



Secure Message Delivery

Technical Overview

Version 1.0 Draft — 28 September 2009

For review and comment

National E-Health Transition Authority Ltd

Level 25

56 Pitt Street

Sydney, NSW, 2000

Australia.

www.nehta.gov.au**Disclaimer**

NEHTA makes the information and other material ('Information') in this document available in good faith but without any representation or warranty as to its accuracy or completeness. NEHTA cannot accept any responsibility for the consequences of any use of the Information. As the Information is of a general nature only, it is up to any person using or relying on the Information to ensure that it is accurate, complete and suitable for the circumstances of its use.

Document Control

This document is maintained in electronic form. The current revision of this document is located on the NEHTA Web site and is uncontrolled in printed form. It is the responsibility of the user to verify that this copy is of the latest revision.

Copyright © 2009, NEHTA.

This document contains information which is protected by copyright. All Rights Reserved. No part of this work may be reproduced or used in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems—without the permission of NEHTA. All copies of this document must include the copyright and other information contained on this page.

Document Information

Change History

Version	Date	Comments
0.1	2009-02-04	Initial draft based on pathology package tech arch
0.2	2009-03-03	Updated for consistency with other documents
0.3	2009-03-27	Applied new name for spec "Clinical Document Delivery"
1.0 Draft	2009-03-31	Minor consistency updates. Draft release for community.
1.0 Draft	2009-09-28	Significant update for consistency with SMD specification from PIP working group. Renamed CDD -> SMD.

Table of Contents

Document Information	iii
Change History	iii
Table of Contents	iv
Preface	vii
Document Purpose	vii
Scope	vii
Intended Audience.....	vii
Document status	vii
Document Map.....	viii
References and Related Documents	viii
Definitions, Acronyms and Abbreviations.....	viii
1 Service Interfaces	1
1.1 Sealed Message Delivery	1
1.1.1 Purpose	1
1.1.2 Operations	1
1.2 Sealed Message Retrieval	1
1.2.1 Purpose	1
1.2.2 Operations	1
1.3 Sealed Transport Response Delivery	2
1.3.1 Purpose	2
1.3.2 Operations	2
1.4 Sealed Transport Response Retrieval	2
1.4.1 Purpose	2
1.4.2 Operations	2
1.5 Sealed Immediate Message Delivery.....	2
1.5.1 Purpose	2
1.5.2 Operations	3
2 Technical Scenarios	4
2.1 Service Categories	4
2.2 Sealed Message Delivery Scenarios	5
2.2.1 Direct Delivery.....	5
2.2.2 Indirect Delivery via Sender Intermediary	6
2.2.3 Indirect with Message Hosting.....	7
2.2.4 Indirect with Message and Response Hosting.....	8
2.2.5 Indirect via Sender Intermediary with Message and Response Hosting.....	9
2.2.6 Intermediate Transport Responses.....	11
2.2.7 Intermediate Transport Responses with Intermediary Skipped	12
2.3 Sealed Immediate Message Delivery Scenarios	13
2.3.1 Direct Invocation.....	14
2.3.2 Invocation via Sender Intermediary	14
2.3.3 Invocation via Sender and Receiver Intermediary	15
2.4 Retry Scenarios	16
2.4.1 SOAP Request Failure	17
2.4.2 SOAP Response Failure	17
Definitions	19
Shortened Terms.....	19
References	20
Specification Documents.....	20
References	20

This page is intentionally left blank.

Preface

Document Purpose

This document lays out the technical design and operation of the *Secure Message Delivery* specification through identifying interfaces and describing usage scenarios for those interfaces using UML sequence diagrams. It is intended to assist readers in understanding the architecture and technical use cases associated with the detailed specifications in Secure Message Delivery – Endpoint Specifications [SMD-ES]. This document is non-normative.

Scope

The breadth of scope of this document is limited by the Secure Message Delivery Specification as defined in the Secure Message Delivery Overview [SMD-O].

The depth of scope is restricted to high level design and concepts for the solution of Secure Message Delivery.

Intended Audience

This document is intended for the following audiences:

- Solution Architects
- Solution Developers
- Technical Decision makers.

Document status

This document is a draft and has been released for comment and feedback purposes.

Document Map

This diagram represents the relationship between this document and other related specifications within the Secure Message Delivery Specification.

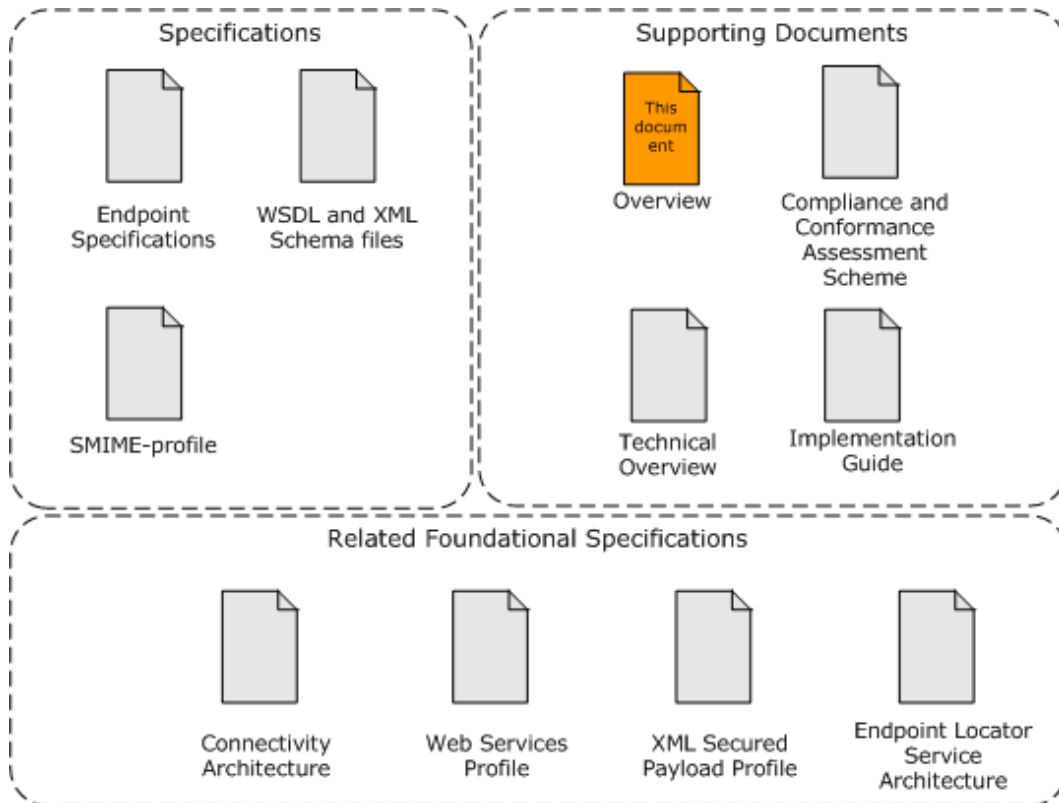


Figure 1: Document Map

References and Related Documents

For a list of all referenced documents, see the [References](#) at the end of the document, on page 20.

