EUROPEAN COMMISSION

DIGIT
Connecting Europe Facility

Interface Control Document for WS – default plugin

Date: 10/10/2016

Document Approver(s):

Approver Name	Role
Joao RODRIGUES	CEF eDelivery

Document Reviewers:

Reviewer Name	Role
Joao RODRIGUES	Programme Manager
Adrien FERIAL	Product Owner
Daniels MAARTEN	Tester
Ioana DRAGUSANU	Developer
Yves ADAM	Business Analyst
Christian Koch	eCodex team
Federico Martini	Developer

Summary of Changes:

Version	Date	Created by	Short Description of Changes
0.1	08/02/2016	Pedro TAVARES	Initial version
0.2	18/02/2016	Pedro TAVARES	Update document after revision with Adrien FERIAL
0.3	02/03/2016	Pedro TAVARES	Update document with the backend.wsdl
0.4	15/03/2016	Pedro TAVARES	Update document with ecodex comments
0.5	29/03/2016	Pedro TAVARES	Update document with Adrien FERIAL comments
0.6	31/03/2016	Pedro TAVARES	Update document with Daniels MAARTEN comments
0.7	13/04/2016	Pedro TAVARES	Update document with the datamodel
0.8	22/04/2016	Pedro TAVARES	Update document after revision with Joao RODRIGUES
0.9	29/04/2016	Pedro TAVARES	Update document with Adrien FERIAL comments
1.0	11/05/2016	Pedro TAVARES	Update document with Yves ADAM and Ioana DRAGUSANU comments
1.1	31/05/2016	Pedro TAVARES	Update document with Federico MARTINI comments and updated schema
1.2	05/09/2016	Yves ADAM	Adapt changes for Domibus 3.2 upgrade, include additional XSD's and diagram

			source file
1.3	15/09/2016	Cosmin Baciu	Update the document for Domibus 3.2-RC1
1.4	30/09/2016	Ioana Dragusanu	Updare the document for Domibus 3.2.0

Table of Contents

1. Introduction

1.1. Purpose of the Interface Control Document

This document will univocally define the participant's interface to the Access Point (Corner Two and Corner Three in the four corner topology that will be explained later in this document) component of the eDelivery building block. This document describes the WSDL and the observable behaviour of the interface provided by Domibus 3.2-RC1 and included in the default-ws-plugin.

This interface control document will be used for the understanding of the Access Point (Corner Two and Corner Three in the four corner model) services provided by Domibus 3.2-RC1 delivered by CEF.

There is 1 interface described in this document:

Interface	Description	Version
Backend.wsdl	The backend webservices for Domibus	3.2-RC1

1.2. Scope of the document

This document covers the service interface of the Access Point. It includes information regarding the description of the services available, the list of use cases, the information model and the sequence of message exchanges for the services provided. This specification is limited to the service interface of the Access Point. All other aspects of its implementation are not covered by this document. The ICD specification provides both the provider (i.e. the implementer) of the services and their consumers with a complete specification of the following aspects:

- *Interface Functional Specification*, this specifies the set of services and the operations provided by each service and this is represented by the flows explained in the use cases.
- *Interface Behavioural Specification,* this specifies the expected sequence of steps to be respected by the participants in the implementation when calling a service or a set of services and this is represented by the sequence diagrams presented in the use cases.
- *Interface Message standards,* this specifies the syntax and semantics of the data and this is

1.3. Audience

This document is intended to:

- The Directorate Generals and Services of the European Commission, Member States (MS) and also companies of the private sector wanting to set up a connection between their backend systems and the Access Point. In particular:
    - Architects will find it useful for determining how to best exploit the Access Point to create a fully-fledged solution and as a starting point for connecting a Back-Office system to the Access Point.
    - Analysts will find it useful to understand the Access Point that will enable them to have an holistic and detailed view of the operations and data involved in the use cases.

- Developers will find it essential as a basis of their development concerning the Access Point plugin services.
- Testers can use this document in order to test the interface by following the use cases described.

1.4. References

The table below provides the reader with the list of reference documents.

#	Document	Contents outline
[REF1]	Introduction to the Connecting Europe Facility - eDelivery building block	Overview of eDelivery
[REF2]	Using HTTP Methods for RESTful Services	Short description of HTTP Methods for RESTful Services
[REF3]	Business Document Metadata Service Location - Software Architecture Document	This document is the Software Architecture document of the CIPA eDelivery Business Document Metadata Service Location application (BDMSL) sample implementation. It intends to provide detailed information about the project: 1) An overview of the solution 2) The different layers 3) The principles governing its software architecture.
[REF4]	ebXML (Electronic Business using eXtensible Markup Language)	ebXML (Electronic Business using eXtensible Markup Language)
[REF5]	Web Services Description Language (WSDL) 1.1	Web Services Description Language (WSDL) 1.1 WS-I Basic Profile Version 1.1
[REF6]	XML Schema 1.1	XML Schema 1.1
[REF7]	Extensible Markup Language (XML) 1.0	Extensible Markup Language (XML) 1.0
[REF8]	Hypertext Transfer Protocol 1.1	Hypertext Transfer Protocol 1.1
[REF9]	SOAP Messages with Attachments	SOAP Messages with Attachments
[REF10]	AS4 Profile of ebMS 3.0 Version 1.0	AS4 Profile of ebMS 3.0 Version 1.0
[REF11]	eSens - profile	http://wiki.ds.unipi.gr/display/ESENS/PR+-+AS4
[REF12]	eDelivery – Pmode Configuration	eDelivery – Pmode Configuration *(will be available at a later stage)*
[REF13]	http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/	XSDs for ebms3
[REF14]	http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/cs02/ebms_core-3.0-spec-cs-02.pdf	ebXML (Electronic Business using eXtensible Markup Language)

2. Interface Functional Specification

In order to understand the Use Cases that will be described below it is important to explain the topology; i.e. the four – corner model.

