



WS-I RSP WG Usage Scenarios

1

2 **Document Status: WS-I Final Material**

3 **Version: 1.0**

4 **Date: December 15, 2008**

5 **Editors:**

6 Jacques Durand, Fujitsu (jdurand@us.fujitsu.com)

7 Marc Goodner, Microsoft (mgoodner@microsoft.com)

8 Charles Le Vay, IBM (ccl@us.ibm.com)

9 Ram Jeyaraman, Microsoft (ram.jeyaraman@microsoft.com)

10 **Notices**

11 © Copyright 2008 by the Web Services-Interoperability Organization. All rights
12 reserved.

13 The material contained herein is not a license, either expressly or impliedly, to
14 any intellectual property owned or controlled by any of the authors or
15 developers of this material or WS-I. The material contained herein is provided
16 on an "AS IS" basis and to the maximum extent permitted by applicable law,
17 this material is provided AS IS AND WITH ALL FAULTS, and the authors and
18 developers of this material and WS-I hereby disclaim all other warranties and
19 conditions, either express, implied or statutory, including, but not limited to,
20 any (if any) implied warranties, duties or conditions of merchantability, of
21 fitness for a particular purpose, of accuracy or completeness of responses, of
22 results, of workmanlike effort, of lack of viruses, and of lack of negligence.
23 ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT,
24 QUIET POSSESSION, AND CORRESPONDENCE TO DESCRIPTION OR NON-
25 INFRINGEMENT WITH REGARD TO THIS MATERIAL.

26

27 IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR WS-I BE
28 LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE
29 GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY
30 INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES
31 WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN
32 ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT RELATING TO THIS
33 MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE
34 POSSIBILITY OF SUCH DAMAGES.

35

36 **Executive Overview**

37 The RSP WG has decided to approach defining requirements for the RSP profile
38 in terms of realistic and detailed use cases, called usage scenarios.

39 This document describes these usage scenarios. These scenarios will serve as
40 detailed input for the profiling work, providing evidence of potential
41 interoperability issues and/or need for best practice guidelines.

42

43 **Table of Contents**

1	Introduction	4
44	1.1 Status of this Document	4
45	1.2 Role of this Document	4
46	1.3 Properties of Usage Scenarios	4
47	1.4 Artifacts and Specifications Coverage	4
	2 Definitions	6
	3 Conventions in Defining Scenarios	8
	4 Reliable One-way (ROW)	10
48	4.1 Description	10
49	4.2 Sequence Diagram	10
50	4.3 Scenario Constraints and Assumptions	11
51	4.4 Message Exchanges Details	11
	5 Reliable One-way, anonymous client (ROW-anon)	14
52	5.1 Description	14
53	5.2 Sequence Diagram	14
54	5.3 Scenario Constraints and Assumptions	15
55	5.4 Message Exchanges Details	15
	6 Reliable Request-Response (RRR)	18
56	6.1 Description	18
57	6.2 Sequence Diagram	18
58	6.3 Scenario Constraints and Assumptions	19
59	6.4 Message Exchanges Details	19
	7 Reliable Request-Response, anonymous client (RRR-anon)	22

60	7.1	Description	22
61	7.2	Sequence Diagram	22
62	7.3	Scenario Constraints and Assumptions.....	23
63	7.4	Message Exchanges Details	23
	8	MakeConnection Protocol.....	27
64	8.1	Use of the MC Anonymous URI	27
	9	Reliable, Secure Conversation Establishment and Cancellation.....	28
65	9.1	RequestSecurityToken, CreateSequence (RST-CS).....	28
66	9.2	TerminateSequence, Cancel (TS-Cancel).....	28
	10	Secure Request-Response (SRR)	30
67	10.1	Description	30
68	10.2	Sequence Diagram	30
69	10.3	Scenario Constraints and Assumptions.....	31
70	10.4	Message Exchanges Details	31
	11	Secure Request-Response, anonymous client (SRR-anon)	33
71	11.1	Description	33
72	11.2	Sequence Diagram	33
73	11.3	Scenario Constraints and Assumptions.....	34
74	11.4	Message Exchanges Details	35
	12	Revision History	37
75			

76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111

1 Introduction

1.1 Status of this Document

This document is an Editors Draft; it has not yet been accepted by the Working Group as reflecting the current state of discussions. It is a work in progress, and should not be considered authoritative or final. Other documents may supersede this document.

This document will be updated from time to time to incorporate new usage scenarios as they are identified.

1.2 Role of this Document

The usage scenarios in this document do not represent exhaustive ways to combine the specifications targeted for the RSP profile, but only those ways that seem to exhibit interoperability issues or that need guidance.

The usage scenarios in this document represent input material candidate for profiling, and should not be interpreted as best practices for integrating the specifications targeted for the RSP profile. The RSP profile may actually restrict them, or propose better alternatives.

Other patterns of usage that do not fit in these scenarios are legitimate as long as the final RSP does not preclude them. Conversely, some of these scenarios or their options, may later be precluded by RSP.

1.3 Properties of Usage Scenarios

A Usage Scenario is illustrative of real usage conditions, and of the rationale behind them. It describes assumed or possible environmental constraints, e.g. addressing, security, and reliability.

A Usage scenario details all contextual exchanges needed to enable it end-to-end (establishment of security context, or reliability sequences) and related options.

1.4 Artifacts and Specifications Coverage

The usage scenarios in this document involve the following Web services artifacts and specifications, subject to profiling, either individually or in composition:

Specifications:

- WS-I Basic Profile 1.2
- WS-I Basic Profile 2.0
- WS-I Basic Security Profile 1.0
- WS-I Basic Security Profile 1.1

- 112 • WS-ReliableMessaging 1.2
- 113 • WS-SecureConversation 1.4
- 114 • WS-MakeConnection 1.1

115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152

2 Definitions

The following terms will be used throughout this document to refer to the various factors that make up individual scenarios.

Addressable client: A client that is capable of accepting connections on a network endpoint.

Anonymous client: A client that does not accept incoming connections.

Application Traffic Message: A SOAP message containing application data.

Asynchronous request-response message exchange: A SOAP message exchange in which a requester sends a SOAP message to a service and receives a response message. "Asynchronous" in this context refers to the manner in which the underlying transport protocol is used to carry the request and response messages. The response message is sent over a separate connection to the requester (a "callback").

Message Exchange Unit: A unit representing a coherent atomic exchange of elements (and related messages).

One-way message: An application SOAP message for which no application SOAP response is expected.

Reliable messaging: The act of sending SOAP messages using the WS-ReliableMessaging 1.1 protocol.

Reliable message: A message sent reliably using the WS-ReliableMessaging 1.1 protocol.

Request message: An application SOAP message for which an application SOAP response is expected.