Definitions, Acronyms and Abbreviations

For a lists of abbreviations, acronyms and abbreviations, see the [Definitions section](#) at the end of the document, on page 19

1 Service Interfaces

This chapter describes the web services that are defined for secure message delivery (SMD). These services are initially described in isolation, so this chapter should be read in conjunction with chapter 2 “Technical Scenarios” which puts these services together into different scenarios that achieve the goals of secure message delivery.

The Secure Message Delivery specification defines five service interfaces:

- Sealed Message Delivery;
- Sealed Message Retrieval;
- Sealed Transport Response Delivery;
- Sealed Transport Response Retrieval; and
- Sealed Immediate Message Delivery.

This chapter only contains a high level description of these service interfaces. Full details of these technical services can be found in the *Secure Message Delivery: Endpoint Specifications* [SMD-ES].

These service interfaces have a one-to-one correspondence with the WSDL interfaces defined in the *Secure Message Delivery — Service Interface Specification WSDL and XML Schema files v1.0* [SMD-WX].

1.1 Sealed Message Delivery

1.1.1 Purpose

This service interface allows a Receiver to receive a message either directly or through an Intermediary. This service is provided by the Receiver, a Receiver Intermediary or a Sender Intermediary.

1.1.2 Operations

The following operation is offered by this service interface:

- `deliver`
This operation allows the service provider to receive messages.

1.2 Sealed Message Retrieval

1.2.1 Purpose

This service interface allows a Receiver to retrieve messages from a Receiver Intermediary when the Receiver uses such an intermediary. This service is provided by the Receiver Intermediary.

1.2.2 Operations

The following operations are offered by this service interface:

- `list`
This operation allows a Receiver to query a Receiver Intermediary for a list of documents that are available for the Receiver.
- `retrieve`
This operation allows a Receiver to retrieve messages from a Receiver Intermediary. The Receiver supplies the invocation identifiers for

messages to be returned, or if no identifiers are provided, the Receiver Intermediary will select the messages to be returned. The invocation identifiers for messages can be obtained from the Receiver Intermediary via the `list` operation

1.3 Sealed Transport Response Delivery

1.3.1 Purpose

This service interface allows a Sender to receive transport responses for messages sent to a Receiver. Transport responses are used to indicate delivery status. This service is provided by the Sender, Sender Intermediary or Receiver Intermediary.

Two types of transport response are distinguished in the specification:

1. A *final* transport response, which indicates that a message has either been successfully delivered or could not be delivered.
2. *Intermediate* transport responses, which are used to provide additional status information relating to message delivery.

Intermediate transport responses are optional for all roles, and are only permitted by mutual consent between communicating parties.

1.3.2 Operations

The following operation is offered by this technical service:

- `deliver`
This operation allows the service provider to receive transport responses. The operation can carry one or more transport responses.

1.4 Sealed Transport Response Retrieval

1.4.1 Purpose

This service interface allows transport responses to be retrieved from an Intermediary when the Sender uses the Intermediary option.

1.4.2 Operations

The following operations are offered by this technical service:

- `retrieve`
This operation allows a Sender to retrieve a set of transport responses from an Intermediary.
- `remove`
This operation allows a Sender to indicate that a set of transport responses have been successfully retrieved and no longer need to be made available for retrieval.

1.5 Sealed Immediate Message Delivery

1.5.1 Purpose

This service interface allows a Sender to send a message and receive an immediate application response as the return value of the invocation. Its primary purpose is to support request/response interactions, for example,

queries, but might also be used for more general message delivery in some situations.

This service is provided by the Receiver, a Receiver Intermediary or a Sender Intermediary.

1.5.2 Operations

The following operation is offered by this service interface:

- `deliver`

This operation allows the invoker to deliver a message and receive an application response in the return value of the invocation.

2 Technical Scenarios

This chapter documents a set of Secure Message Delivery scenarios corresponding to the high level processes defined in the Secure Message Delivery - Overview [SMD-O]. As discussed in this document, there are two modes of interaction:

1. *Deferred mode*, which provides one-way messaging, often delivered in a store and forward fashion with no expectation of an immediate or synchronous response.
2. *Immediate mode*, which provides two-way messaging with an expectation of an immediate and synchronous response.

These scenarios describe how the service interfaces identified in chapter 1 can be used to achieve the objectives of Secure Message Delivery, that is:

- delivering messages;
- confirming message delivery and status through transport responses (deferred mode); or
- receiving an immediate application response (immediate mode).

Senders need to decide whether they send messages and receive transport or application responses directly or through a Sender Intermediary. Receivers need to decide whether they receive messages and send transport or application responses directly or through a Receiver Intermediary. Each needs to determine their requirements and weigh up the features and tradeoffs.

Section 2.1 briefly introduces service categories, which identify the business nature of services. Section 2.2 describes a set of deferred mode message delivery scenarios. Section **Error! Reference source not found.** describes a set of immediate mode delivery scenarios. Section 2.4 describes retry scenarios resulting from communication or other failure. Note that these scenarios are not exhaustive but cover most common interactions and configurations.

2.1 Service Categories

Service categories are used to identify the business nature of each invocation. Service categories are published for each service instance in a service directory. The use of service directories is described further in [SMD-SD].

For the retrieval interfaces and the Transport Response Delivery interface, fixed service categories are defined in the endpoint specifications [SMD-ES].

For Sealed Message Delivery and Sealed Immediate Message Delivery interfaces, the service category should reflect the business or clinical nature of the opaque payload. Thus, the service categories for these interfaces depend on the applications supported by the messaging (e.g. pathology result reporting, referral). A set of standard service categories for clinical payloads is being defined separately from the Secure Message Delivery specification.

2.2 Sealed Message Delivery Scenarios

The sequence diagrams in the following subsections describe a set of interaction scenarios that might arise in the usage of Secure Message Delivery in deferred mode. Each diagram identifies the roles involved in the scenarios and shows the sequence of interface invocations in the scenario using UML sequence diagrams. In the diagrams that follow, the following abbreviations are used as operation name prefixes and refer to the interfaces identified in chapter 1 as follows:

SMD - Secure Message Delivery

SMR - Secure Message Retrieval

TRD - Transport Response Delivery

TRR - Transport Response Retrieval

As noted previously, these scenarios are not exhaustive but illustrate common interaction patterns.

Note that the *final* and *intermediate* parameter values in the TRD *deliver* invocations are used to distinguish final and intermediate transport responses, as discussed in section 1.3.

2.2.1 Direct Delivery

2.2.1.1 Process

1. The Sender invokes the *deliver* operation on the Receiver SMD interface to deliver a message directly to the receiver.
2. The Receiver confirms delivery at some point afterwards by invoking the *deliver* operation on the Sender TRD interface.