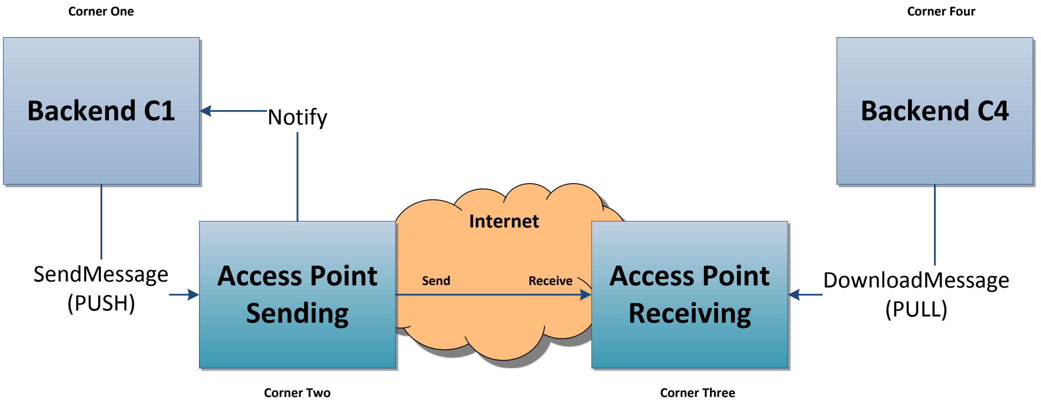


Figure 1 - the four corner model

In this model we have the following elements:

- Corner One (C1): Backend C1 is the system that will use the sending AP (Access Point)
- Corner Two (C2): Sending Access Point C2
- Corner Three (C3): Receiving Access Point C3
- Corner Four (C4): Backend C4 is the system that will use the receiving AP (Access Point)

There are two backend adapters (i.e. corner one and corner four). They send messages to and download messages from the AS4 APs configured in the PMode configuration files.

2.1. Use cases

2.1.1. Actors

Actor	Definition
Backend C1	Any participant sending messages to any other Backend C4 and using the Sending AP C2 for that purpose.
Backend C4	Any participant downloading messages from any other Backend C1 and using the Receiving AP C3 for that purpose.

2.1.2. Use cases diagram

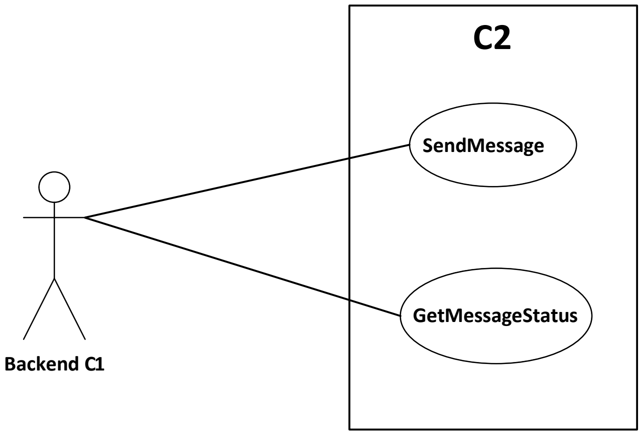


Figure 2 - Backend C1 Use cases diagram

ID	UC	Short description	Oper.	System
UC01	**Send Message**	Submit any type of document from an Backend C1 to a Backend C4	sendMessage	Domibus 3.2-RC1
UC03	**Get Status of the Message**	Get the status of the Message	getMessageStatus	Domibus 3.2-RC1

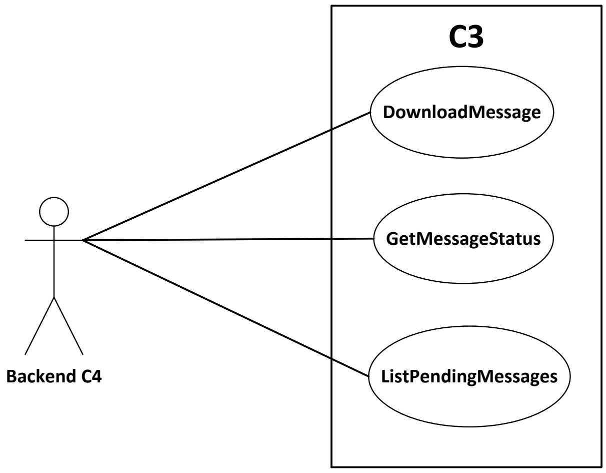


Figure 3 - backend C4 Use cases diagram

ID	UC	Short description	Oper.	System
UC02	**Download Message**	Download the message from the Receiving AP C3	downloadMessage	Domibus 3.2-RC1
UC03	**Get Status of the Message**	Get the status of the Message	getMessageStatus	Domibus 3.2-RC1
UC04	**ListPending Messages**	Check the pending messages to be downloaded by the Backend C4 from C3	listPendingMessages	Domibus 3.2-RC1

2.1.3. Detailed use cases

The following paragraphs define the use cases listed above with more detail.

The *Interface Functional Specification* is described in the detailed uses cases using the Request and the Response examples. It is important to remark that the Inputs and Responses provided as examples for the uses cases are based on a specific PMode configuration.

As defined in the eSens Specification Library, a PMode is the contextual information that governs the processing of a particular message (thus is basically a set of configuration parameters). The PMode associated with a message determines, among other things, which security and/or which reliability protocol and parameters, as well as which MEP (Message Exchange Pattern) is being used when sending a message. The technical representation of the PMode configuration is implementation-dependent. C1 and C4 may be one or more participants.

The state machine diagrams presented below depict the various states in which a message may be during its lifecycle when submitting or downloading the message. These are presented in order to have a more comprehensive vision of the process that the messages go through. It is also important to remark that also the sequence diagram of the basic flow is presented in the use cases.

On C2, the state machine diagram for submitting the message:

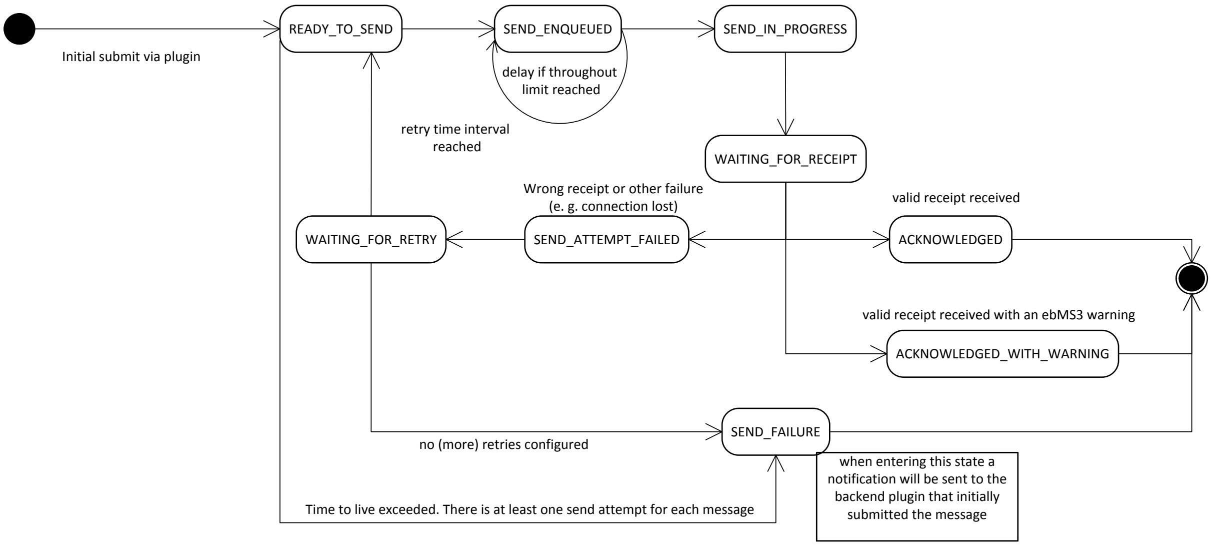

READY_TO_SEND
SEND_ENQUEUED
SEND_IN_PROGRESS
Initial submit via plugin
delay if throughout limit reached
retry time interval reached
WAITING_FOR_RECEIPT
Wrong receipt or other failure (e. g. connection lost)
valid receipt received
WAITING_FOR_RETRY
SEND_ATTEMPT_FAILED
ACKNOWLEDGED
valid receipt received with an ebMS3 warning
ACKNOWLEDGED_WITH_WARNING
SEND_FAILURE
no (more) retries configured
when entering this state a notification will be sent to the backend plugin that initially submitted the message
Time to live exceeded. There is at least one send attempt for each message

Figure 4 – State machine of C2

On C3, the state machine diagram for downloading the message:

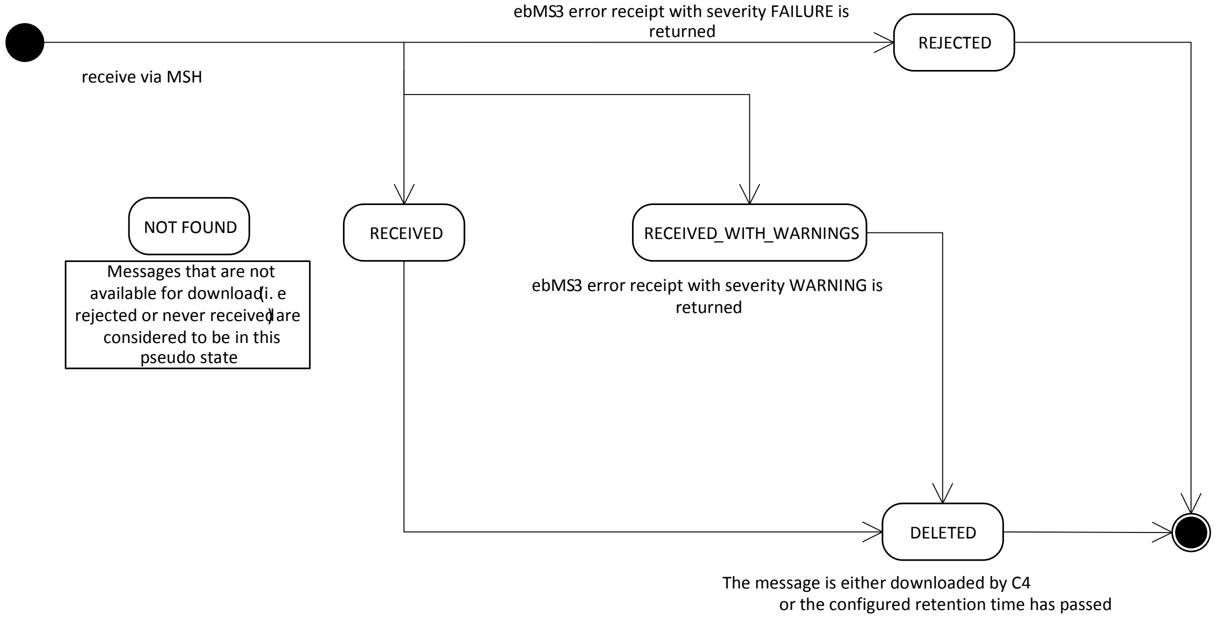

ebMS3 error receipt with severity FAILURE is returned
REJECTED
receive via MSH
NOT FOUND
Messages that are not available for download(i. e rejected or never received) are considered to be in this pseudo state
RECEIVED
RECEIVED_WITH_WARNINGS
ebMS3 error receipt with severity WARNING is returned
DELETED
The message is either downloaded by C4 or the configured retention time has passed

Figure 5 - State machine of C3

2.1.3.1. *UC01 – Send Message*

Brief description

Send any message with attachments from Backend C1 to the Backend C4. The response from C2 to C1 is synchronous and contains a messageId.

The state machine of the outgoing messages is the following:

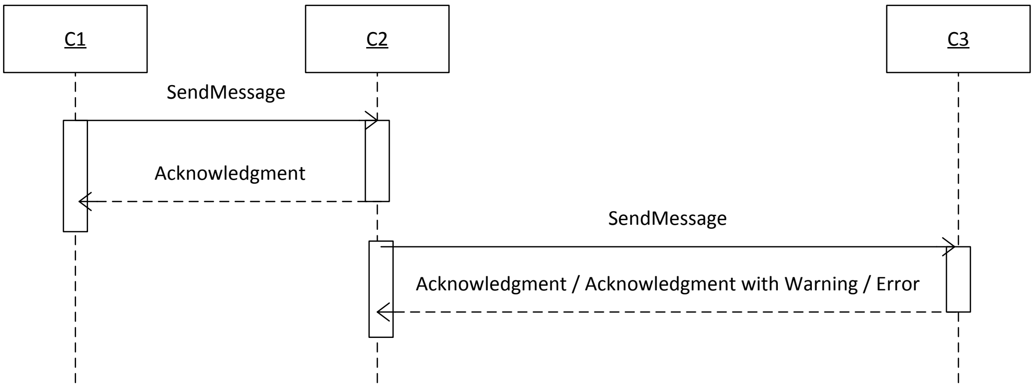


Figure 6 - Sequence Diagram C1 to C2 – SendMessage

Actors

C1			Backend C1
	C2		Access Point C2
		C3	Access Point C3

Preconditions

C1			Backend C1 has a message to send.
C1			The message is valid. A message is valid if it respects the message standard format (see Annex 5.1 - Interface Message standards)
	C2	C3	The Sending AP (C2) and the Receiving AP (C3) are up and running and properly configured.

