

Preliminary Minutes WSRM Face To Face Meeting

Wed May 28

## **1 Introductions**

### **1.1 Proposed Agenda**

Wed May 28 (8:30 AM) thru Friday May 30 (12:30 PM)

Conference Bridge Details:

(8:30 AM to 10:30 AM, and 3:00 to 5:00 PM) Wed and Thur

10:30 AM to 12:00 AM Friday

Dial-In number: 877.302.8255

International Dial-In number: 303.928.2609

Conference ID: 4541308

Wednesday:

AM:

8:00 Continental Breakfast

8:30 Introductions and Review of Agenda

9:00 Roll Call, identification of input contributions, and Minutes approval

32 voting members - 16 present,

9:30 Discussion and Resolution of Requirements Issues

Rel 8

Intermediary is defined in Soap. Clearer in Soap 1.2

Active vs passive Intemediary – Sunil states we do not have an issue here.

IM changing MessageID is considered harmful.

Is IM changing any WSRM Header field is considered harmful?

Payrits – keeping silent is not acceptable. Need to define how an intermediary node can behave, in processing headers.

Discussion on ensuring that a participating intermediary is used for both the request and ack path.

Message Exchange pattern vs. Header fields.

NAT at TCP layer is transparent to soap layer, thus it is a non-participating intermediary.

10:30 Break

10:45 Requirements Issues Cont.

PM

- 12:00 Lunch
- 1:00 Requirements Issues Cont
- 2:30 Break
- 3:00 Discussion of Requirements Document
- 5:00 Homework Assignments, meeting closes 5:30 PM

Thursday:

AM:

- 8:00 Continental Breakfast
- 8:30 Review of WS-Reliability Input Specification and schedule planning
- 10:30 Break
- 10:45 Discussion / Resolutions f WS-Reliability Specification Issues

PM

- 12:00 Lunch
- 1:00 Requirements Issue Discussion
- 2:30 Break
- 3:00 Requirements Issue Formal Resolutions / Requirements Doc Review
- 5:00 Homework Assignments, meeting closes 5:30 PM

Friday:

AM:

- 8:00 Continental Breakfast
- 8:30 Review of Homework Assignments
- 10:15 Break
- 10:30 Concluding Discussion / Resolutions
- 11:30 Future Meeting Planning
- 12:00 Meeting Adjourns / Lunch

Sunil pointed out that Lunch will not be served on Friday, and that the afternoon breaks need to be at 3:00 PM rather than 2:30.

## **1.2 Roll Call**

First Name	Last Name	Company	Email	
Nobuyuki	Yamamoto	Hitachi (by phone)	no_yama@bisd.hitachi.co.jp	F
Magdolna	Gerendai	Nokia (by phone)	magdolna.gerendai@nokia.com	M
Jeff	Turpin	Cyclone Commerce	jturpin@cyclonecommerce.com	M
kiwasa	kiwasa	Fujitsu	kiwasa@jp.fujitsu.com	M
Tom	Rutt	Fujitsu	tom@coastin.com	T
Eisaku	Nishiyama	Hitachi	nishiy_e@itg.hitachi.co.jp	M
Mark	Hansen	Individual	khookguy@yahoo.com	M
paolo	romano	Individual	romanop@dis.uniroma1.it	M
Venkat	Danda	IONA	venkat.danda@iona.com	M
Dock	Allen	Mitre Corporation	Dock@mitre.org	M
Alan	Weissberger	NEC Corporation	ajwdct@technologist.com	M
Szabolcs	Payrits	Nokia	Szabolcs.Payrits@nokia.com	M
Sunil	Kunisetty	Oracle	Sunil.Kunisetty@oracle.com	M
marc	goodner	SAP	marc.andrue.goodner@sap.com	M
Pete	Wenzel	SeeBeyond	pete@seebeyond.com	M
Doug	Bunting	Sun	doug.bunting@Sun.com	M
Scott	Werden	WRQ	scottw@wrq.com	M
Pramila	Mullan	France Telecom	pramila.mullan@rd.francetelecom.com	F
Ricky	Ho	Cisco (observer)		
Junichi	Tatemura	NEC Corporation	tatemura@ccrl.sj.nec.com	F

Voting members 31, Quorum 16. Reached Quorum, even without Phone attendees.