Response message: An application SOAP message triggered by a request message.

Secure messaging: In the general sense this term refers to the act of sending a message with one or more of the following security qualities: integrity, confidentiality, and authenticity. For the purposes of this document it is assumed that these attributes will be provided through the use of either SSL/TLS or WS-SecureConversation 1.3.

Sequence Lifecycle Message: A message that contains one of: CreateSequence, CreateSequenceResponse, CloseSequence, CloseSequenceResponse, TerminateSequence, TerminateSequenceResponse as the child element of the SOAP body element.

Sequence Traffic Message: A message containing a Sequence header block.

Synchronous request-response message exchange: A SOAP message exchange in which a requester sends a SOAP message to a service and receives a response message. "Synchronous" in this context refers to the way in which the

153 underlying transport protocol used to carry the request and response messages.
154 The response message is returned on the back channel of the request message.

155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176

3 Conventions in Defining Scenarios

A scenario may be viewed under different perspectives, which will be captured and represented differently in this document.

These main perspective lines are:

- Overall description and usage rationale.
- Sequence diagram describing the messages choreographies. These will show flow diagrams, where solid lines represent **requests** over an underlying protocol, and dashed lines represent **responses** sent back over the back-channel offered by the request.
- Constraints and assumptions underlying to the entire scenario (e.g. addressing constraints of one of the endpoints)

In addition, the message choreography as reported in the activity diagram can be decomposed as a sequence of *message exchange units*, a unit representing a coherent atomic exchange of elements (and related messages) such as CreateSequence/ CreateSequenceResponse, or AckRequested /SequenceAcknowledgement, or yet an exchange of a SecurityContextToken element.

The scenario definition introduces a description of how each one of these units of message exchanges, is carried out. This is done in form of a table that shows various dimensions or aspects of the execution of such a unit. The general layout for each instance of such a table is as follows:

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
(example: RM protocol CreateSequence/ CreateSequenceResponse)	Addressing and correlation	<p>The following are examples of addressing information whose values may be called out or be specified for specific legs of an exchange.</p> <ul style="list-style-type: none"> • wsa:ReplyTo • wsa:RelatesTo • wsa:To • wsa:Action
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> • Underlying MEP being used and how (HTTP) • Any reliance on connection establishment (e.g. MakeConnection)

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
	Piggybacking	(patterns allowed by the scenario)
	Security	(may be relevant or not depending on the scenario)
	Error handling	(content details and addressing aspects)

177

178

179 4 Reliable One-way (ROW)

180

181 4.1 Description

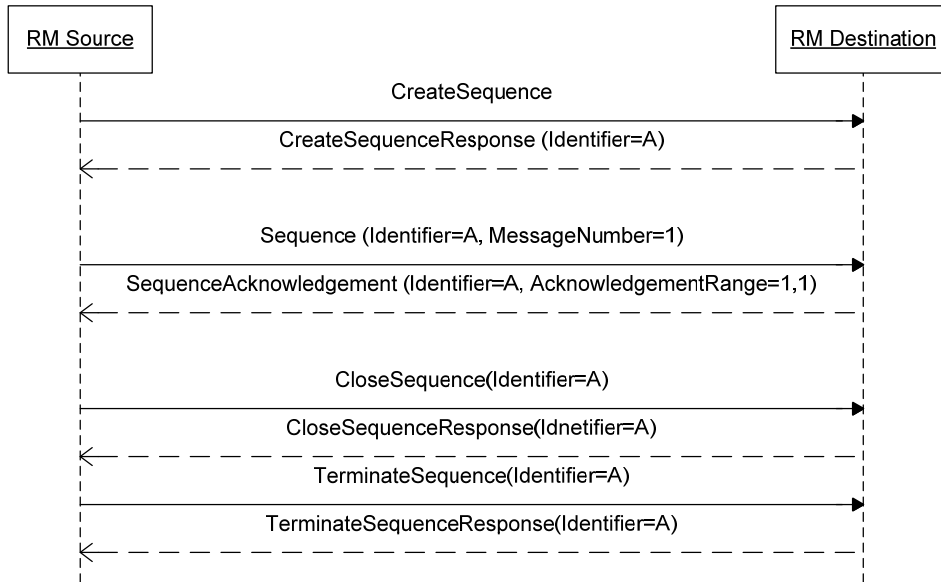
182 Scenario summary: Reliable One-way Exchange, where the client endpoint is
183 addressable. The initiator (requestor) is called the Client, the other endpoint
184 the Service.

185 Use Case: The most common use case is of a client that initiates a request to a
186 service for which no response is expected. The message is sent reliably. The
187 client is addressable, and both parties decide to NOT make use of the
188 underlying protocol back channel for any response to the client. Secure
189 conversation may be used.

190 4.2 Sequence Diagram

191 The complete scenario includes the following exchanges. The following diagram
192 does not illustrate any optional underlying protocol back-channel use:

- 193 • [optional] Secure Conversation Establishment and Cancelation
- 194 • Reliable Sequence establishment (CS/CSR)
- 195 • Application reliable exchange (1 instance of One-way message)
- 196 • Acknowledgement exchanges (either after this message, or later a consolidated
197 Ack)
- 198 • [optional] Sequence Closing
- 199 • Sequence Termination



200

201

Figure 1 - Reliable One-way

202

4.3 Scenario Constraints and Assumptions

203

No addressing constraints for either client or service endpoints.

204

Assumptions:

205

- In this usage scenario the client assumes the service endpoint has a preference for issuing any responses as new requests over the underlying protocol.

206

207

208

Scenario Constraints:

209

- There are no specific constraints in this scenario. Both endpoints are addressable.

210

211

Description:

212

- If WSDL is used then there must be no out messages defined.

213

214

4.4 Message Exchanges Details

215

4.4.1 Sequence Lifecycle Messages

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Sequence establishment (CS/CSR) Sequence closing (optional) (CIS/CISR) Sequence termination (TS/TSR) 	Addressing and correlation	<ul style="list-style-type: none"> Wsa:ReplyTo : (on CS / CIS / TS) client endpoint reference Wsa:RelatesTo: (expected on CSR / CISR / TSR, relates to request) Wsa:To
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Two (HTTP) requests in opposite directions. Endpoints involved in exchange must be prepared for new HTTP connection
	Piggybacking	Not applicable. Additional SOAP headers may be present.
	Security	Message level security: Optional following guidelines from WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

216

217 **4.4.2 Sequence Traffic Messages**

218 Note that there are no differences in Sequence Traffic messages for an addressable
219 and anonymous client.

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Application Message exchange : A One-way message (as defined in terminology)	Addressing and correlation	<ul style="list-style-type: none"> wsa:To
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Underlying request (HTTP) No application message on HTTP response, though possibly SOAP envelope with a Fault.
	Piggybacking	Not Applicable.
	Security	Message level security: Optional, RM headers must follow guidelines from WS-RM sections 5 and 6 if the sequence is protected.
	Error handling	WS-Addressing rules apply in handling faults.

