services. One of the reasons could be that while the service provider is attempting to comply with a model, interoperability problems are eliminated and this ensures that the service provider’s services are implemented as all other services in the public sector. OWSA models are thus intended for use in new services or in connection with modernizing/upgrading efforts.

The recommendations do not in other ways affect existing web services or change the responsibilities and requirements for existing solutions.

1.5 Summary

2 Architectural overview



2.1 The logical service structure components

Service consumer

The service consumer usually uses a system where the application layer is separate from the Reliability handling, so that the business strategy is not affected by the RM implementation. A separate RM layer ensures that messages (typically) are to reach their goal once and only once. The RM layer handles resending if a positive delivery receipt has not yet been received.

Intermediaries

This model is not dependent on intermediaries. However, an important strength in Reliable Messaging is that reliability is implemented on the message level and intermediaries do not need to have insight into how sender and recipient ensure delivery via Reliable Messaging.

There is no requirement that a specific technology or transport protocol must be used the entire path from sender to the final recipient (service consumer).

Service provider

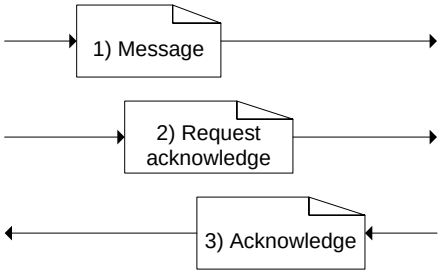
The service provider also has a system where the application layer is separate from the Reliability handling so that business strategy does not depend on whether it has received a given number of incoming messages, but because of RM implementation, this has actually occurred – and that a delivery receipt has been sent.

OIO infrastructure services

Services and XML schemas are registered in ISB.

Reliable Messaging - receipts

Implementation of Reliable Messaging can be basically described as a method to ensure at message level that after sending of message(s), a technical delivery receipt is requested to document the delivery. Any message that may be missing from the list of received messages will then re-sent.



In addition to the above, simplified illustration showing the sending of a message, reliable messaging can meet other requirements than one and only one delivered message. E.g. reliable messaging can also ensure the order of messages. Negative ackowledgements may be used allowing to list only messages that are missing instead of a long list of messages actually received.

2.2 Architectural properties

Characteristic architectural quality properties for the OWSA model RM, and how they implement principles from OIO's *White Paper on Enterprise Architecture*:

OIO White Paper principle	**Architecture property**		**Technology characteristics**	***Architecture characteristics***
Interope-rability	Integrati on, interface s and commun	External communicatio n and interfaces	Physical network and topology	*Not provided*
			Transport protocols	*Not provided*
			Service specifications	*Not provided*

OIO White Paper principle	**Architecture property**		**Technology characteristics**	***Architecture characteristics***
	ication	Locations	Physical location	*Not provided*
			Network segments	*Any two end-points Reliable Messaging enabled.* *Intermediaries are not significant as Reliability is ensured at message level and does not depend on the transport layer.*
		Reliability	Acknowledgements through use of non-application level SOAP messages	*WS-ReliableMessaging (new specification under OASIS - WS-RX).*
		Quality of information and data	Data reference integrity	*Not provided*
	Semantics	Information	Data and combination of data for consolidated data	*Not provided*
		Terminology	Data syntax	*XML and XML schema (XSD)*
			Data semantics	*Not provided*
		Source	The source responsible for data quality	*Not provided*
Security	Security and identity management	Identification / Authentication	User data.	*Not provided*
			Digital certificates (PKI)	*Not provided*
		Integrity	Encryption	*Not provided*
		Confidentiality	Encryption	
		Authorisation	Resource access rights	*Not provided*
		Non-repudiation	Signature	*Not provided*
		Maintainability	Identity management	*Not provided*
Flexibility	Resiliency to change and flexibilit	Work structure	The functions to be specified as service specifications and their interconnection Components, services,	*Not decided*