The process is illustrated in the sequence diagram of Figure 2

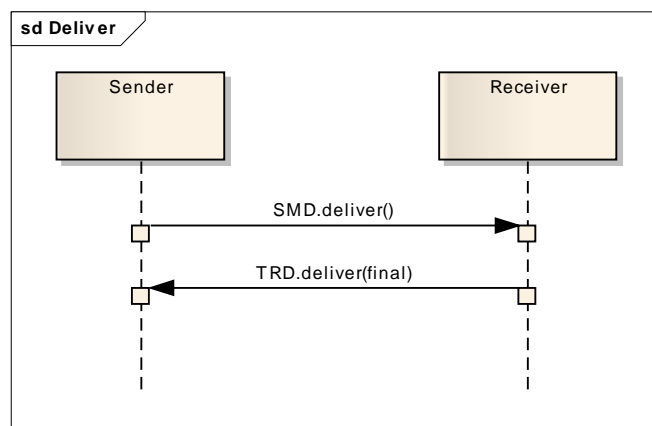


Figure 2: Direct Delivery

2.2.1.2 Features

- The message is delivered directly with no delays incurred through store/retrieve or intermediary processing.
- The transport response is similarly delivered directly with no delays incurred through store/retrieve or intermediary processing.

2.2.1.3 Tradeoffs

- Sender and Receiver must host accessible web services, which normally implies:
 - a publically accessible server with reasonable availability;

- a registered domain name; and
- a static IP address.

For small organisations, this is not always possible due to limited IT support or network presence.

- Sender and Receiver applications must be accessible to each other through corporate firewalls, if they exist. Security policy in many larger organisations might prevent such access.

2.2.2 Indirect Delivery via Sender Intermediary

2.2.2.1 Process

1. The Sender invokes the `deliver` operation on the Sender Intermediary SMD interface to deliver a message to the receiver.
2. The Sender intermediary retransmits the message directly to the Receiver.
3. The Receiver confirms delivery at some point afterwards by invoking the `deliver` operation on the Sender Intermediary TRD interface.
4. The Sender Intermediary retransmits the message to the Sender by invoking the `deliver` operation on the Sender TRD interface.

The process is illustrated in the sequence diagram of Figure 3.

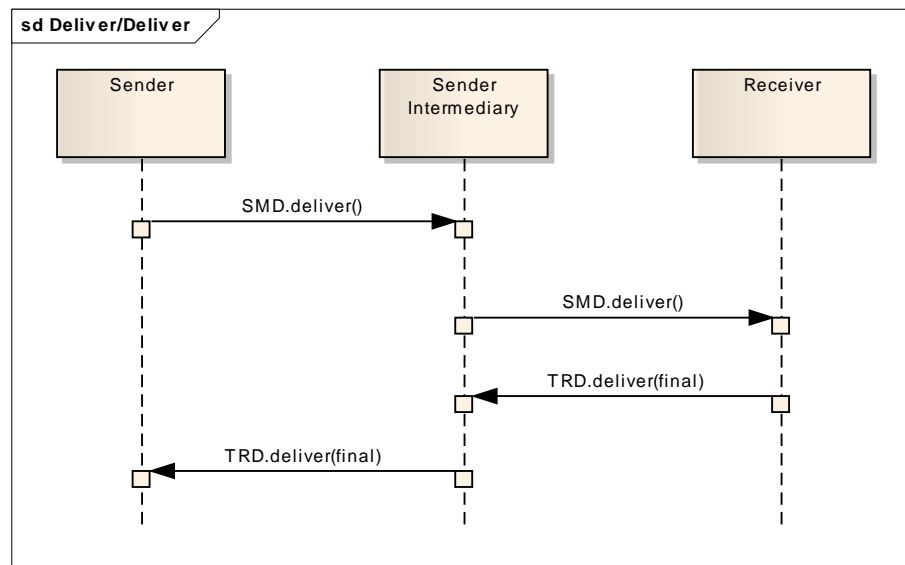


Figure 3: Indirect Delivery via Sender Intermediary

2.2.2.2 Features

- The Sender can traverse firewalls through a trusted relationship with the Sender Intermediary.
- Assuming the Sender Intermediary is highly available and always online, the Sender can "send-and-forget" without concern for the availability of the receiver.
- The Sender TRD service only needs to be accessible to the Sender Intermediary.

2.2.2.3 Tradeoffs

- The Receiver must host accessible web services, which normally implies:
 - a publically accessible server with reasonable availability;
 - a registered domain name; and

- a static IP address.

For small organisations, this is not always possible due to limited IT support or network presence.

- The Receiver application must be accessible through a corporate firewall, if it exists, and must also be able to access the Sender Intermediary through that firewall to deliver the transport response. Security policy in many larger organisations might prevent such access.
- The Sender must still host a web service, although it need not be as accessible or available as in the direct case.
- The Sender Intermediary could introduce delays in the message delivery process.
- The use of an intermediary adds cost and makes delivery status tracking more complex.

2.2.3 Indirect with Message Hosting

2.2.3.1 Process

1. The Sender invokes the `deliver` operation on the Receiver Intermediary SMD interface to deliver a message to the Receiver.
2. The Receiver Intermediary stores the message for subsequent retrieval.
3. The Receiver retrieves the message, first by listing the available messages and then by retrieving those messages using the Receiver Intermediary SMR interface.
4. The Receiver confirms delivery by invoking the `deliver` operation on the Receiver Intermediary TRD interface.
5. The Receiver Intermediary retransmits the transport response by invoking the `deliver` operation on the Sender TRD interface.

The process is illustrated in the sequence diagram of

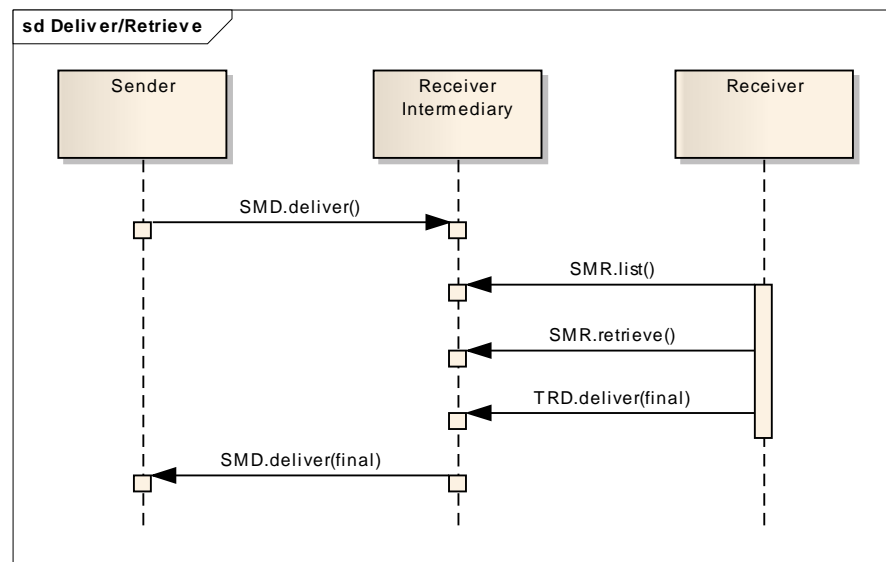


Figure 4: Indirect with Message Hosting

2.2.3.2 Features

- The Receiver does not need to operate a web service and can thus receive messages with limited network presence.
- The use of a trusted intermediary allows the Receiver to operate from behind a firewall.