220 **4.4.3 Acknowledgment Messages**

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Acknowledgements driven by either (a) spontaneous new requests as determined by Ack policy, or (b) in response to AckRequested messages 	Addressing and correlation	<ul style="list-style-type: none"> wsrn:AcksTo EPR: client endpoint reference
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> For AckRequested: Underlying protocol request (HTTP) or AcksTo EPR. For Acks: Sent to AcksTo EPR per WS-RM processing rules
	Piggybacking	<ul style="list-style-type: none"> For AckRequested: can be piggybacked on application one-ways, or sent separately. For Acks: possibly over SOAP requests containing application messages sent to client endpoint.
	Security	If the sequence is protected then acknowledgements must be secured per the rules in WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

221

222 5 Reliable One-way, anonymous client (ROW-anon)

223

224 5.1 Description

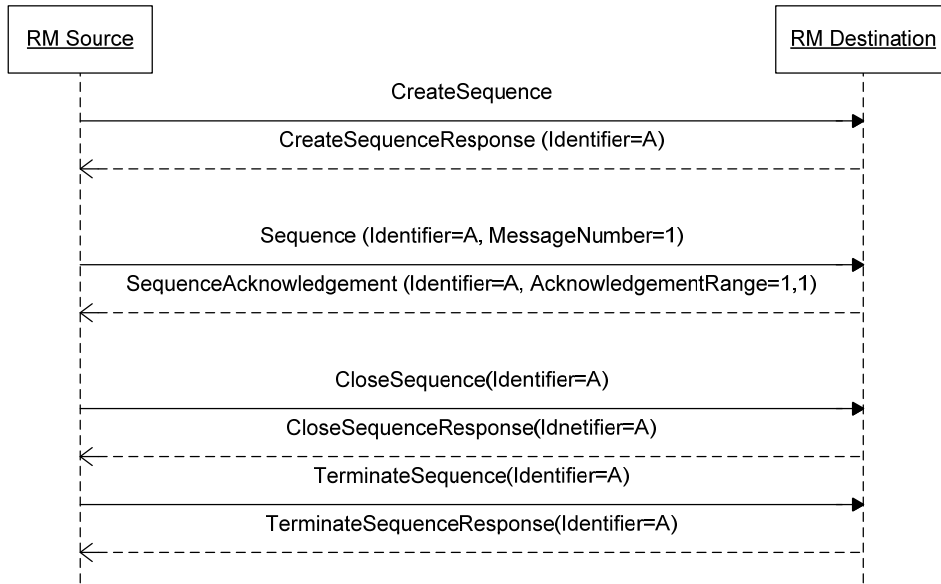
225 Scenario summary: Reliable One-way Exchange, with the use of an anonymous
226 client endpoint. The initiator (requestor) is called the Client and is anonymous,
227 the other endpoint the Service.

228 Use Case: The most common use case is of a client that initiates a request to a
229 service for which no response is expected. The message is sent reliably. The
230 client is addressable, and both parties decide to make use of the underlying
231 protocol back-channel for all responses to client. Secure conversation may be
232 used .

233 5.2 Sequence Diagram

234 The complete scenario includes the following exchanges. Every response uses
235 the underlying protocol back channel:

- 236 • [optional] Secure Conversation Establishment and Cancelation
- 237 • Reliable Sequence establishment (CS/CSR)
- 238 • Application reliable exchange (1 instance of One-way message)
- 239 • Acknowledgement exchanges (either after this message, or later a consolidated
240 Ack)
- 241 • [optional] Sequence Closing
- 242 • Sequence Termination



243

244

Figure 2 - Reliable One-way, anonymous client

245

5.3 Scenario Constraints and Assumptions

246

No addressing constraints for either client or service endpoints.

247

Assumptions:

248

- In this usage scenario, client assumes the service endpoint has a preference for not issuing requests back to it and will use the back channel for all its responses.

249

250

251

Scenario Constraints:

252

- There are no specific constraints in this scenario.

253

Description:

254

- If WSDL is used then there must be no out messages defined.

255

5.4 Message Exchanges Details

256

5.4.1 Sequence Lifecycle Messages

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Sequence establishment (CS/CSR) Sequence closing (optional) (CIS/CISR) Sequence termination (TS/TSR) 	Addressing and correlation	<ul style="list-style-type: none"> [optional] Wsa:ReplyTo : (on CS / CIS / TS) anonymous wsa:RelatesTo: (expected on CSR / CISR / TSR, relates to request) wsa:To wsa:Action
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Single (HTTP) request-reply MEP
	Piggybacking	Not applicable. Additional SOAP headers may be present.
	Security	Message level security: Optional following guidelines from WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

257 **5.4.2 Sequence Traffic Messages**

258 Note that there are no differences in Sequence Traffic messages for an addressable
259 and anonymous client.

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Application Message exchange : A One-way message (as defined in terminology)	Addressing and correlation	<ul style="list-style-type: none"> wsa:To
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Underlying request (HTTP) No application message on HTTP response, though possibly SOAP envelope with RM headers, or a Fault.
	Piggybacking	Not Applicable.
	Security	Message level security: Optional, RM headers must follow guidelines from WS-RM sections 5 and 6 if the sequence is protected.
	Error handling	WS-Addressing rules apply in handling faults.

260 5.4.3 Acknowledgement Messages

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Acknowledgements driven by either (a) piggybacking over responses (as determined by Ack policy not represented here), or (b) AckRequested messages, or (c) MakeConnection messages. 	Addressing and correlation	<ul style="list-style-type: none"> wsrn:AcksTo EPR: anonymous
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> For AckRequested: Underlying request (HTTP) For Acks: back-channel of underlying protocol (response to application message, or response to MakeConnection.)
	Piggybacking	<ul style="list-style-type: none"> For AckRequested: can be piggybacked on application one-ways, or sent separately. For Acks: only SOAP responses of one-ways (empty SOAP body).
	Security	If the sequence is protected then acknowledgements must be secured per the rules in WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

261

262 6 Reliable Request-Response (RRR)

263 6.1 Description

264 Scenario summary: Reliable asynchronous Two-way Exchange, NO use of
265 anonymous endpoint: both endpoints are addressable. The initiator (requestor)
266 is called the Client, the other endpoint the Service.