OIO White Paper principle	**Architecture property**		**Technology characteristics**	***Architecture characteristics***
			composites, applications, systems, layering, service orientation	
	y	Readiness for change	Enclosure of services and data Granularity of enclosure	*Not decided*
		Resiliency to change	The component's dependence on a specific infrastructure platform	*Reference implementations are conducted on on a Microsoft .NET and a Java platform.*
Scalability	Stability and capacity	Work volume	Work capacity	*Not decided*
		Work variations	Scalability in terms of variation and development of capacity	*Not decided*
		Response time	Response time from involved parties activates the service until result comes back again	*Not decided*
		Opening hours and uptime	Opening hours: The time period during the day when the services are available	*Not decided*
			Uptime: The percentage of the opening hours when the service **must** answer	*Not decided*
		Recovery	The time it takes to download the backup and start in a consistent level	*Not decided*

3 Reliability

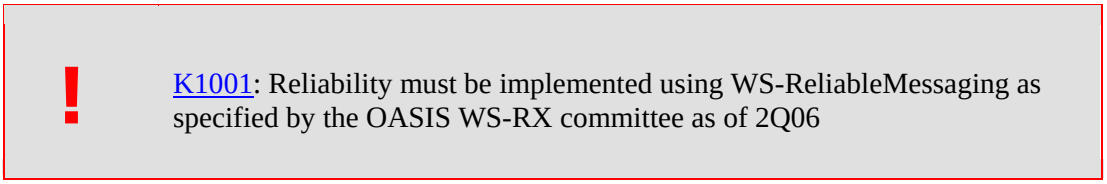
3.1 WS-ReliableMessaging

The basic WS standards SOAP and WSDL are not responsible for ensuring reliability in the delivery of messages. Reliability is provided by Model RM via WS-ReliableMessaging. This usually means that a message is received once and only once (but other procedures are possible – e.g. *at least one* time). In addition, ReliableMessaging can ensure that the order of a series of logically connected messages is maintained, so that the application layer receives these messages in the intended (sent) order.

WS-ReliableMessaging is under specification by OASIS WS-RX committee bringing together former work from two different initiatives:

Gartner, Inc. Research Director, Charles Abrams, noted, "Reliable messaging is key to the future growth of business-to-business Web services. The need for secure and verifiable transactions among trading partners is paramount, for both narrow supply chains and broad industry groupings. Until now, there have been two duplicative efforts, WS-Reliability, an OASIS Standard, and WSRM, a major vendor-backed specification effort. This has caused some end user confusion and concern. Now that both efforts are under the auspices of OASIS, with major vendor support from both WS-Reliability and WSRM, end user needs should be better served. Eventual reconciliation of both efforts under the administration of OASIS, which is directing more web service standards development efforts than any other organization, will measurably progress advanced web service utilization."

As of 2Q06 WS-ReliableMessaging is the best bet for the coming standard, which is why OWSA RM chooses to rely on it. Instead of everyone defining their own set of acknowledgements and the exchange of these, this offers a common practice, which is not far from the coming standard. One must prepare though in the implementation for being able to correct for the final standard in future upgrades. The OWSA RM will be updated one version when the standard stabilizes.



Work under the OASIS WS-RX committee:

http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=ws-rx

The draft specification can be downloaded from this URL:

http://www.oasis-open.org/committees/download.php/15177/wsrm-1.1-spec-cd-01.pdf

According to the WS-ReliableMessaging specification, each reliable message sequence must be initiated with a *create sequence* request to be accepted with a *create sequence response*.