Basic Flow			C2 State	C3 state
Actor	Step			
C1	1	Backend C1 submits the message.	n/a	
C2	2	Message arrives at C2 with the state READY_TO_SEND.	READY_TO_SEND	
C2	3	C2 sends an ACK to C1 containing the ID of the message.	READY_TO_SEND	
C1	4	Backend C1 receives the ACK from C2 containing the ID of the message.	READY_TO_SEND	
C2	5	The message directly passes through to SEND_ENQUEUED, meaning that it is available for processing. All messages go through this step regardless of load.	SEND_ENQUEUED	
C2	6	The Sending AP (C2) sends the message to the Receiving AP (C3). The message status is updated to SEND_IN_PROGRESS whilst the message is being sent.	SEND_IN_PROGRES	
C3	7	The Receiving AP (C3) receives the message from the Sending AP (C2)	SEND_IN_PROGRES	
C2	9	Once the sending process finishes the status changes to WAITING_FOR_RECEIPT.	WAITING_FOR_RECEIPT	
C3	8	Once the reception is finished by C3, the status changes to RECEIVED	WAITING_FOR_RECEIPT	RECEIVED
C3	10	The Receiving AP (C3) responds ACK to C2	WAITING_FOR_RECEIPT	RECEIVED

C2	11	The status of the message changes to ACKNOWLEDGED. If configured in the PMode for non-repudiation, the receipt SHOULD contain a single ebbpsig:NonRepudiationInformation child element. The value of eb:MessageInfo/eb:RefToMessageId MUST refer to the message for which this signal is a receipt.	ACKNOWLEDGED	RECEIVED
	12	Use case ends in successful condition.		

Alternative flow			C2 State	C3 state
C2	A2.1	**The Backend C1 has not provided a messageID (the field is missing)**		
C2	A2.1.1	C2 creates one automatically and notifies C1 with a system response that contains the created messageID. The messageID does not change during the lifecycle of the message.		
C2	A2.1.2	Continue to step 3		
C3	A8.1	**Undefined condition**	SEND_IN_PROGRES	
C3	A8.1.1	Once the reception is finished by C3, the status changes to RECEIVED WITH WARNINGS	SEND_IN_PROGRES	RECEIVED WITH WARNINGS
C3	A8.1.2	The Receiving AP (C3) reponds ACK WITH WARNING to C2	SEND_IN_PROGRES	RECEIVED WITH WARNINGS
C2	A8.1.3	C2 receives the response from C3 and the status changes to ACKNOWLEDGED WITH WARNING.	ACKNOWLEDGED WITH WARNING	RECEIVED WITH WARNINGS
C2	A8.1.4	Continue to step 12		

Exception flow			C2 State	C3 state
C2	**E2.1**	**The ID provided by C1 already exists**		
	E2.1.1	The parameter PMode[1].ReceptionAwareness.DuplicateDetection must be set to true		
	E2.1.2	The status of the message changes to SEND_FAILURE..	SEND_FAILURE	
C2	**E11.1**	**Wrong receipt received or any other failure (e.g connection lost)**		
C2	E11.1.1	The status of the message changes to SEND_ATTEMPT_FAILED	SEND_ATTEMPT_FAILED	
C2	E11.1.2	Cron job awakens and sees the message in the SEND_ATTEMPT_FAILED status	SEND_ATTEMPT_FAILED	
C2	E11.1.3	The status changes to WAITING_FOR_RETRY	WAITING_FOR_RETRY	
C2	E11.1.3	Continue to step 3		
C2	**E11.1.3.1**	**The maximum number or retries (Configurable via PMode on a by-usecase basis) has been reached**		
C2	E11.1.3.1.1	The status of the message changes to SEND_FAILURE.	SEND_FAILURE	
	E11.1.3.1.2	A notification can be sent to the Backend C1 that initially submitted the message.	SEND_FAILURE	
C2	E11.1.3.1.3	Use case ends in failure condition.		

Post conditions

Successful conditions

The operation is a success if GetMessageStatus in C2 is ACKNOWLEDGED or ACKNOWLEDGED WITH WARNING and this means that the Receiving AP (C3) has received the message submitted by the Backend C1 and the status in C3 is RECEIVED or RECEIVED WITH WARNINGS. The method GetMessageStatus must be called with the identifier of the message received in the response or specified in the request.

Failure Conditions

Errors may be sent as SOAP Fault or as http:5XX .

Example of the request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:ns="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/" xmlns:_1="http://org.ecodex.backend/1_1/" xmlns:xm="http://www.w3.org/2005/05/xmlmime">
  <soap:Header>
    <ns:Messaging>
      <ns:UserMessage>
        <ns:PartyInfo>
          <ns:From>
            <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">C2</ns:PartyId>
            <ns:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</ns:Role>
          </ns:From>
          <ns:To>
            <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered"> C3</ns:PartyId>
            <ns:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</ns:Role>
          </ns:To>
        </ns:PartyInfo>
        <ns:CollaborationInfo>
          <ns:Service type="EU-CEG">ConnectivityService</ns:Service>
          <ns:Action>SubmitMessage</ns:Action>
          <ns:ConversationId>f54e1b0e-7179-4a94-a2a8-73a6df831eae</ns:ConversationId>
        </ns:CollaborationInfo>
        <ns:MessageProperties>
          <ns:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C1</ns:Property>
          <ns:Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C4</ns:Property>
        </ns:MessageProperties>
        <ns:PayloadInfo>
          <ns:PartInfo href="cid:message">
            <ns:PartProperties>
              <ns:Property name="MimeType">application/xml</ns:Property>
              <ns:Property name="Description">Message Payload</ns:Property>
            </ns:PartProperties>
          </ns:PartInfo>
        </ns:PayloadInfo>
      </ns:UserMessage>
    </ns:Messaging>
  </soap:Header>
  <soap:Body>
    <_1:sendRequest>
      <payload
payloadId="cid:message">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPGhlbGxvPndvcmxkPC9oZWxsbz4=</payload>
    </_1:sendRequest>
  </soap:Body>
</soap:Envelope>
```

Example of the response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns6:sendResponse xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/" xmlns:eb="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/" xmlns:S12="http://www.w3.org/2003/05/soap-envelope"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:ns6="http://org.ecodex.backend/1_1/">
      <messageID>4078cfea-74e9-4058-9d14-1dceee597abd@domibus.eu</messageID>
    </ns6:sendResponse>
  </soap:Body>
</soap:Envelope>
```