### 1.3 Approval of Prior Minutes

<td on Thursday or Friday>

## 2 Wednesday Requirements Issues Discussion

### 2.1 Rel 8 - Intermediaries

Intermediary is defined in Soap. Clearer in Soap 1.2

Message Exchange pattern vs. Header fields.

NAT at TCP layer is transparent to soap layer, thus it is a non-participating intermediary.

Are SOAP intermediaries store and forward entries that also implement WSRM over SOAP? Or are they strictly soap entities that are unaware of wsrn

Discussion:

SOAP 1.2 has a better definition of intermediary.

Definitions for participating intermediary – participant in the formal message exchange pattern

And non-participating intermediary – transparent to the soap message exchange pattern

If the sender wants an intermediary to do something the sender will have to define it. If the intermediary is active, does it send an id, does it send an ack,

Proposed by nokia.

Do we want to deal with intermediaries that change the message exchange pattern? The intermediary cannot change the message header field especially the message id.

IM changing message id is considered harmful.

Is IM changing any wsrn header field considered harmful?

Payrits - keeping silent is not acceptable. Need to define how an intermediary node can behave in processing headers.

We need to clearly specify the behaviours of the intermediary with respect to the soap header.

If you expect the intermediary to impact flow or fault management then you need to specify behaviours of the intermediary.

Either you send a message to the intermed which returns an ack and then forwards to the next node.

Other solution an intermediary forwards the message to the next node and then.

NAT at TCP layer is transparent to soap layer thus it is non-participating intermediary.

Paolo: Two concepts end to end reliability vs point to point – both are possible end user defines what type of reliability is supported. Borrowed from ebxml spec.

Alan – if you allow intermediaries to change the messages then you have to specify the limits on what they can do.

Dock: How does the presence of an intermediary impact generation of fault messages and acks. Only an issue if participating intermediary i.e. can impact M.E.P.

Paolo – could express intermediaries directly in the Header

Scott – stay away from routing!

Message Exchange Pattern is defined in SOAP 1.2

Could we allow extensions in wsrn for other routing protocols? Sunil says no!

Dock – start with intermediaries which do nothing at soap layer – add capabilities required one by one.

Pete – one pattern for actor next receiver – rm header targeted at ultimate receiver

One pattern for both next receiver and ultimate receiver.

WS-Rel does not state anything about actor – protocol does not use actor. Original author was end to end protocol between ultimate sender and ultimate receiver;. Intermediaries were transparent.

Discussion on what ws rel spec assumes about soap actor. Symbols state from/to attributes overlaps with soap actor. This causes a problem. Could instead use soap actor. Other alternative is to not use it.

If an intermediary is not acting in the role specified it cannot touch that header. Soap defined intermediary was intended as pass thru only. Soap intermediary processes only those headers for which it is the actor.

Alan – rechanging the destination – back up receiver when primary path has failed. An intermediary does backup rerouting. In this case, the intermediary would change the path and destination by changing the header – to field to another destination.

Doug – URI message is sent to

Doug – WS Rel schema does not use soap actor – therefore the recipient is the ultimate destination of the soap message.

Doug – intermediaries are out of scope vs intermediaries must not process the wsrn headers.

Suggestion – to make use of intermediaries outside of scope of spec.

We need to agree on what architectures and what technical models we are supporting.

Question on whether the decision on soap actor is precluded by this decision.

Pete – the ws rel spec disallows intermediaries to ack since we do not allow soap actor attribute.

Attempt by Sunil at straw proposal:

- a) intermediaries out of wsrn scope, soap:actor excluded in schema
- b) intermediaries out of wsrn scope, soap:actor optional
- c) describe intermediary behaviour in detail

Doug has different straw poll:

Schema:

- Allow optional soap actor vs.
- Disallow soap actor (as in WS-Rel V1.0)

Text:

- Describe whatever we do in schema
- Be strong about excluding soap actor use
- Describe Reliable delivery with intermediaries
- Describe Reliable delivery with actor to actor
- Describe Reliable Delivery with end to end (as in WS-Rel V1.0)

Sunil – dealing with this extra stuff is out of scope for WS-RM

Paolo – since there is no Routing standard, we could only allow end to end. However this is a big limitation.

Suggest limit scope to end to end reliability, and any intermediaries must be transparent to end to end view of protocol.

Payrits: Support reliability for end to end. Where one end is ultimate destination, other end is sender.

Dock: this leads to intermediaries not being able to change the behaviour of the protocol.