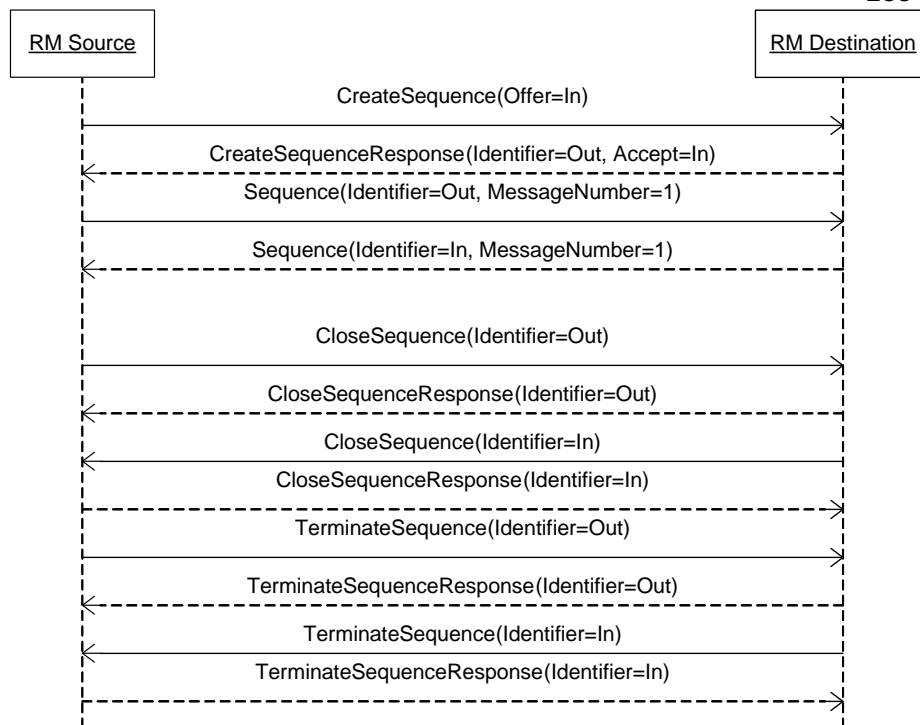
267 Use Case: A common use case is of a client that initiates a request to a service,
268 for which a response is expected on a separate connection. The request
269 message is sent reliably. The service responds with a separate service
270 invocation reliably carrying the response to the client. Both endpoints are
271 addressable, and both decide to NOT make use of the underlying protocol back-
272 channel for any response. Secure conversation may be used.

273 6.2 Sequence Diagram

274 The complete scenario includes the following exchanges. None of them uses
275 the underlying protocol back-channel:

276

- 277 • [optional] Secure Conversation Establishment and Cancelation
- 278 • Reliable Sequence establishment client-to-service (CS/CSR), with offered
279 service-to-client sequence.
- 280 • Application reliable request client-to-service
- 281 • Application reliable response service-to-client
- 282 • Acknowledgement exchange client-to-service. (not shown)
- 283 • Acknowledgement exchange service-to-client. (not shown)
- 284 • [optional] Sequence Closing client-to-service.
- 285 • [optional] Sequence Closing service-to-client.
- 286 • Sequence Termination client-to-service.
- 287 • Sequence Termination service-to-client.



289 **6.3 Scenario Constraints and Assumptions**

290 No addressing constraints for either client or service endpoints.

291 **Assumptions:**

- 292 • In this usage scenario, both client and service assume the other
- 293 endpoint has a preference for issuing any responses to their request
- 294 messages, as new requests over the underlying protocol.

295 **Scenario Constraints:**

- 296 • No specific constraints in this scenario. Both endpoints are addressable.

297 **Description:**

- 298 • When WSDL is used then there will be either request-response
- 299 operations or independent in and out messages defined. If WSDL is used
- 300 then there must be no out messages defined.

301

302 **6.4 Message Exchanges Details**

303

304 **6.4.1 Sequence Lifecycle Messages**

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Client-service Sequence establishment (CS/CSR) Client-service Sequence closing (optional) (CIS/CISR) 	Addressing and correlation	<ul style="list-style-type: none"> Wsa:ReplyTo : (on CS / CIS / TS) client endpoint reference Wsrn:Offer (on CS) Wsrn:Accept (on CSR) Wsa:RelatesTo: (expected on CSR / CISR / TSR, relates to request) Wsa:To
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Two (HTTP) requests in opposite directions. Endpoints involved in exchange must be prepared for new HTTP connection
<ul style="list-style-type: none"> Client-service Sequence termination (TS/TSR) Service-client Sequence closing (optional) (CIS/CISR) Service-client Sequence termination (TS/TSR) 	Piggybacking	Not applicable.
	Security	Message level security: Optional following guidelines from WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

305

306 **6.4.2 Sequence Traffic Messages**

307 (Only varies from table in scenario 6 by ReplyTo value.)

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Application Message request : A One-way, request, or response message	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo : client endpoint reference wsa:RelatesTo: For a response message, URI / message ID of the request.
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Underlying request (HTTP) No application message on HTTP response, though possibly SOAP envelope with a Fault.

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
	piggybacking	Not applicable.
	Security	Message level security: Optional, RM headers must follow guidelines from WS-RM sections 5 and 6 if the sequence is protected.
	Error handling	WS-Addressing rules apply in handling faults.

308 **6.4.3 Acknowledgment Messages**

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Acknowledgements from Service, driven by (a) spontaneous new requests as determined by Ack policy, or (b) in response to AckRequested messages Acknowledgements from Client, driven by either (a) spontaneous new requests as determined by Ack policy, or (b) in response to AckRequested messages 	Addressing and correlation	<ul style="list-style-type: none"> AcksTo (for sequence sent to Service): client endpoint reference, or other (NOT anonymous) AckRequested (for sequence sent to Service): sent with wsa:ReplyTo aligned with AcksTo element. AcksTo (for sequence sent to Client): service endpoint reference, or other (NOT anonymous) AckRequested (for sequence sent to Client): sent with wsa:ReplyTo aligned with AcksTo element.
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> For AckRequested: Underlying request (HTTP) For Acks: new request of underlying protocol
	Piggybacking	<ul style="list-style-type: none"> For AckRequested: can be piggybacked on application one-ways, or sent separately. For Acks: possibly over SOAP requests containing application messages sent to client endpoint.
	Security	If the sequence is protected then acknowledgements must be secured per the rules in WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

309 7 Reliable Request-Response, anonymous client (RRR-anon)

310

311 7.1 Description

312 Scenario summary: Reliable asynchronous Two-way Exchange, with one
313 anonymous endpoint (or behaving as such). The initiator (requestor) is called
314 the Client, the other endpoint the Service.