Example of a Receipt for Acknowledgment:

```
<soap:Envelope xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:eb3="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/" xmlns:ebbp="http://docs.oasis-open.org/ebxml-bp/ebbp-signals-2.0" xmlns:ebint="http://docs.oasis-
open.org/ebxml-msg/ebms/v3.0/ns/multihop/200902/" xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:wsa="http://www.w3.org/2005/08/addressing" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
 <S12:Header xmlns:S12="http://www.w3.org/2003/05/soap-envelope">
   <eb3:Messaging S12:mustUnderstand="true" id="_ebmessaging_N65542" wsu:Id="_0a90f0b6-b44c-43c7-837c-f81a83c97a4d">
     <eb3:SignalMessage>
```

```
    <eb3:MessageInfo>
     <eb3:Timestamp>2016-04-28T15:40:38.915Z</eb3:Timestamp>
     <eb3:MessageId>68b5a252-3af2-42e3-9dec-6b29a5ec3d2d@domibus.eu</eb3:MessageId>
     <eb3:RefToMessageId>44e20f12-6f08-4220-8989-0b82622b453b@domibus.eu</eb3:RefToMessageId>
    </eb3:MessageInfo>
    <eb3:Receipt>
     <ebbp:NonRepudiationInformation>
      <ebbp:MessagePartNRInformation>
       <ds:Reference xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" URI="#_68fc8d83-d9ec-4557-a7a2-9a9d2d81621e">
        <ds:Transforms>
         <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="soap"/>
         </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>adXiiJB9dZqBPPdOiAyqf71EAns8lcvXbuDgJLAY/kU=</ds:DigestValue>
       </ds:Reference>
      </ebbp:MessagePartNRInformation>
      <ebbp:MessagePartNRInformation>
       <ds:Reference xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" URI="cid:message">
        <ds:Transforms>
         <ds:Transform Algorithm="http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-1.1#Attachment-Content-Signature-
Transform"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>ORQHupO3X8Cw/ou02ld5SOPQI67JJccX/cl1wcnhUPM=</ds:DigestValue>
       </ds:Reference>
      </ebbp:MessagePartNRInformation>
     </ebbp:NonRepudiationInformation>
    </eb3:Receipt>
   </eb3:SignalMessage>
  </eb3:Messaging>
  <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
soap:mustUnderstand="true">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-9472133f-1b8a-48dc-9e67-62907a13dd05">
    <ds:SignedInfo>
     <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
      <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds eb3 ebbp ebint soap wsa wsse
wsu"/>
     </ds:CanonicalizationMethod>
     <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
     <ds:Reference URI="#_0a90f0b6-b44c-43c7-837c-f81a83c97a4d">
      <ds:Transforms>
       <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="ds ebbp ebint soap wsa wsse"/>
       </ds:Transform>
      </ds:Transforms>
      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
      <ds:DigestValue>VExWpf0Knd9WsDs2sqrnnbuSW99V25zYqFqFJz/tZwM=</ds:DigestValue>
     </ds:Reference>
    </ds:SignedInfo>

<ds:SignatureValue>MpbhEgDJBHtwwyl6ecTHOvLVtGgMI7l5rHWtbeBXnLGtMBrpEEziPlJ+FYy1Tk0uDIGTqsJKhqegTAl9wbj4o/osd2fKmpCsX
H3GjAbDnlfwRFpNbfiwG874EtXi2Jy/qwY9pv9JMtHCP/Lloj7zKD+8gbiRMMOvGh2z+7IcpObY/drVdpgWia7by/qZtQP9MTlnRwxatscuF/WvNa
RudNfXusHuW1FEjsNDhBYtQ/U1mRDklhF9rIi1/YSFmJ4hSwnJ6jTzaAwDiyClxsmY+Z19VwzDdeEGXsKCWa0kp64zRapj8mT9ED7ZcmAuMtrhrX
p1+1q4bqN5TOAaMw6iQQ==</ds:SignatureValue>
    <ds:KeyInfo Id="KI-ebadf807-a56e-4829-87f7-ff1c6f1d5b38">
     <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="STR-76ed2da3-37e0-4af7-
a86c-73755525e68f">
      <wsse:KeyIdentifier EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-
1.0#X509SubjectKeyIdentifier">2z3u6o+KPctatpQhhk6S9qEItb4=</wsse:KeyIdentifier>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
   </ds:Signature>
  </wsse:Security>
 </S12:Header>
 <soap:Body wsu:Id="N65660"/>
```

```
</soap:Envelope>
```

Example of a Receipt with Error

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
    <env:Header>
        <eb:Messaging xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" env:mustUnderstand="true"
wsu:Id="_c49b0368-de33-4d9d-90e1-e80011415126">
            <eb:SignalMessage>
                <eb:MessageInfo>
                    <eb:Timestamp>2016-04-28T16:04:37.467Z</eb:Timestamp>
                    <eb:MessageId>6b2c15e2-69c7-4509-a6dc-f7da92b091bd@domibus.eu</eb:MessageId>
                    <eb:RefToMessageId>bd667106-e4fb-472d-a2cc-
1c7c1ddc5b24@domibus.eu</eb:RefToMessageId>
                </eb:MessageInfo>
                <eb:Error category="PROCESSING" errorCode="EBMS:0010" origin="ebMS"
refToMessageInError="bd667106-e4fb-472d-a2cc-1c7c1ddc5b24@domibus.eu" severity="failure"
shortDescription="ProcessingModeMismatch">
                    <eb:Description xml:lang="">The ebMS header or another header (e.g. reliability, security)
expected by the MSH is not compatible with the expected content, based on the associated P-Mode.</eb:Description>
                    <eb:ErrorDetail>Payload profiling for this exchange does not include a payload with CID:
cid:message</eb:ErrorDetail>
                </eb:Error>
            </eb:SignalMessage>
        </eb:Messaging>
        <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" env:mustUnderstand="true">
                      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="SIG-ceee7ef2-7dc2-
4dd5-ac20-64dd39930295">
                <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                        <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
PrefixList="env"/>
                    </ds:CanonicalizationMethod>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
                    <ds:Reference URI="#_c49b0368-de33-4d9d-90e1-e80011415126">
                        <ds:Transforms>
                            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                                <ec:InclusiveNamespaces
xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList=""/>
                            </ds:Transform>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>

<ds:DigestValue>h3h/rjVfxae7jN/mtVa9UGr9Rn7+KUXTZDlrZagcp3Y=</ds:DigestValue>
                    </ds:Reference>
                </ds:SignedInfo>