Support architecture of end-to end, thus any intermediaries may not change messages.

Sunil: this could be an extensibility.

Proposed to resolve issue with a new Requirements:

- WSRM must only support end to end reliable messages, where one end is the sender, and the other end is the ultimate destination. Because of this there is no need to define intermediaries.

Payrits stated another potential requirement:

- Spec needs to cover the fault case when the processor of the header is not the ultimate destination. This is not defined in the soap processing model.

There were several concerns expressed regarding this additional requirement. Since the system receiving this message is not participating in the protocol, how can we make statement about what that system should do.

**Sunil moved, Dock Seconded. Resolve rel 8 with following requirement:**

**WSRM must only support end to end reliable messages, where one end is the sender, and the other end is the ultimate destination. Because of this there is no need to define intermediaries.**

- **No opposition, motion to resolve rel 8 passes.**

## **2.2 Rel 004:**

From Dave Ingham:

Synchronous SOAP HTTP binding:

We say that a synchronous SOAP HTTP binding is in use if the outbound Reliable Message is sent in the HTTP POST request and the Acknowledgment Message is contained in the corresponding HTTP response message. We refer to this binding as "synchronous" in order to imply the WS-RM exchange inherits the blocking behaviour of the HTTP exchange.

Asynchronous SOAP HTTP binding:

We say that an asynchronous SOAP HTTP binding is in use if the Acknowledgement Message is sent in a separate HTTP POST request/response exchange from the outbound Reliable Message. We refer to this HTTP binding as "asynchronous" in order to emphasize that the WS-RM message exchanges are not tied to the blocking request/response behaviour of the HTTP transport. Two alternate usage patterns are possible. We say that the "callback" pattern is being used if the Acknowledgement Message is contained in the HTTP POST request of a second HTTP exchange operating in the opposite direction to the one containing the outbound Reliable Message. We say that the "polling" pattern is being used if a second HTTP POST request is issued in the same direction as the one containing the outbound Reliable Message to act as a request for acknowledgement. The Acknowledgement Message is contained in the HTTP response to this request. This polling pattern is expected to be used in situations where it is inappropriate for the sender of reliable messages to receive HTTP requests.

Chris Ferris, through Mark Little, suggested using new terms (since http post does not imply blocking if HTTP pipelining is implemented) such as:

Http Response Acknowledgement Pattern  
Callback Acknowledgement Pattern

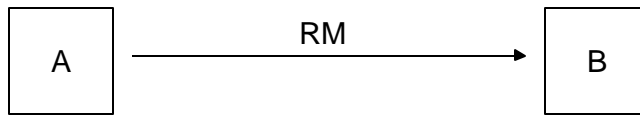
This led to discussion of new terms for binding (i.e, mapping) soap/wsrn layer to the Transport layer:

Response Acknowledgement Pattern  
Callback Acknowledgement Pattern  
Polling Acknowledgement Pattern

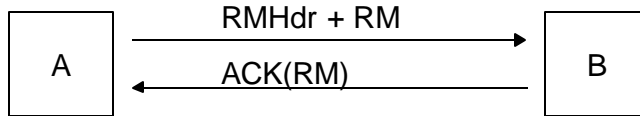
Each of these involves a different binding of soap/wsrn layer to the underlying transport.

Payrits drew the following diagrams on the whiteboard, to illustrate his point about bindings between layers:

WSRM User / Application Layer



SOAP / WSRM Layer



Underlying Transport Layer

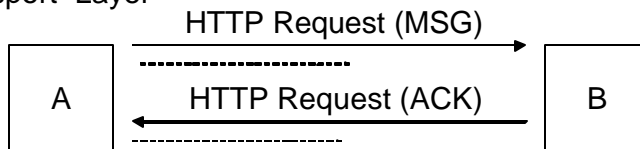
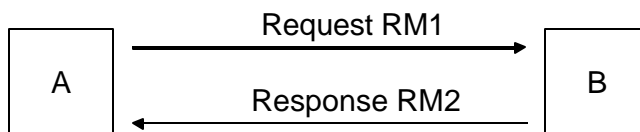