- Assuming the Receiver Intermediary is highly available, the Sender has reasonable assurance that the Receiver Intermediary interface will be available when required.

2.2.3.3 Tradeoffs

- Message delivery time is governed by the frequency of retrievals initiated by the Receiver.
- The Sender must host an accessible web service, which normally implies:
 - a publically accessible server with reasonable availability;
 - a registered domain name; and
 - a static IP address.

For small organisations, this is not always possible due to limited IT support or network presence.

- The Sender application must be accessible through a corporate firewall if it exists, and must also be able to access the Receiver Intermediary through that firewall. Security policy in many larger organisations might prevent such access.
- The use of an intermediary adds cost and makes delivery status tracking more complex.

2.2.4 Indirect with Message and Response Hosting

2.2.4.1 Process

1. The Sender invokes the `deliver` operation on the Receiver Intermediary SMD interface to deliver a message to the Receiver.
2. The Receiver Intermediary stores the message for subsequent retrieval.
3. The Receiver retrieves the message, first by listing the available messages and then by retrieving those messages using the Receiver Intermediary SMR interface.
4. The Receiver confirms delivery by invoking the `deliver` operation on the Receiver Intermediary TRD interface.
5. The Receiver Intermediary retransmits the transport response by invoking the `deliver` operation on the Sender Intermediary TRD interface.
6. The Sender Intermediary stores the message for subsequent retrieval.
7. The Sender retrieves the transport response by invoking the `retrieve` operation on the Sender Intermediary TRR interface.
8. The Sender confirms retrieval of the transport response by invoking the `remove` operation on the Sender Intermediary TRR interface.

The process is illustrated in the sequence diagram of Figure 5

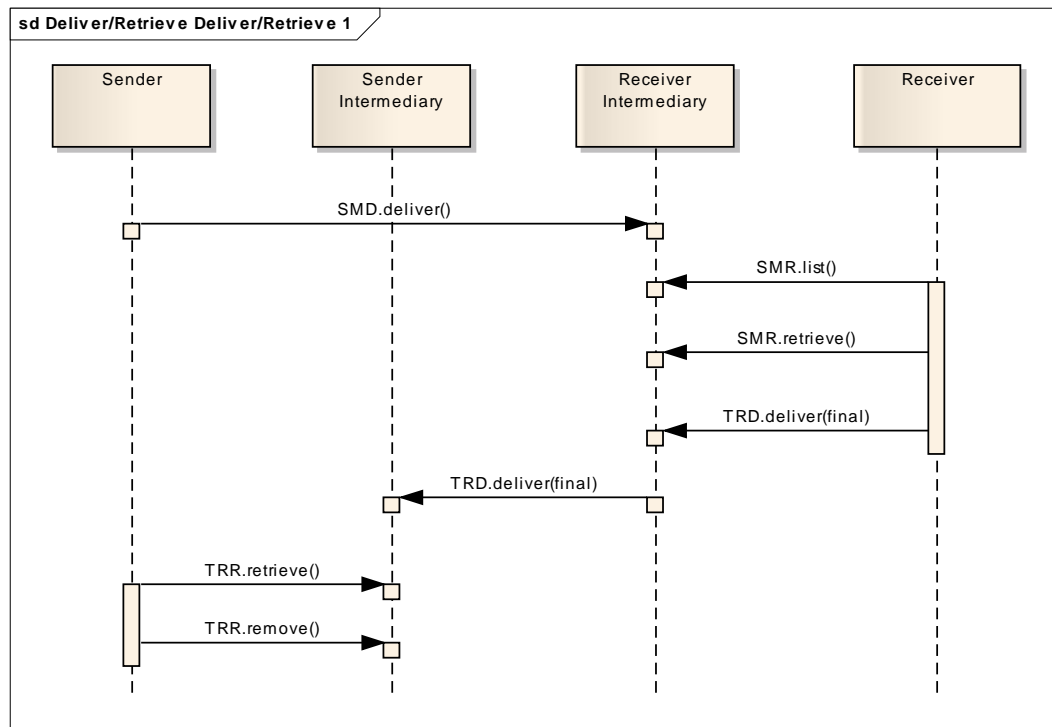


Figure 5: Indirect with Message and Response Hosting

2.2.4.2 Features

- Neither Sender nor Receiver need to operate a web service and can thus receive messages with limited network presence.
- The use of a trusted intermediary allows the Receiver to operate from behind a firewall.
- Assuming the Receiver Intermediary is highly available, the Sender has reasonable assurance that the Receiver Intermediary interface will be available when required.
- The use of a Sender Intermediary for transport response retrieval reduces firewall traversal issues for the Sender.

2.2.4.3 Tradeoffs

- Message and transport response delivery time is governed by the frequency of retrievals initiated by the Receiver and Sender respectively.
- The Sender application must be able to access the Receiver Intermediary through a corporate firewall if it exists. Security policy in many larger organisations might prevent such access.
- The use of two intermediaries adds cost and makes delivery status tracking more complex.

2.2.5 Indirect via Sender Intermediary with Message and Response Hosting

2.2.5.1 Process

1. The Sender invokes the `deliver` operation on the Sender Intermediary SMD interface to deliver a message to the Receiver.
2. The Seder Intermediary retransmits the message by invoking the `deliver` operation on the Receiver Intermediary SMD interface.
3. The Receiver Intermediary stores the message for subsequent retrieval.

4. The Receiver retrieves the message, first by listing the available messages and then by retrieving those messages using the Receiver Intermediary SMR interface.
5. The Receiver confirms delivery by invoking the `deliver` operation on the Receiver Intermediary TRD interface.
6. The Receiver Intermediary retransmits the transport response by invoking the `deliver` operation on the Sender Intermediary TRD interface.
7. The Sender Intermediary stores the message for subsequent retrieval.
8. The Sender retrieves the transport response by invoking the `retrieve` operation on the Sender Intermediary TRR interface.
9. The Sender confirms retrieval of the transport response by invoking the `remove` operation on the Sender Intermediary TRR interface.

The process is illustrated in the sequence diagram of Figure 6.