<ds:SignatureValue>mIOw1IFVzYXkdC79xlE8leKkMEGn15GFMKs9X0W+aPHWW6niNwg2O1zAsHvVQKFPjNAFmVFya4DaXn3Adv9+/bis1Hzw
wf/ccVQokOgMKFkyUb5KpFVWWBa3TPXUXzUcjFtZ1a0dG3eVmWa4lg9+eOBGNvnbkpsPLjkazGPKvqV+cAXg0vjewtzWQklakOD1noBUE1ETa
BJb9aKMkFviteELoG+pRMB5IMWt7zvQ3J7WfStupLBbqyxfhKCFl7nckV27mPR0fDBryX1grR/LseeQqLuwHz1j+pqPdVm2owPVabAITVYd3xBrhi
pdZ7ZxQ1GOM+kL0O0hW4snHArWZA==</ds:SignatureValue>
                <ds:KeyInfo Id="KI-004d7428-247c-4187-b734-f32710230e13">
                    <wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="STR-3792fa17-5fe5-4f9d-a20f-c7165d65dfea">
                        <wsse:KeyIdentifier EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-
token-profile-1.0#X509SubjectKeyIdentifier">2z3u6o+KPctatpQhhk6S9qEItb4=</wsse:KeyIdentifier>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
        </wsse:Security>
```

```
	</env:Header>
	<env:Body>
		<env:Fault>
			<env:Code>
				<env:Value>env:Receiver</env:Value>
			</env:Code>
			<env:Reason>
				<env:Text xml:lang="en">An error occurred while processing your request. Please check the message header for more details.</env:Text>
			</env:Reason>
		</env:Fault>
	</env:Body>
</env:Envelope>
```

Special requirements

N/A

Security

When Access Point C2 is configured to require authentication, Backend C1 must authenticate itself to send messages.
There are two possible roles ROLE_USER and ROLE_ADMIN.
In case C1 authenticates with the ROLE_USER, the value of the “originalSender” in the message properties must be equal to the authenticated user:
e.g <ns:Property name=“originalSender”>urn:oasis:names:tc:ebcore:partyid-type:unregistered:C1</ns:Property>

ROLE_ADMIN has the right to send messages on behalf of other users (the value of the “originalSender” is not checked).

2.1.3.2. *UC02 - Download Message*

Brief description

Download any type of message sent from Backend C1 to Backend C4. The retrieval of the message is based on a PULL mechanism. C4 downloads the message from C3.

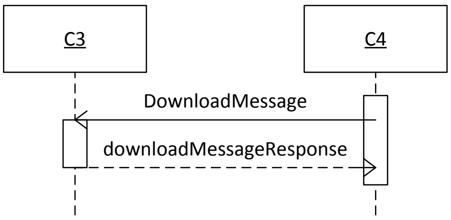


Figure 7 - Sequence Diagram C4 to C3 – DownloadMessage

Actors

C3	Access Point C3
C4	Backend C4

Preconditions

C3	There is at least one message sent by AP C2 and successfully received in the Receiving AP C3.
C3	The Receiving AP (C3) is up and running and properly configured.
C4	C4 Has previously requested information about pending messages from C3. C3 has returned a response containing the messageID('s) of the message(s) received (cf. UC04).

Basic Flow

Actor	Step		C3 State
C4	1	Requests, to the Receiving AP C3, the service downloadMessage by providing the messageID	RECEIVED
C3	2	Receiving AP C3 retrieves and sends the information downloadMessageResponse to C4 as response to his request. This is the payload (message content and attachments) and metadata, analogous to the message sent from C1 to C2 in UC01.	RECEIVED
C4	3	Receives the message as sent by C1	RECEIVED

C3	4	Deletes the payload of the message from the database. NB: While the message metadata is still recoverable by a Domibus administrator all payload data is purged. This is necessary to be able to prove message exchanges in case of disputes. It is possible to produce the signature of a payload but not the payload itself.	DELETED
C3	5	The status of the message changes to *DELETED* when the message is downloaded by the Backend C4	DELETED
	6	Use case ends	DELETED
Alternative flow			**C3 State**
C3	A1.1	**Configured retention time has passed**	RECEIVED
C3	A1.1.1	Go directly to step 4	RECEIVED

Exception flow			**C3 State**
C3	**E2.1**	**Wrong messageID (malformed or missing), this is a condition of failure.**	**(NOT FOUND)**
C3	E2.1.1	The NOT FOUND status is a pseudo state for messages that are not available for download (were never received or were rejected).	(NOT FOUND)
C3	E2.1.2	Use case ends in failure condition	(NOT FOUND)
C3	E2.2	**Message status is REJECTED**	
C3	E2.2.1	Respond to C4 with an error indicating that the message from C1 was rejected at receive time	
C3	E2.2.2	Use case ends in failure condition	
Post conditions			
		Successful conditions	
		It is a success if the Message Status is *ACKOWLEDGED* or *ACKOWLEDGED WITH WARNINGS* on C2 and *DELETED* on C3 which means that the Backend C4 has downloaded the message submitted by the Backend C1. Payload of the message is deleted from C3's database.	
		Failure Conditions	
		No message payload is returned to C4. The response contains a description of the encountered error. The operation is not a success if GetMessageStatus is *SEND_FAILURE* on C2 and *NOT FOUND* or *REJECTED* on C3 and this means that the Backend C4 has not been able to download the message submitted by the Backend C1. If the downloadMessage is called with a wrong messageID (malformed or missing), this is a condition of failure. The *NOT FOUND* status is a pseudo state for messages that are not available for download (were never received or were rejected).	

Example of the request

```
 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:_1="http://org.ecodex.backend/1_1/">
  <soap:Header/>
  <soap:Body>
    <_1:downloadMessageRequest>
      <messageID>4078cfea-74e9-4058-9d14-1dceee597abd@domibus.eu</messageID>
    </_1:downloadMessageRequest>
  </soap:Body>
</soap:Envelope>
```

Example of the response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
    <eb:Messaging soap:mustUnderstand="true" xmlns:ns7="http://org.ecodex.backend/1_1/"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:eb="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/">
      <eb:UserMessage mpc="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC">
        <eb:MessageInfo>
          <eb:Timestamp>2016-03-31T09:00:44.418Z</eb:Timestamp>
          <eb:MessageId>4078cfea-74e9-4058-9d14-1dceee597abd@domibus.eu</eb:MessageId>
        </eb:MessageInfo>
        <eb:PartyInfo>
          <eb:From>
                        <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">C2</ns:PartyId>
            <eb:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</eb:Role>
          </eb:From>
          <eb:To>
            <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered"> C3</ns:PartyId>
            <eb:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</eb:Role>
          </eb:To>
        </eb:PartyInfo>
        <eb:CollaborationInfo>
          <eb:AgreementRef/>
          <eb:Service type="EU-CEG">ConnectivityService</eb:Service>
          <eb:Action>SubmitMessage</eb:Action>
          <eb:ConversationId>f54e1b0e-7179-4a94-a2a8-73a6df831eae</eb:ConversationId>
        </eb:CollaborationInfo>
        <eb:MessageProperties>
          <eb:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C1</eb:Property>
          <eb:Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C4</eb:Property>
        </eb:MessageProperties>
        <eb:PayloadInfo>
          <eb:PartInfo href="cid:message">
            <eb:Schema/>
            <eb:PartProperties>
              <eb:Property name="MimeType">application/xml</eb:Property>
              <eb:Property name="Description">Message Payload</eb:Property>
            </eb:PartProperties>
          </eb:PartInfo>
        </eb:PayloadInfo>
      </eb:UserMessage>
    </eb:Messaging>
  </soap:Header>
  <soap:Body>
    <ns6:downloadMessageResponse xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/" xmlns:eb="http://docs.oasis-
open.org/ebxml-msg/ebms/v3.0/ns/core/200704/" xmlns:S12="http://www.w3.org/2003/05/soap-envelope"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime" xmlns:ns6="http://org.ecodex.backend/1_1/">
      <payload
payloadId="cid:message">PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz4KPGhlbGxvPndvcmxkPC9oZWxsbz4=</payload>
    </ns6:downloadMessageResponse>
  </soap:Body>
</soap:Envelope>
```