Figure – One way MEP interlayer binding examples

WSRM User / Application Layer

Request /Response MEP



SOAP / WSRM Layer

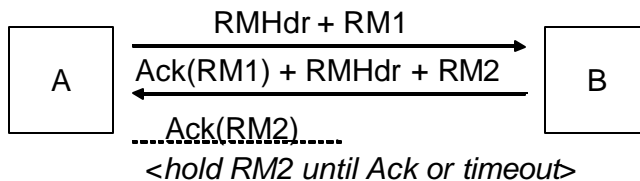
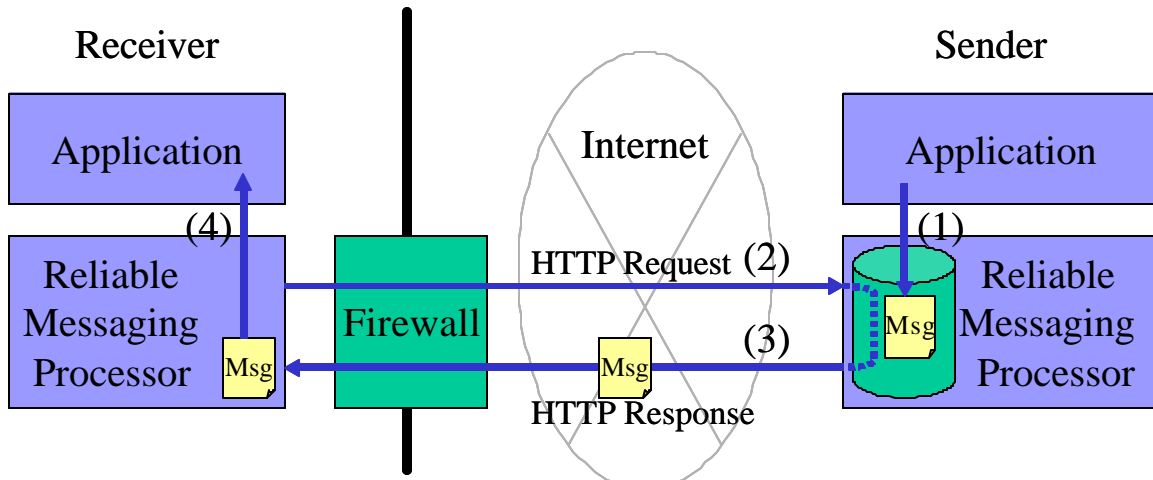


Figure – Request response MEP inter layer Binding Example

### 2.3 Rel 09 Pull model

Iwasa presented his two slides.

## 2.Firewall consideration - Pull model



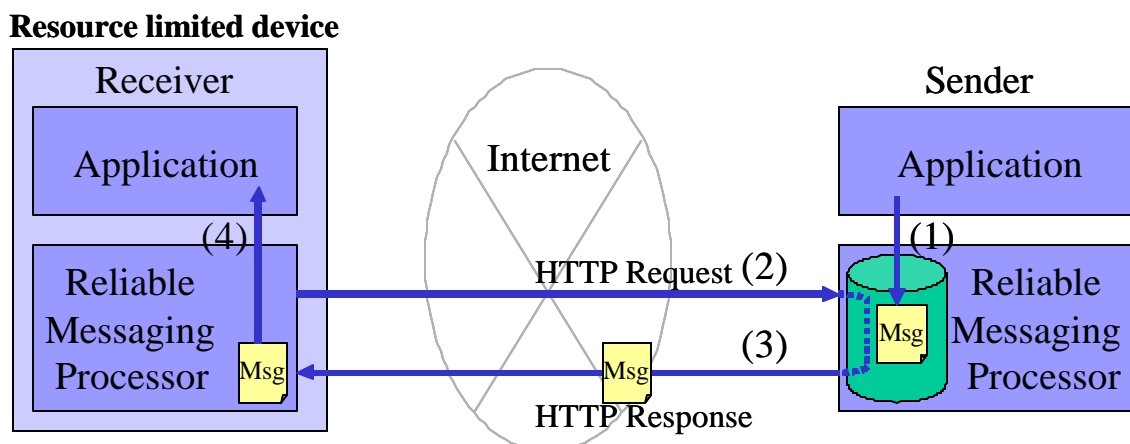
\* The sender can't initiate a HTTP connection to the receiver, since receiver's firewall doesn't accept the connection from outside.

- (1) Sender application puts a message to be sent to Receiver in a storage at Reliable Messaging Processor. The message has an identification who can retrieve it.
- (2) and (3) Receiver is polling a message stored for the receiver every XX minutes. When the receiver found any messages, it will retrieve the message in the HTTP response.
- (4) The receiver give the message to its application.