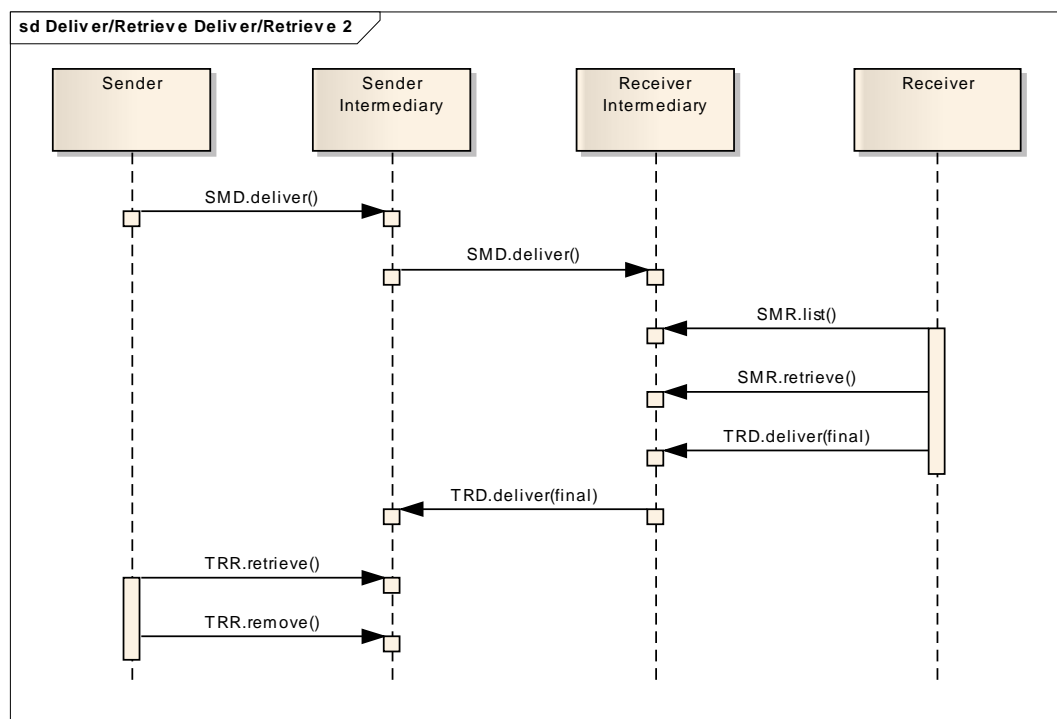


Figure 6: Indirect via Sender Intermediary with Message and Response Hosting

2.2.5.2 Features

- Neither Sender nor Receiver need to operate a web service and can thus receive messages with limited network presence.
- The use of trusted intermediaries for sending and receiving allows both the Sender and Receiver to operate from behind a firewall.
- Assuming the Intermediaries are highly available, the Sender and Receiver have reasonable assurance that the delivery interfaces will be available when required.

2.2.5.3 Tradeoffs

- Message and transport response delivery time is governed by the frequency of retrievals initiated by the Receiver and Sender respectively.
- The use of two intermediaries adds cost and makes delivery status tracking more complex.

2.2.6 Intermediate Transport Responses

2.2.6.1 Process Fragment

The sequence diagram of figure Figure 7 specifies a fragment of the delivery process involving two intermediaries, with intermediate transport responses used by the Sender Intermediary to track message status. The steps are as follows:

1. The Sender Intermediary retransmits the message by invoking the `deliver` operation on the Receiver Intermediary SMD interface.
2. The Receiver Intermediary stores the message for subsequent retrieval.
3. The Receiver retrieves the message, first by listing the available messages and then by retrieving those messages using the Receiver Intermediary SMR interface.
4. The Receiver Intermediary sends an intermediate transport response using the `deliver` operation on the Sender Intermediary TRD interface, indicating that the Receiver has attempted to retrieve the message.
5. The Receiver sends intermediate transport response using the `deliver` operation on the Receiver Intermediary TRD interface, indicating that it has retrieved the message and is waiting for the clinical application to accept the message before confirming.
6. The Receiver Intermediary retransmits the intermediate transport response by invoking the `deliver` operation on the Sender Intermediary TRD interface.
7. The Receiver confirms delivery by invoking the `deliver` operation on the Receiver Intermediary TRD interface with a final transport response.
8. The Receiver Intermediary retransmits the final transport response by invoking the `deliver` operation on the Sender Intermediary TRD interface.
9. The Sender Intermediary stores the final transport response for subsequent retrieval by the Sender.

Other scenarios might include a different set of intermediate responses, but the processing is largely the same.

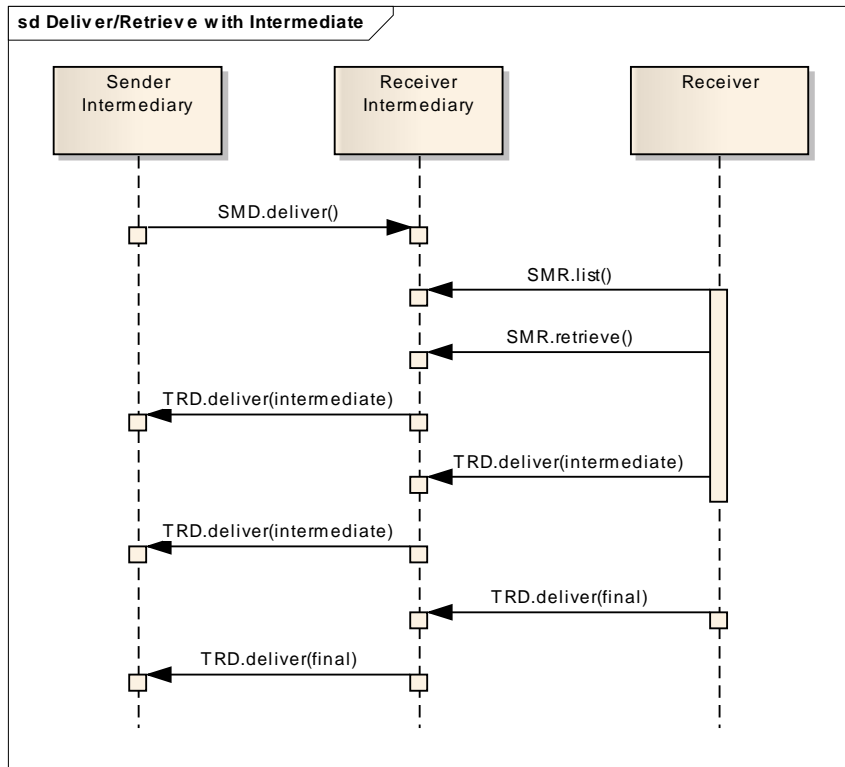


Figure 7: Intermediate Transport Responses

2.2.6.2 Features

- The Sender Intermediary is able to track message status on behalf of the Sender. This supports, for example, call centre operations when messages apparently go missing.

2.2.6.3 Tradeoffs

- There is an increase in complexity and messaging overheads associated with generation, delivery and tracking of transport responses.

2.2.7 Intermediate Transport Responses with Intermediary Skipped

2.2.7.1 Process Fragment

The sequence diagram of figure Figure 8 specifies a fragment of the delivery process involving two intermediaries, with intermediate transport responses used by the Sender Intermediary to track message status. In this case, however, the Receiver Intermediary has indicated that it does not want to handle intermediate transport responses. The steps are as follows:

1. The Sender Intermediary retransmits the message by invoking the `deliver` operation on the Receiver Intermediary SMD interface.
2. The Receiver Intermediary stores the message for subsequent retrieval.
3. The Receiver retrieves the message, first by listing the available messages and then by retrieving those messages using the Receiver Intermediary SMR interface.
4. The Receiver sends intermediate transport response using the `deliver` operation on the Sender Intermediary TRD interface, indicating that it has retrieved the message and is waiting for the clinical application to accept the message before confirming. Note here that the transport response is sent direct to the Sender Intermediary.