Security

When Access Point C3 is configured to require authentication, Backend C4 must authenticate itself to download messages.
There are two possible roles ROLE_USER and ROLE_ADMIN.
In case C4 authenticates with the ROLE_USER, the value of the “finalRecipient” in the message properties must be equal to the authenticated user:
e.g <ns:Property name=“finalRecipient”>urn:oasis:names:tc:ebcore:partyid-type:unregistered:C4</ns:Property>

ROLE_ADMIN has the right to download messages on behalf of other users (the value of the “finalRecipient” is not checked).

*2.1.3.3. **UC03 - Get the status of the Message***

Brief description

Get the status of the Message sent from Backend C1 or received by the Backend C4

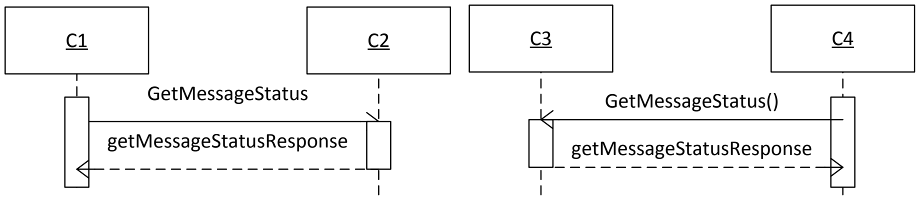


Figure 8 - Sequence Diagram – GetMessageStatus

Actors

C1	Backend C1
C4	Backend C4

Preconditions

There is at least one message sent by Backend C1 or to be downloaded by Backend C4.

The Sending AP C2 and the Receiving AP C3 are up and running and properly configured.

Basic flow event

Step

1. Backend C1 or the Backend C4 launch a GetStatusRequest using the messageId.
2. The Access Point (Sending AP C2 or Receiving AP C3) retrieve the getMessageStatusResponse.
3. Use case ends.

Exception flow

N/A

Post conditions

Successful conditions

The operation is a success if GetMessageStatusResponse retrieves any status of the following:

- READY_TO_SEND
- SEND_ENQUEUED
- SEND_IN_PROGRESS
- WAITING_FOR_RECEIPT
- ACKNOWLEDGED
- ACKNOWLEDGED_WITH_WARNING
- SEND_ATTEMPT_FAILED
- SEND_FAILURE
- NOT_FOUND
- WAITING_FOR_RETRY
- RECEIVED
- RECEIVED_WITH_WARNINGS
- DELETED

Failure Conditions

The message doesn't exist .

Special requirements

N/A

Security

When the Access Point is configured to require authentication, the Backend must authenticate itself to get the status of the message.
There are two possible roles ROLE_USER and ROLE_ADMIN.

Only ROLE_ADMIN has the right to get the message status.

Example of the request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:_1="http://org.ecodex.backend/1_1/">
  <soap:Header/>
  <soap:Body>
    <_1:getStatusRequest>
      <messageID>4078cfea-74e9-4058-9d14-1dceee597abd@domibus.eu</messageID>
    </_1:getStatusRequest>
  </soap:Body>
</soap:Envelope>
```

Example of the response

```
 <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
     <ns6:getMessageStatusResponse xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/" xmlns:eb="http://docs.oasis-
open.org/ebxml-msg/ebms/v3.0/ns/core/200704/" xmlns:S12="http://www.w3.org/2003/05/soap-envelope"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
xmlns:ns6="http://org.ecodex.backend/1_1/">RECEIVED</ns6:getMessageStatusResponse>
   </soap:Body>
</soap:Envelope>
```

2.1.3.4. *UC04 – List Pending Messages*

Brief description

Check the pending messages to be downloaded by the Backend C4 from C3 filtered by the authenticated user (if any)

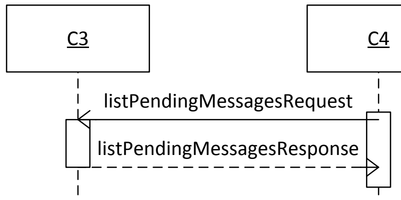


Figure 9 - Sequence Diagram C4 to C3 – ListPendingMessages

Actors	
C4	Backend C4
Preconditions	
	There is at least one message be downloaded by Backend C4. The Receiving AP C3 is up and running and properly configured.

Basic flow event

Step

1. Backend C4 launches the service listPendingMessages.
2. The Access Point (Receiving AP C3) retrieves the list of messageIds for messages with status RECEIVED or RECEIVED_WITH_WARNINGS.
3. Use case ends.

Exception flow	
N/A	
Post conditions	
	Successful conditions
	The operation is a success if listPendingMessagesResponse contains all the messageIDs of the pending messages to be downloaded or the list is empty in the case that there are no pending messages.

Failure Conditions

N/A

Special requirements

N/A

Security

When Access Point C3 is configured to require authentication, Backend C4 must authenticate itself to get the status of the message.
There are two possible roles ROLE_USER and ROLE_ADMIN.

In case C4 authenticates with the ROLE_USER, pending messages will be filtered by the authenticated user (matching the property “finalRecipient”)

ROLE_ADMIN lists all the pending messages sent to this plugin.

Example of the request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:_1="http://org.ecodex.backend/1_1/">
  <soap:Header/>
  <soap:Body>     <_1:listPendingMessagesRequest></_1:listPendingMessagesRequest>
  </soap:Body>
</soap:Envelope>
```