Example 1: *wsrm:CreateSequence*

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" ...>

<soap:Header>

</soap:Header>

<soap:Body>
       <wsrm:CreateSequence ...>
              <wsrm:AcksTo ...> wsa:EndpointReferenceType </wsrm:AcksTo>
              <wsrm:Expires ...> xs:duration </wsrm:Expires>
              <wsrm:Offer ...>
                     <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
                     <wsrm:Expires ...> xs:duration </wsrm:Expires>
              </wsrm:Offer>
       </wsrm:CreateSequence>

</soap:Body>

</soap:Envelope>
```

Example 2: *wsrm:CreateSequenceResponse*

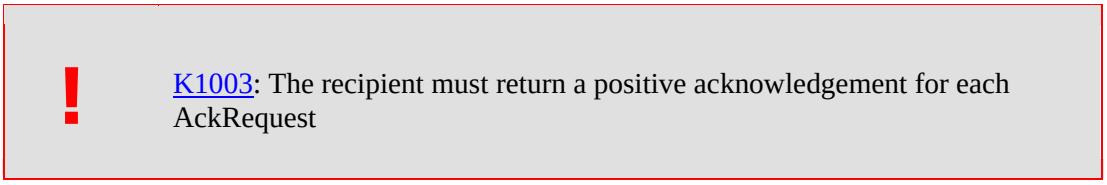
```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" ...>

<soap:Header>

</soap:Header>

<soap:Body>
       <wsrm:CreateSequenceResponse ...>
              <wsrm:Identifier ...> xs:anyURI </wsrm:Identifier>
              <wsrm:Expires ...> xs:duration </wsrm:Expires>
              <wsrm:Accept ...>
                     <wsrm:AcksTo ...> ... </wsrm:Expires>
                     <wsrm:Expires ...> ... </wsrm:Expires>
              </wsrm:Accept>
       </wsrm:CreateSequenceResponse>

</soap:Body>

</soap:Envelope>
```

After sending the message or a series of messages, the sender asks for a technical delivery receipt – in other words, an acknowledgement. Reliable Messaging uses positive (”I have received 2,4 and 5”) as well as negative (”I have not received 1,3 and 6”) acknowledgements. The profile is based on an explicit demand for acknowledgement (instead of ongoing and implicit via LastMessage) as well as positive receipts, with a list of all received messages.

K1002: The sender must include a AckRequested for each message or series of related messages with the need for guaranteed delivery

Example 3: *wsrm:AckRequested*

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" ...>

<soap:Header>

       <wsrm:Sequence>
              <wsu:Identifier>http://…/seq123</wsu:Identifier>
              <wsrm:MessageNumber>1</wsrm:MessageNumber>
       </wsrm:Sequence>
       <wsrm:LastMessage>
       <wsrm:AckRequested>
              ...
       </wsrm:AckRequested>

</soap:Header>

<soap:Body>

       ...
</soap:Body>

</soap:Envelope>
```



K1003: The recipient must return a positive acknowledgement for each AckRequest

Example 4: *wsrm:SequenceAckowledgement*

```
<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://
www.w3.org/2001/XMLSchema" ...>

<soap:Header>

       <wsrm:SequenceAcknowledgement>
              <wsu:Identifier>http://…/seq123</wsu:Identifier>
              <wsrm:AcknowledgementRange Upper=”1" Lower=”1"/>
       </wsrm:SequenceAcknowledgement>

</soap:Header>

<soap:Body>

       ...

</soap:Body>

</soap:Envelope>
```

[Are there any known scenarios where use of explicit AckRequest and positive Acks are not the proper way?]

4 Development and Test

[Any real life experience that calls for ”how to develop and test” guidelines?]

5 Policy

In this specific situation, a decision must be made as to the interval between attempts to resend this message/series of message(s) and how many times to try before giving up.

Appendix - Checklist

In order to implement a Web Service according the OWSA Model [RM], the following requirements must be met:

ID	Specification
K1001	Reliability must be implemented using WS-ReliableMessaging as specified by the OASIS WS-RX committee
K1002	The sender must include a AckRequested for each message or series of related messages with the need for guaranteed delivery
K1003	The recipient must return a positive acknowledgement for each AckRequest