5. The Receiver confirms delivery by invoking the `deliver` operation on the Receiver Intermediary TRD interface with a final transport response.
6. The Receiver Intermediary retransmits the final transport response by invoking the `deliver` operation on the Sender Intermediary TRD interface.
7. The Sender Intermediary stores the final transport response for subsequent retrieval by the Sender.

Similar scenarios could occur where a different party in the "chain" is not willing to handle intermediate transport responses.

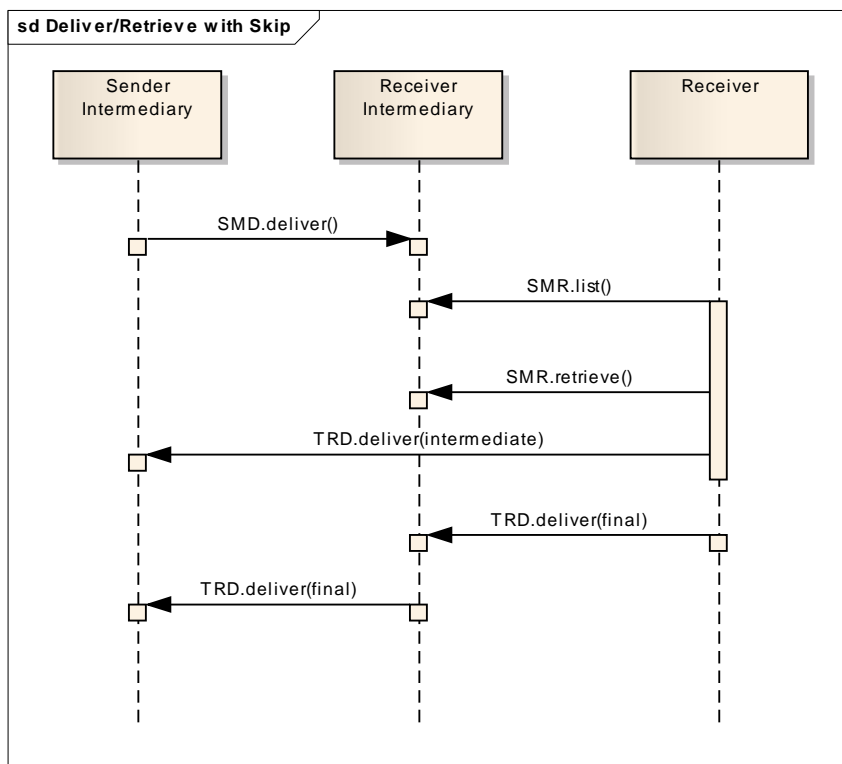


Figure 8: Intermediate Responses with Intermediary Skipped

2.2.7.2 Features

- The Sender Intermediary has some ability to track message status on behalf of the Sender.

2.2.7.3 Tradeoffs

- There is an increase in complexity and messaging overheads associated with generation, delivery and tracking of transport responses.
- There is a further increase in the complexity of message tracking due to the asymmetric path of the intermediate responses.
- The Sender Intermediary might not accept connections from the Receiver.
- The Receiver might not be able to transmit the intermediate response to the Sender Intermediary due to a corporate firewall, if it exists.

2.3 Sealed Immediate Message Delivery Scenarios

The sequence diagrams in the following subsections describe a set of interaction scenarios that might arise in the usage of Secure Message Delivery in immediate mode. Each diagram identifies the roles involved in the scenarios and shows the sequence of interface invocations in the scenario using UML sequence diagrams. In the diagrams that follow, the abbreviation

SIMD refers to the Sealed Immediate Message deliver interface and is used as an operation name prefix.

As noted previously, these scenarios are not exhaustive but illustrate common interaction patterns.

2.3.1 Direct Invocation

2.3.1.1 Process

1. The Sender invokes the `deliver` operation on the Receiver SIMD interface to deliver a message directly to the receiver.
2. The Receiver returns an application response in the return value of the operation.

The process is illustrated in the sequence diagram of Figure 9.

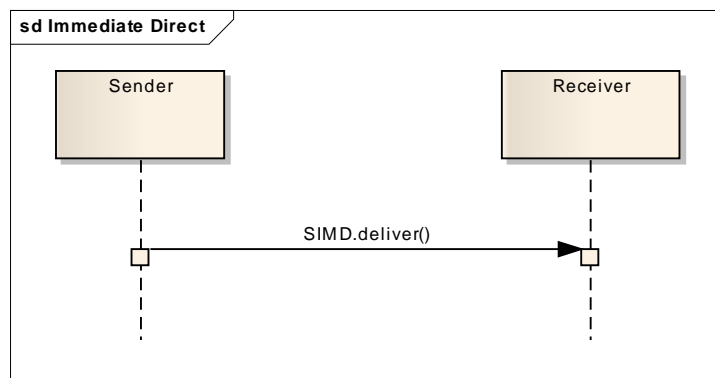


Figure 9: Immediate Direct Invocation

2.3.1.2 Features

- Messages are exchanged in both directions in a single invocation.
- Both request and response messages are delivered directly with no delays incurred through store/retrieve or intermediary processing.

2.3.1.3 Tradeoffs

- The Receiver must host an accessible web service, which normally implies:
 - a publically accessible server with reasonable availability;
 - a registered domain name; and
 - a static IP address.

For small organisations, this is not always possible due to limited IT support or network presence.

- Sender and Receiver applications must be accessible to each other through corporate firewalls, if they exist. Security policy in many larger organisations might prevent such access.
- The Receiver has no assurance on the delivery of the application response message and must rely on the Sender to retry if the response is not received.

2.3.2 Invocation via Sender Intermediary

2.3.2.1 Process

1. The Sender invokes the `deliver` operation on the Sender Intermediary SIMD interface to deliver a message to the Receiver.

2. The Sender Intermediary immediately (before responding) retransmits the message directly to the Receiver.
3. The Receiver returns an application message in the return value of the Sender Intermediary invocation.
4. The Sender Intermediary returns the application message from the Receiver in the return value of the Sender invocation.