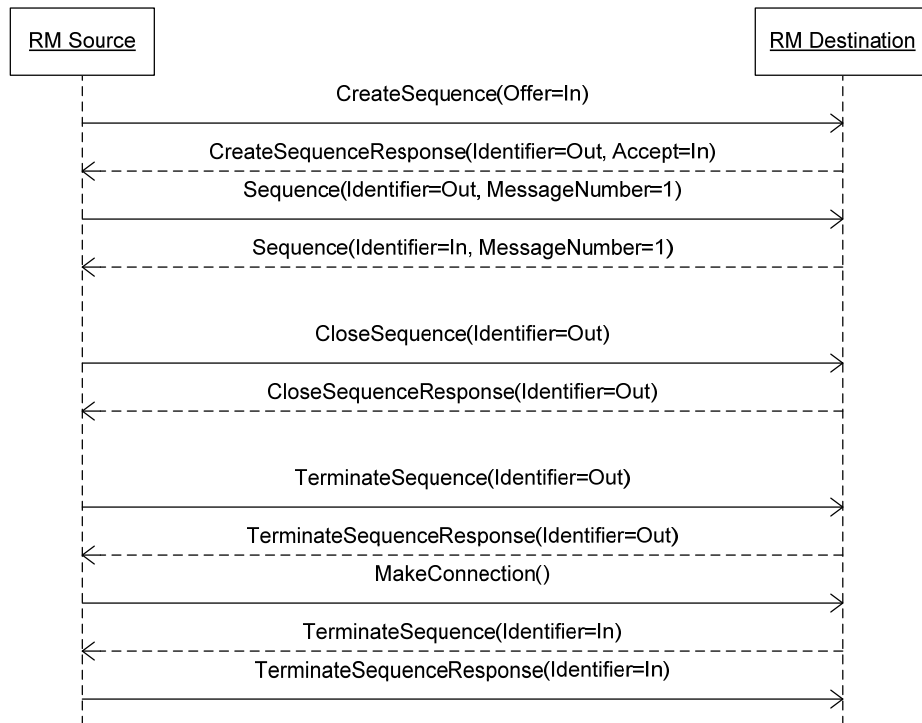
315 Use Case: A common use case is of a client that initiates a request to a service,
316 for which a response is expected on the same connection. The request message
317 is sent reliably. The Service responds reliably on the back channel which carries
318 the response to the client. Both endpoints may be addressable, but the Client
319 for some reason has connectivity issues (e.g. firewall) and cannot receive
320 incoming requests, therefore behaves as an anonymous endpoint. Any message
321 from Service to Client will need to make use of the underlying protocol back
322 channel created by a previous request. Secure conversation may be used.

323

324 7.2 Sequence Diagram

325 The complete scenario includes the following exchanges. All communication
326 must be initiated by the Client. All of the messages sent from the Client to the
327 service are over new connections. All of the messages sent from the Service to
328 Client use the underlying protocol back-channel of a previous request.

- 329 • [optional] Secure Conversation Establishment and Cancelation
- 330 • Reliable Sequence establishment client-to-service (CS/CSR), with offered
331 service-to-client sequence (accepted if reliable responses).
- 332 • Application reliable request client-to-service (1 instance of One-way message)
- 333 • Application reliable response service-to-client (as response in 1 instance of
334 Synchronous request-response exchange, or as response to MakeConnection)
- 335 • Acknowledgement exchange client-to-service.
- 336 • Acknowledgement exchange service-to-client (using back-channel).
- 337 • [optional] Sequence Closing client-to-service.
- 338 • [optional] Sequence Closing service-to-client (using back-channel).
- 339 • Sequence Termination client-to-service.
- 340 • Sequence Termination service-to-client (using back-channel).



341

342 **7.3 Scenario Constraints and Assumptions**

343 No addressing constraints for either client or service endpoints.

344 **Assumptions:**

- 345 • In this usage scenario, the client only is behaving as non-addressable.
- 346 All transfers from Service to Client use the back-channel of underlying
- 347 protocol.

348 **Scenario Constraints:**

- 349 • Both endpoints may be addressable, but the Client may have
- 350 connectivity issues that makes it behave as non-addressable.

351 **Description:**

- 352 • When WSDL is used then there will be either request-response
- 353 operations or independent in and out messages defined. If WSDL is used
- 354 then there must be no out messages defined.

355

356 **7.4 Message Exchanges Details**

357

358 **7.4.1 Sequence Lifecycle Messages**

359 The difference from the RRR usage scenario is that the Client's ReplyTo is anonymous.

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Client-service Sequence establishment (CS/CSR) Client-service Sequence closing (optional) (CIS/CISR) 	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo (from Client): (on CS / CIS / TS) anonymous wsrn:Offer (on CS from Client) wsrn:Accept (on CSR to Client) wsa:RelatesTo: (expected on CSR / CISR / TSR, relates to their request messages) wsa:To
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> For Client-service exchanges: a single (HTTP) request-response. For Service-client exchanges: the CIS / TS message is over an HTTP response, back-channel offered by MakeConnection. The CISR / TSR message is over an HTTP request.
<ul style="list-style-type: none"> Client-service Sequence termination (TS/TSR) Service-client Sequence closing (optional) (CIS/CISR) Service-client Sequence termination (TS/TSR) 	Piggybacking	Not applicable.
	Security	Message level security: Optional following guidelines from WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

360

361 **7.4.2 Sequence Traffic Messages**

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Application Message request : A One-way message or a response of a	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo (in Client request) : anonymous wsa:RelatesTo: For a response message, URI / message ID of the request.

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Synchronous request-response exchange from Client (unrelated to the initial request), or as response to MakeConnection	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Underlying request (HTTP) No application message on HTTP response to the Request, though possibly SOAP envelope with a Fault. Service to client messages over an HTTP response, back-channel offered by MakeConnection (or in case of variant, reuse of back-channel of any other subsequent request)
	Piggybacking	Possible piggybacking of RM headers or other headers on this message.
	Security	Message level security: Optional, RM headers must follow guidelines from WS-RM sections 5 and 6 if the sequence is protected.
	Error handling	WS-Addressing rules apply in handling faults.

362

363 7.4.3 Acknowledgment Messages

364 The difference from the RRR Usage scenario is that the Client's Acksto EPR is
365 anonymous.

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Manifestation / Control
<ul style="list-style-type: none"> Acknowledgements from Service, driven by (a) piggybacking over responses (as determined by Ack policy not represented here), or (b) in response to AckRequested messages, or (c) in response to 	Addressing and correlation	<ul style="list-style-type: none"> Acksto (for sequence sent to Service): anonymous Acksto (for sequence sent to Client): service endpoint reference, or other (NOT anonymous)
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> For AckRequested (from Client): Underlying request (HTTP) For Acks (from Service): response of underlying protocol (HTTP) For AckRequested (from Service): Underlying response (HTTP). For Acks (from Client): new request of underlying protocol (HTTP)

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Manifestation / Control
<p>MakeConnection message.</p> <ul style="list-style-type: none"> Acknowledgements from Client, driven by either (a) spontaneous new requests as determined by Ack policy, or (b) in new request as response to AckRequested messages 	Piggybacking	<ul style="list-style-type: none"> For AckRequested or Acks from Client: can be piggybacked on application one-ways. For AckRequested or Acks from Service: can be piggybacked on application responses.
	Security	If the sequence is protected then acknowledgements must be secured per the rules in WS-RM sections 5 and 6.
	Error handling	WS-Addressing rules apply in handling faults.