Figure – Firewall use case for Pull capability



### 3.Limited resource device consideration - Pull model



\* The sender can't initiate the HTTP connection to the receiver, since receiver has no IP address. Or receiver may not have capability to receive a message incoming.

- (1) Sender application puts a message to be sent to Receiver in a storage in Reliable Messaging Processor. The message has an identification who can retrieve it.
- (2) and (3) Receiver is polling a message stored for the receiver every XX minutes. When the receiver found any messages, it will retrieve the message in the HTTP response.
- (4) The receiver send give the message to its application.

Figure – Limited system use case for Pull capability

It was pointed out that these figures do not show the ack phase of reliable message protocol. Iwasa agreed that that would be required, but was not shown for simplicity.

Discussion turned to how to realize the requirement (receiver of Reliable message cannot receive underlying protocol request).

Tom R stated this could be treated as a special case of Request/Response MEP at WSRM user level, where the request has a empty body.

Payrits stated that another solution could be a special request.

Marc G stated that there is more than one way to meet the requirement.

Payrits presented a use case for mobile phones:

He presented two solutions, one requiring storage by the WSRM user, the other mechanism the storage is done by the WSRM layer.

Marc G stated that is we agree to this as a requirement, we could defer on the particular solutions.

Possible Requirement:

WSRM protocol must allow Reliable messages to be received by an endpoint which cannot accept an underlying protocol request message or underlying protocol one-way message.

Question is whether we need a separate requirement from the already accepted requirement for supporting reliable request response MEP.

Iwasa asked that we leave this Issue open, to allow time for him to consult his developers.

## 2.4 Rel 13-22 as Spec Issues:

Iwasa suggested moving issues Rel 13 thru rel 22 as spec issues:

#### **2.4.1 Rel 13**

Agreed to move Rel 13 as spec issue

#### **2.4.2 Rel 14 regarding wsdl definitions**

Tom Rutt suggested that we either make this a requirement or request a charter clarification:

Could change, in the charter:

“The specification to be created will provide WSDL definitions for reliable messaging and the message formats will be specified as SOAP headers and/or body content.”

To:

“The specification to be created will provide message formats specified as SOAP headers and/or body content.”

Tom Rutt stated that the WSS security people have the same problem. This is not specific to WSRM.

Doug Bunting stated that WSDL 1.2 features might be able to do this.

Alan suggested leaving it in charter, not including it as a requirements, and marking it in the spec as an item for further study in the specification.

Potential resolution:

Move this as a Spec level issue, and close with agreement that this will be marked in the spec as an item for further study, (i.e not solved in the first version of the spec.

Sunil suggested to leave it open as a spec Issue for now, and not make it a requirement.

Agreed to make this a spec issue, and leave open.

#### **2.4.3 Rel 15**

**Alan moved, Iwasa seconded, to resolve this issue with a new requirement:**

**“WSRM spec must identify fault cases and WSRM protocol must support the reporting of these identified faults.”**

- **No opposition, motion to resolve Rel 15 passes, agree to add new requirement.**

#### **2.4.4 Rel 16 -**

Agreed to move to spec issue

#### **2.4.5 Rel 17 – Persistence requirements**

Agreed to move configurability of persistence requirements as a spec issue.

#### **2.4.6 Rel 19 -**

We are no longer using terms synch/asynch. We have defined binding patterns, but have not yet decided which will be included in the spec’s protocol.

**Dock moves to make two new requirement. Scott seconded.**

**requirement for Spec having a solution for response Ack Binding pattern, for both one-way and request/response MEP.**

**requirement for Spec having a solution for Callback Ack binding pattern, for both one-way and request/response MEP.**

- **No opposition to adding two new requirements Motion passes. Issue still open for discussion of additional requirements.**

Sunil suggested we add a third requirement.

Proposal to have requirement for spec having a solution for Polling Ack Binding pattern, for both one-way and request/response MEP.

Pete stated this might be difficult for request/response MEP.

Payrits stated he does not need the polling Ack binding pattern for any MEP.

No time for further discussion, will continue on Thursday PM.

## **2.5 Homework:**

Tom told everyone to read the WS-rel spec, so we can walk thru the doc Thursday AM to find new Requirements and Spec Issues to add to Issue list.

Payrits will work on making a new requirements doc.

Meeting closed for the day at 5:30.