The process is illustrated in the sequence diagram of Figure 10

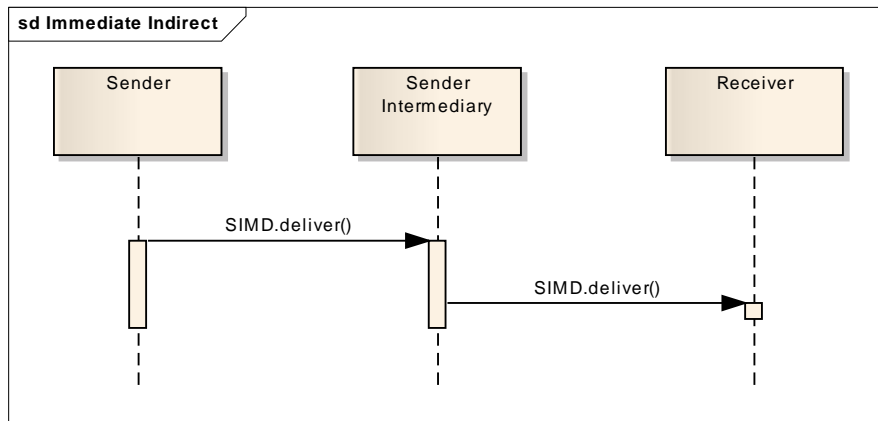


Figure 10: Immediate Invocation via Sender Intermediary

2.3.2.2 Features

- Messages are exchanged in both directions in a single invocation.
- A trust relationship with the Sender Intermediary allows the Sender application to traverse a corporate firewall.

2.3.2.3 Tradeoffs

- The Receiver must host an accessible web service, which normally implies:
 - a publically accessible server with reasonable availability;
 - a registered domain name; and
 - a static IP address.

For small organisations, this is not always possible due to limited IT support or network presence.
- The Receiver application must be accessible through its corporate firewall, if it exists. Security policy in many larger organisations might prevent such access.
- The Receiver has no assurance on the delivery of the application response message and must rely on the Sender to retry if the response is not received.
- The Sender Intermediary might introduce cost and processing delays.

2.3.3 Invocation via Sender and Receiver Intermediary

2.3.3.1 Process

1. The Sender invokes the `deliver` operation on the Sender Intermediary SIMD interface to deliver a message to the Receiver.
2. The Sender Intermediary immediately (before responding) retransmits the message to the Receiver Intermediary.
3. The Receiver Intermediary immediately (before responding) retransmits the message to the Receiver.

4. The Receiver returns an application message in the return value of the Receiver Intermediary invocation.
5. The Receiver Intermediary returns an application message in the return value of the Sender Intermediary invocation.
6. The Sender Intermediary returns the application message from the Receiver in the return value of the Sender invocation.

The process is illustrated in the sequence diagram of

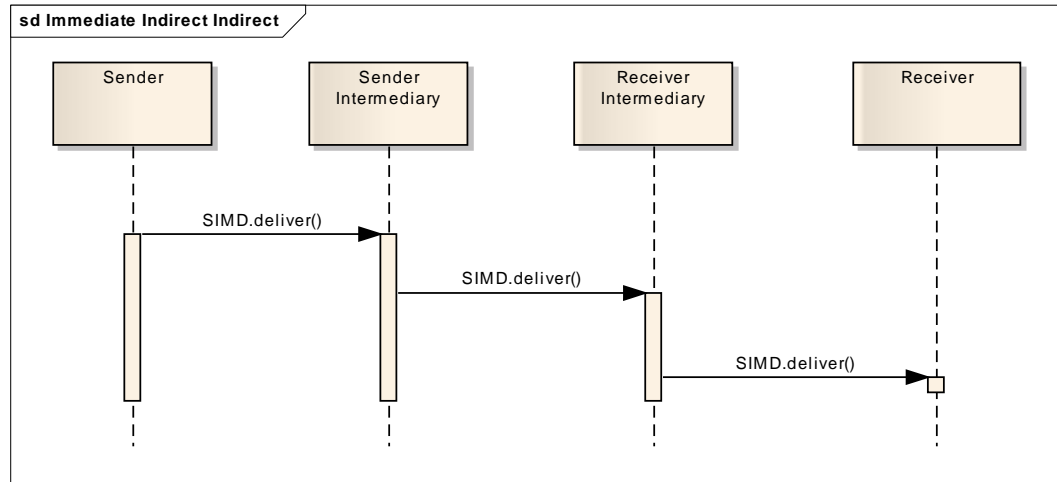


Figure 11: Immediate Invocation via both Sender and Receiver Intermediaries

2.3.3.2 Features

- Messages are exchanged in both directions in a single invocation.
- A trust relationship with the Sender Intermediary allows the Sender application to traverse a corporate firewall.
- A trust relationship with the Receiver Intermediary allows the Receiver application to traverse a corporate firewall.
- The Receiver only needs to be accessible to the Receiver Intermediary, and can operate more easily with a lesser network presence (e.g. private domain name, dynamic IP address).

2.3.3.3 Tradeoffs

- The Intermediaries might introduce cost and processing delays.
- The Receiver has no assurance on the delivery of the application response message and must rely on the Sender to retry if the response is not received.

2.4 Retry Scenarios

The sequence diagrams in the following subsections describe a set of retry scenarios that might arise in the usage of Secure Message Delivery . Each diagram identifies the roles involved in the scenarios and shows the sequence of interface invocations in the scenario using UML sequence diagrams. In the diagrams that follow, the following abbreviations are used as operation name prefixes and refer to the interfaces identified in chapter 1 as follows:

SMD - Secure Message Delivery

TRD - Transport Response Delivery

As noted previously, these scenarios are not exhaustive but illustrate common retry situations.

2.4.1 SOAP Request Failure

2.4.1.1 Process

1. The Sender invokes the `deliver` operation on the Sender Intermediary SMD interface to deliver a message to the receiver.
2. The `deliver` operation fails and either a fault is returned to the Sender or no SOAP response is received by the sender.
3. The Sender invokes the `deliver` operation again on the Sender Intermediary SMD interface. The scenario proceeds successfully as per the scenario in 2.2.2.

The process is illustrated in the sequence diagram of Figure 12

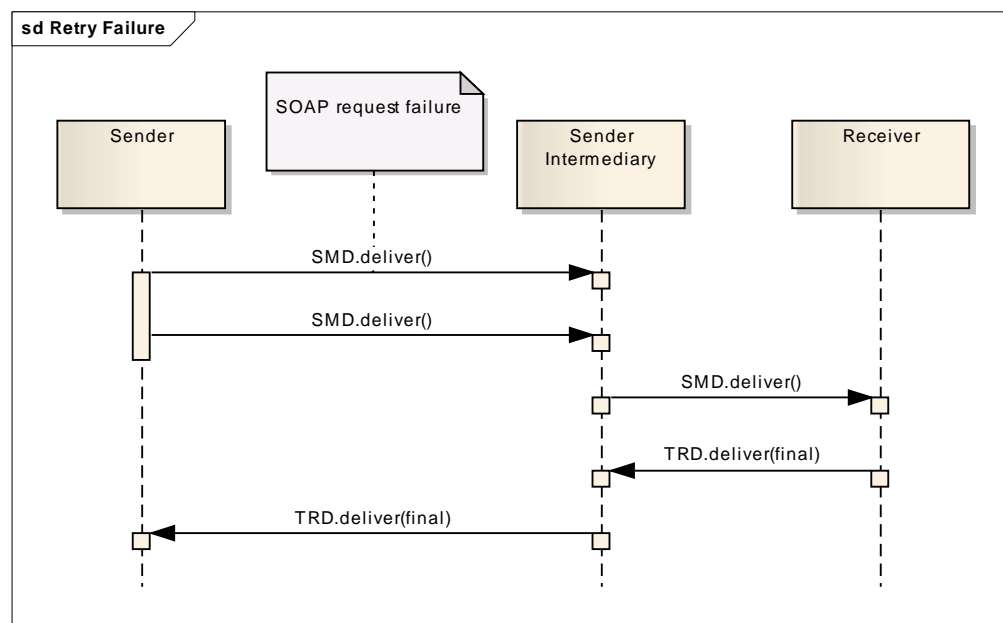


Figure 12: Soap Request Failure

The handling of SOAP request failures for other `deliver` operations is similar.