366

367

368

369

8 MakeConnection Protocol

370

371

372

373

374

Every scenario in this document that includes an Endpoint Reference as part of the message exchange may use the MakeConnection Anonymous URI as the [address] property of that EPR. The use of the MakeConnection Protocol to establish a transport-specific back-channel to allow a message targeted to one of these EPRs to be sent will be done according to the following sub-scenario.

375

8.1 *Use of the MC Anonymous URI*

376

377

378

379

380

381

382

383

An endpoint wishing to use the MakeConnection protocol to receive messages from another endpoint first needs to provide the other endpoint (the endpoint sending messages) with an EPR that includes the MC anonymous URI. This is no different than how any other EPR is provided. For example, in a traditional request-response message exchange, the `wsa:ReplyTo` EPR is used to specify the destination EPR for responses. The client indicates its intention to use the MakeConnection protocol for the delivery of those responses by using the MC anonymous URI in the [address] property of the `wsa:ReplyTo` EPR.

384

385

386

387

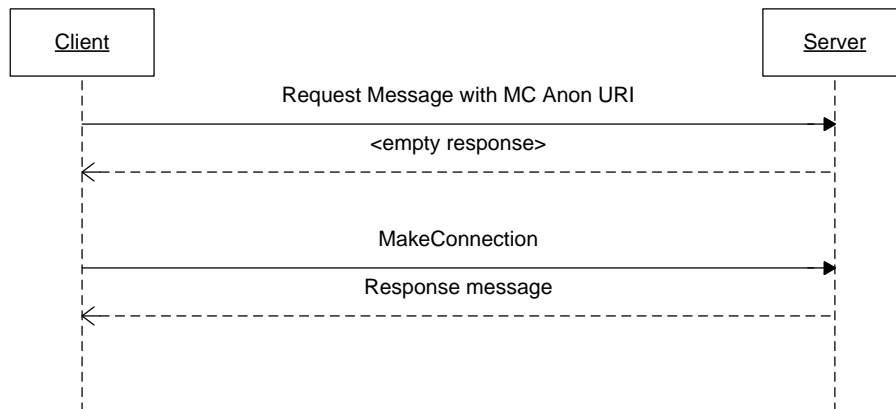
388

389

Once the service receives the request, it can send the response on the back-channel of the original connection. If, however, the service chooses to not send a response on the transport-specific back-channel of the request message then the client uses the MakeConnection message to create a new connection to establish a new back-channel. The service can then use this new back-channel to send the expected response.

390

The overall flow would be:



391

392

393

9 Reliable, Secure Conversation Establishment and Cancellation

394
395
396
397

Scenarios section 4 through section 7 in this document may include additional exchanges for establishing and canceling a secure conversation. The establishment and cancellation of secure conversations will be done according to the following sub-scenarios.

398

9.1 RequestSecurityToken, CreateSequence (RST-CS)

399
400
401

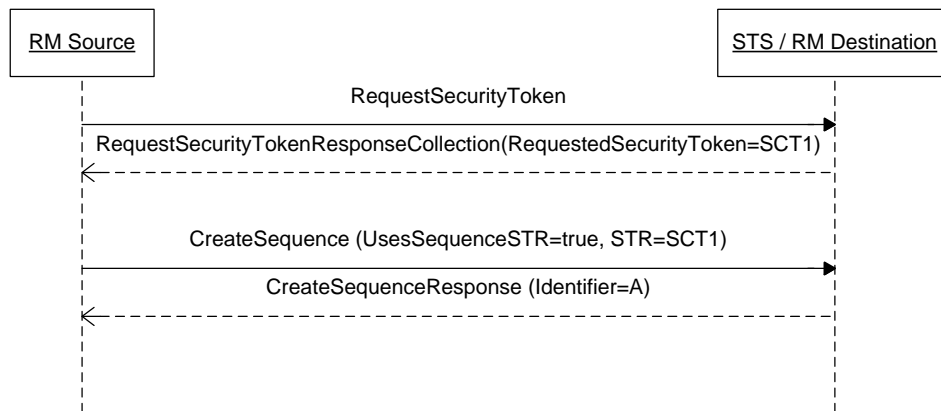
A reliable sequence is assumed to be transferred from start to end within a single secure conversation. The conversation is started with the intent of securing this sequence. The conversation may include more than one sequence.

402

This sub-scenario assumes that the STS / RM Destination is addressable.

403
404
405

Client sends RST (RequestSecurityToken) to the Service endpoint's STS to establish SecurityContextToken. Service endpoint responds with RSTR and new SecurityContextToken.



406

407

Figure 3 - SCT Establishment

408

409

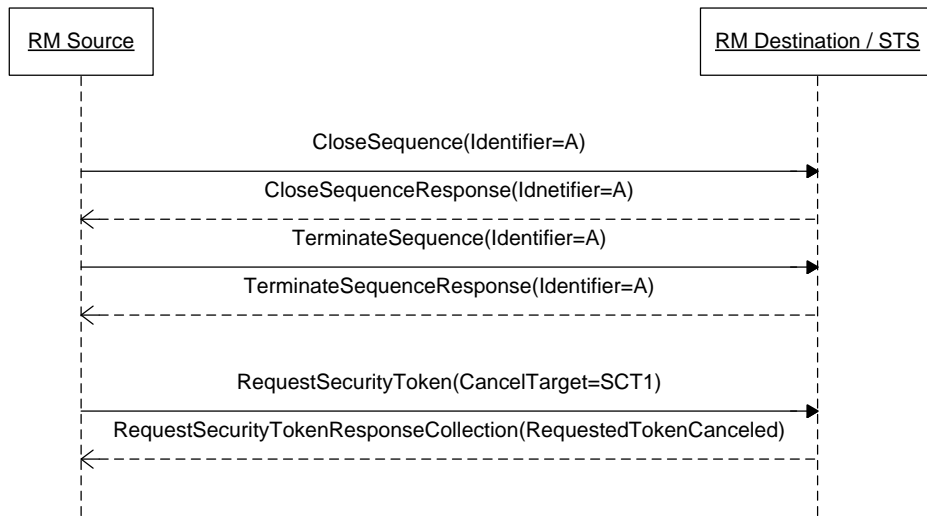
9.2 TerminateSequence, Cancel (TS-Cancel)

410
411
412

In this sub-scenario, the secure conversation was established for an RM sequence. This sub-scenario assumes that the STS / RM Destination is addressable.

413
414
415
416
417

The secure conversation that includes a reliable sequence will be cancelled after the sequence is terminated. Client sends RST (RequestSecurityToken) with a CancelTarget element identifying the SecurityContextToken of the conversation to be terminated. Service endpoint responds with RSTRC confirming the cancellation.



418

419

Figure 4 - SCT Cancellation

420 10 Secure Request-Response (SRR)

421

422 10.1 Description

423 Scenario summary: Secure Two-way Exchange, with NO use of anonymous
424 endpoint: both endpoints are addressable. The initiator (requestor) is called
425 the Client, the other endpoint the Service.

426 Use Case: A common use case is of a client that initiates series of requests to a
427 service, for which a series responses are expected on a separate connection.
428 The request message is sent securely. The Service responds securely on the
429 separate service invocation which carries the response to the client. Both
430 endpoints are addressable and both decide to NOT make use of the underlying
431 protocol back channel created by a previous request.

432

433 Since the client will be sending a series of secure requests to the service,
434 secure conversation is required for performance reasons since it uses less
435 expensive symmetric key operations and improves security, by reducing the
436 exposure of the long term secret

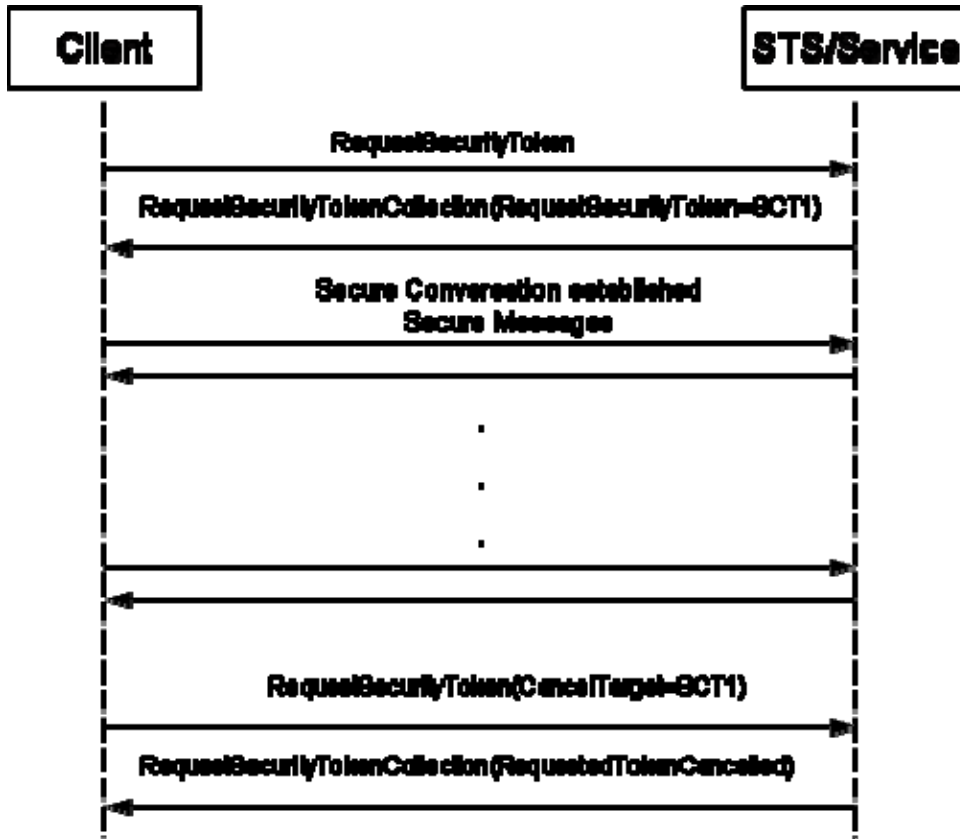
437

438

439 10.2 Sequence Diagram

440 The complete scenario includes the following exchanges. All of the messages
441 sent from the Client to the service are over new connections. All of the
442 messages sent from the Service to Client are over new connections.

- 443 • Client issues a Request Security Token
- 444 • Secure Token Service issues Request Security Token Response
- 445 • Secure Conversation Establishment and Cancellation



446
447

448 **10.3 Scenario Constraints and Assumptions**

449 No addressing constraints for either client or service endpoints.

450 Assumptions:

- 451 • In this usage scenario, both client and service assume the other
- 452 endpoint has a preference for issuing any responses to their request
- 453 messages, as new requests over the underlying protocol.

454

- 455 • Client successfully obtains an SCT from the STS

456

457 Scenario Constraints:

- 458 • No specific constraints in this scenario. Both endpoints are addressable.

459

460 **10.4 Message Exchanges Details**

461

462 **10.4.1** *Secure Conversation Lifecycle Messages*

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Client-service Secure Conversation establishment (RST/RSTR) Client-service Secure Conversation closing (RST-CancelTarget/RSTRC) 	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo (from Client): (on RST / RSTR) client endpoint reference Wsa:RelatesTo: (expected on RST / RSTR , relates to request) Wsa:To
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Two (HTTP) requests in opposite directions. Endpoints involved in exchange must be prepared for new HTTP connection
	Security	Message level security: Secure Conversation
	Error handling	WS-Addressing rules apply in handling faults.

463

464 **10.4.2** *Application Traffic Messages*

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Application Message request : A One-way message or response message	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo : client endpoint reference wsa:RelatesTo: For a response message, URI / message ID of the request.
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Underlying request (HTTP) No application message on HTTP response, though possibly SOAP envelope with a Fault.
	Security	Message level security: Secure Conversation
	Error handling	WS-Addressing rules apply in handling faults.

465

466

11 Secure Request-Response, anonymous client (SRR-anon)

467

468

469 11.1 Description

470 **Scenario summary:** Secure Two-way Exchange, with one anonymous endpoint
471 (or behaving as such). The initiator (requestor) is called the Client, the other
472 endpoint the Service.

473 **Use Case:** A common use case is of a client that initiates series of requests to a
474 service, for which a series responses are expected on the same connection. The
475 request message is sent securely. The Service responds securely on the back
476 channel which carries the response to the client. Both endpoints may be
477 addressable, but the Client for some reason has connectivity issues (e.g.
478 firewall) and cannot receive incoming requests, therefore behaves as an
479 anonymous endpoint. Any message from Service to Client will need to make
480 use of the underlying protocol back channel created by a previous request.