2.4.2 SOAP Response Failure

2.4.2.1 Process

1. The Sender invokes the `deliver` operation on the Sender Intermediary SMD interface to deliver a message to the receiver.
2. The `deliver` operation succeeds but the SOAP response is not received by the Sender.
3. The Sender Intermediary retransmits the message to the Receiver.
4. The Sender invokes the `deliver` operation again with the same message on the Sender Intermediary SMD interface. The Sender Intermediary detects the duplicate and returns a duplicate response.
5. The Sender Intermediary does not retransmit the message again. The scenario proceeds successfully as per the scenario in 2.2.2.

The process is illustrated in the sequence diagram of Figure 13.

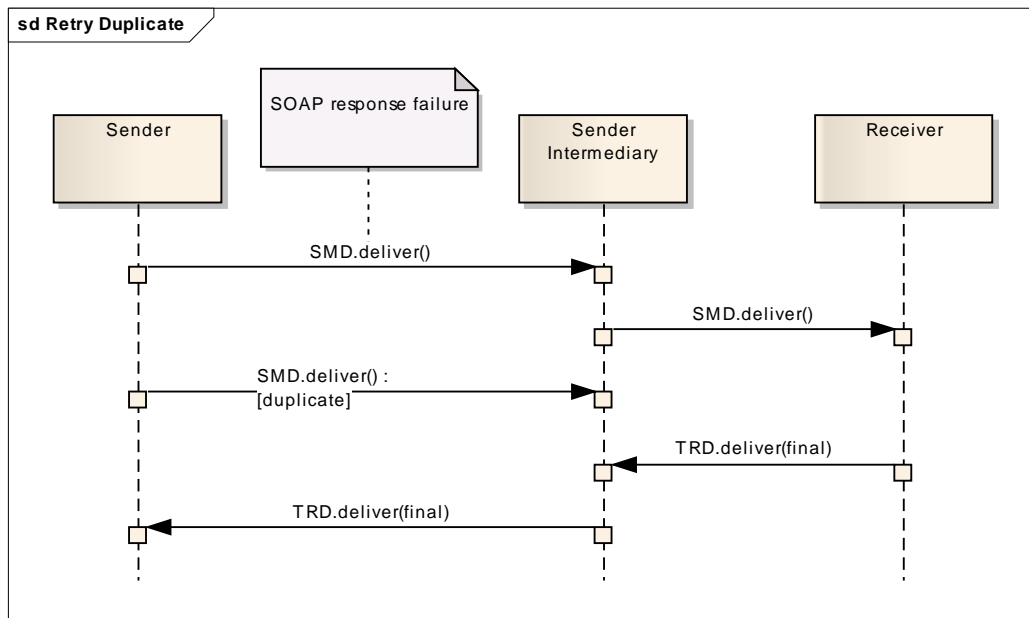


Figure 13: SOAP Response Failure

The handling of SOAP response failures for other deliver operations is similar.

Definitions

This section explains the specialised terminology used in this document.

Shortened Terms

This table lists abbreviations and acronyms in alphabetical order.

Term	Description
CI	Clinical Information
CT	Clinical Terminology
NASH	National Authentication Service for Health
UHI	Unique Healthcare identifiers
Hess	Health eSignature Authority
HL7	Health Level 7
HPI	Healthcare Provider Identifier
HPI-I	Healthcare Provider Identifier for Individuals
HPI-O	Healthcare Provider Identifier for Organisations
NASH	National Authentication Service for Health
NEHTA	National E-Health Transition Authority
PKI	Public Key Infrastructure
QoS	Quality of Service
ELS	Endpoint Locator Service
SLA	Service Level Agreements
SOA	Service Oriented Architecture
UHI	Unique Health Identifiers
URI	Uniform Resource Identifier
WSDL	Web Services Description Language

References

This section lists NEHTA specifications and other documents that provide information for or about this document.

At the time of publication, the document versions indicated are valid. However, as all documents listed below are subject to revision, readers are encouraged to use the most recent versions of these documents.

Specification Documents

The documents listed below are part of the suite delivered in the Secure Message Delivery Specification.

SMD Specification Documents			
[REF]	Document Name	Publisher	Link
[SMD-O]	Secure Message Delivery - Overview	NEHTA 2009	
[SMD-TO]	Secure Message Delivery - Technical Overview	NEHTA 2009	
[SMD-ES]	Secure Message Delivery - Endpoint Specifications	NEHTA 2009	
[SMD-WX]	Secure Message Delivery Service Interface Specification WSDL and XML Schema files v1.0.	NEHTA 2009	
[SMD-AX]	Secure Message Delivery - Compliance and Conformance Assessment Scheme	NEHTA 2009	
[SMD-SPP]	Secure Message Delivery - S/MIME Payload Profile	NEHTA 2009	
[SMD-IG]	Secure Message Delivery - Implementation Guide	NEHTA 2009	To be produced

References

The documents listed below are additional documents that have been cited in this document.

Reference Documents			
[REF]	Document Name	Publisher	Link
[INTER2007]	Interoperability Framework v2.0	NEHTA 2008	http://www.nehta.gov.au/ (Home > Publications)
[NASH-PMP]	National Authentication Service for Health – Project Management Plan	NEHTA 2008	Reference in preparation for future release.
[NATA2005]	National Association of Testing Authorities, April 2005, ISO 15189 - The New Standard for Medical Testing Laboratories	NATA 2005	
[CPIS2008]	Concepts and Patterns for Implementing Services v2.0	NEHTA 2008	http://www.nehta.gov.au/ (Home > Publications)
[WSP2009]	Web Services Profile v3.1	NEHTA 2009	Reference in preparation for future release.
[XSP2009]	XML Secured Payload Profile v1.1	NEHTA 2009	
[QI2008]	Qualified Identifiers v1.0	NEHTA	

Reference Documents			
		2008	
[CA2008]	Connectivity Architecture v1.0	NEHTA 2008	http://www.nehta.gov.au/component/docman/doc_download/624-connectivity-architecture-v10-
[ELS2009]	Endpoint Locator Service	NEHTA 2009	Reference in preparation for future release.
[SMD-SD]	Service Directory Implementation for Secure Messaging	NEHTA 2009	Reference in preparation for future release.
[SMD-ID]	Endpoint Identification for Secure Messaging	NEHTA 2009	Reference in preparation for future release.