481

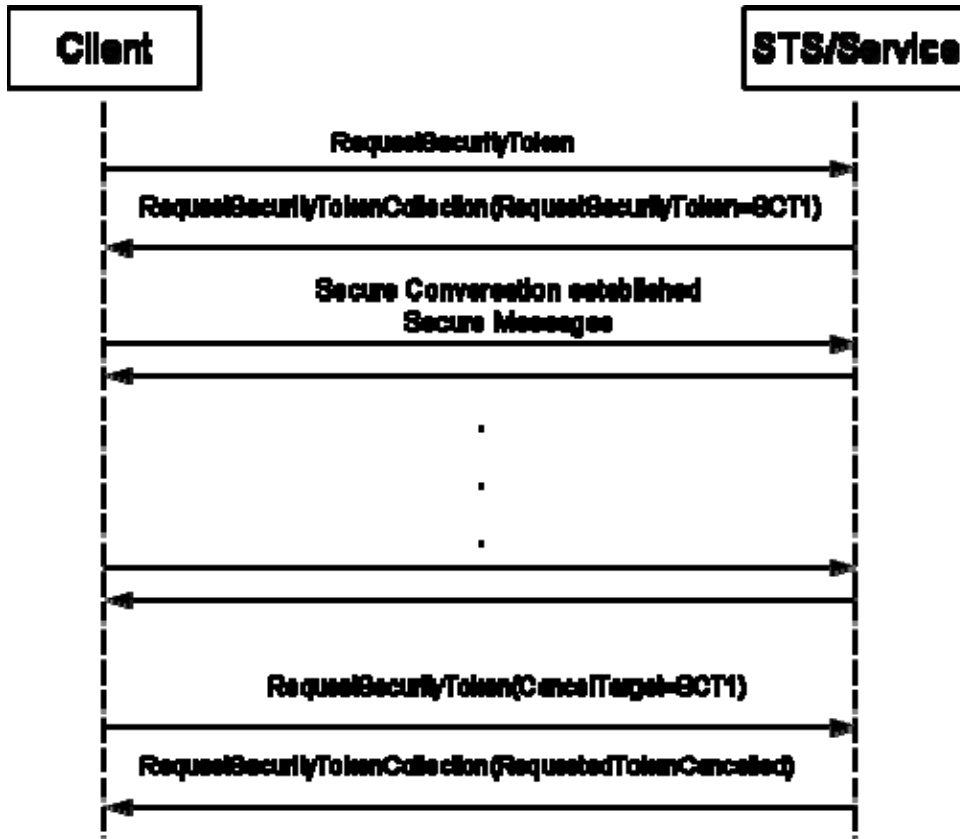
482 Since the client will be sending a series of secure requests to the service,
483 secure conversation is required for performance reasons since it uses less
484 expensive symmetric key operations and improves security, by reducing the
485 exposure of the long term secret
486

487 11.2 Sequence Diagram

488 The complete scenario includes the following exchanges. All communication
489 must be initiated by the Client. All of the messages sent from the Client to the
490 service are over new connections. All of the messages sent from the Service to
491 Client use the underlying protocol back-channel of a previous request.

492

- 493 • Client issues a Request Security Token
- 494 • Secure Token Service issues Request Security Token Response
- 495 • Secure Conversation Establishment and Cancellation



496
497
498

499 **11.3 Scenario Constraints and Assumptions**

500 No addressing constraints for either client or service endpoints.

501 Assumptions:

- 502 • In this usage scenario, the client only is behaving as non-addressable.
- 503 All transfers from Service to Client use the back-channel of underlying
- 504 protocol.
- 505 • Client successfully obtains an SCT from the STS

506

507 Scenario Constraints:

- 508 • Both endpoints may be addressable, but the Client may have
- 509 connectivity issues that make it behave as non-addressable.

510

511 **11.4 Message Exchanges Details**

512

513 **11.4.1 *Secure Conversation Lifecycle Messages***

514 The difference from the RRR usage scenario is that the Client's ReplyTo is anonymous.

Scenario Message Exchange Unit(s)	Aspects of the Message Exchange Unit	Message Details
<ul style="list-style-type: none"> Client-service Secure Conversation establishment (RST/RSTR) Client-service Secure Conversation closing (RST-CancelTarget/RSTRC) 	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo (from Client): (on RST / RSTR) anonymous
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> For Client-service exchanges: a single (HTTP) request-response. For Service-client exchanges: the RSTR message is over an HTTP response, back-channel offered by MakeConnection if the original connection does not contain the response. Thus, a new back channel must be created. The RSTR message is over an HTTP response.
	Security	Message level security: Secure Conversation
	Error handling	WS-Addressing rules apply in handling faults.

515

516 **11.4.2 *Application Traffic Messages***

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
Application Message request : A One-way message or a response of a Synchronous request-response exchange from Client (unrelated to the initial request), or as	Addressing and correlation	<ul style="list-style-type: none"> wsa:ReplyTo (in Client request) : anonymous wsa:RelatesTo: For a response message, URI / message ID of the request.
	Underlying protocol binding and connection establishment	<ul style="list-style-type: none"> Underlying request (HTTP) No application message on HTTP response to the Request, though possibly SOAP envelope with a Fault. Service to client messages over an HTTP response, back-channel

Scenario Message Exchange Unit	Aspect of the Message Exchange Unit	Message Details
response to MakeConnection	Security	Message level security: Secure Conversation
	Error handling	WS-Addressing rules apply in handling faults.

517

518

12 Revision History

Rev	Date	By Whom	What
0.1	2006-09-1	Jacques Durand	Initial draft.
0.2	2006-09-12	Marc Goodner and Jacques Durand	Review / various edits,
0.3	2006-09-28	Jacques Durand	Updated scenario 1 (-> ROW-anon), added ROW-addressed, RA2W-addressed, RS2W-all.
0.4	2006-09-30	Marc Goodner & Jacques Durand	Added flow diagrams, for SecureConversation exchanges and for ROW scenario.
0.5	2006-10-13	Jacques Durand	Various edits, Added RA2W-1anon scenario after discussion with Marc.
0.6	2006-10-30	Marc Goodner	Edits from Plenary discussion.
0.7	2007-03-28	Doug Daivs	Added MC scenario
0.8	2007-04-6	Charles Le Vay	Added SC stand-alone scenarios
0.9	2008-09-26	Ram Jeyaraman	Fixed the arrow lines (responses over backchannel are shown with dotted lines) in the diagrams. Some editorial corrections.
1.0	2008-10-05	Ram Jeyaraman	Added MC to list of specs in Section 1.4. Modified the definition of "Anonymous client" and "Addressable request-response message exchange".