Example of the response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns6:listPendingMessagesResponse xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/" xmlns:eb="http://docs.oasis-open.org/ebx
msg/ebms/v3.0/ns/core/200704/" xmlns:S12="http://www.w3.org/2003/05/soap-envelope" xmlns:xmime="http://www.w3.org/2005/05/xr
xmlns:ns6="http://org.ecodex.backend/1_1/">
      <messageID>4078cfea-74e9-4058-9d14-1dceee597abd@domibus.eu</messageID>
    </ns6:listPendingMessagesResponse>
  </soap:Body>
  </soap:Envelope>
```

2.2. Security

The Default WS Plugin supports following types of authentication:

- Basic Authentication

- X509Certificates Authentication - Blue Coat Authentication

Note: Blue Coat is the name of the reverse proxy at the commission. It forwards the request in HTTP with the certificate details inside the request (“Client-Cert” header key).

Basic Authentication takes precedence on both **http and https**.
X509Certificates is expected on **https** when no Basic Authentication was found.
Blue Coat certificates are expected on **http** when no Basic Authentication was found.

By default there are two users inserted in the database “admin” and “user” both with passwords “123456”.

“admin” has the role ROLE_ADMIN and “user” has the role ROLE_USER

The self signed X509Certificate of the “blue_gw" alias is also in the db with the ROLE_ADMIN.

ORIGINAL_USER is ignored for ROLE_ADMIN while for ROLE_USER it has to match the originalSender/finalRecipient in the message properties.

ROLE_ADMIN has the right to call:

- sendMessage with any value for originalSender property
- downloadMessage (any message among messages notified to this plugin)
- listPendingMessages will list all pending messages for this plugin
- getMessageStatus and getMessageErrors

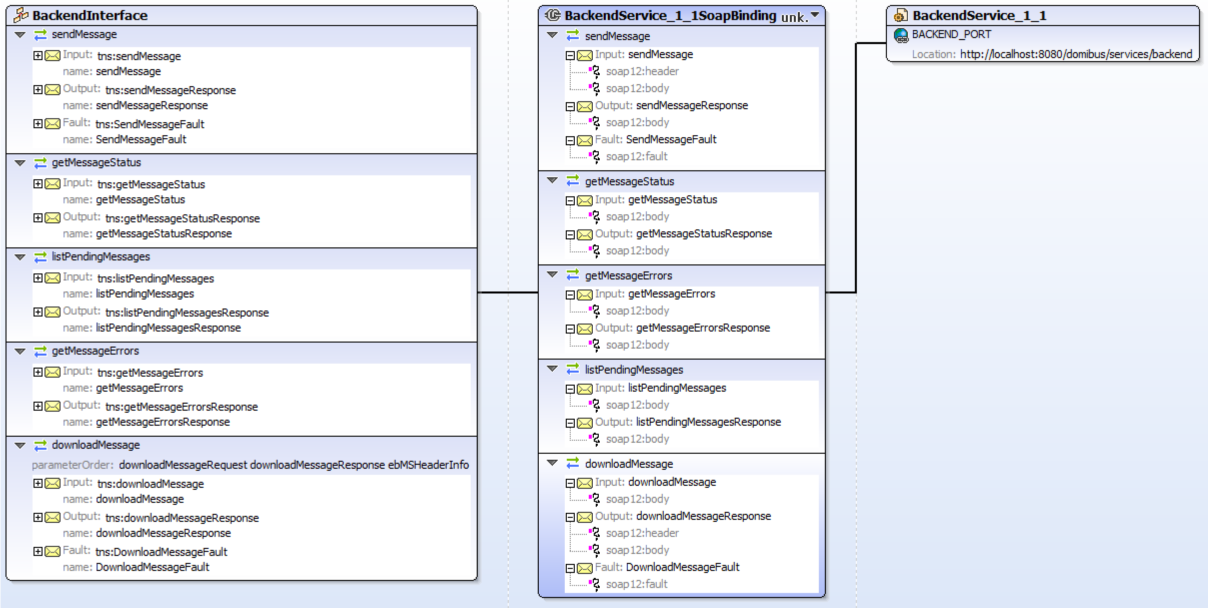
ROLE_USER has the right to call:

- sendMessage with originalSender equal to the originalUser
- downloadMessage, only if finalRecipient equals the originalUser
- listPendingMessages, only messages with finalRecipient equal to the originalUser

3. Interface Behavioural Specification

3.1. WSDL model for Domibus 3.2-RC1

3.1.1. *The WSDL schema*



BackendInterface
sendMessage
Input: tns:sendMessage
name: sendMessage
Output: tns:sendMessageResponse
name: sendMessageResponse
Fault: tns:SendMessageFault
name: SendMessageFault
getMessageStatus
Input: tns:getMessageStatus
name: getMessageStatus
Output: tns:getMessageStatusResponse
name: getMessageStatusResponse
listPendingMessages
Input: tns:listPendingMessages
name: listPendingMessages
Output: tns:listPendingMessagesResponse
name: listPendingMessagesResponse
getMessageErrors
Input: tns:getMessageErrors
name: getMessageErrors
Output: tns:getMessageErrorsResponse
name: getMessageErrorsResponse
downloadMessage
parameterOrder: downloadMessageRequest downloadMessageResponse ebMSHeaderInfo
Input: tns:downloadMessage
name: downloadMessage
Output: tns:downloadMessageResponse
name: downloadMessageResponse
Fault: tns:DownloadMessageFault
name: DownloadMessageFault

BackendService_1_1SoapBinding unk.
sendMessage
Input: sendMessage
soap12:header
soap12:body
Output: sendMessageResponse
soap12:body
Fault: SendMessageFault
soap12:fault
getMessageStatus
Input: getMessageStatus
soap12:body
Output: getMessageStatusResponse
soap12:body
getMessageErrors
Input: getMessageErrors
soap12:body
Output: getMessageErrorsResponse
soap12:body
listPendingMessages
Input: listPendingMessages
soap12:body
Output: listPendingMessagesResponse
soap12:body
downloadMessage
Input: downloadMessage
soap12:body
Output: downloadMessageResponse
soap12:header
soap12:body
Fault: DownloadMessageFault
soap12:fault

BackendService_1_1
BACKEND_PORT
Location: http://localhost:8080/domibus/services/backend

Figure 10 - WSDL model for Domibus 3.2-RC1

The WSDL defines the envelope that consists of one AP Header and one AP Body.

The service sends a message and receives a response. There are **five operations:**

- sendMessage
- getMessageStatus
- listPendingMessages
- getMessageErrors → It can be used If you get a SEND_FAILURE or REJECTED as response in the getMessageStatus service status in which case this operation can be used to get the details of the encountered errors. There can be multiple errors as each retry might produce one.
- downloadMessage

To encapsulate errors, the *fault* element is specified for only two services (sendMessage and downloadMessage):

- <wsdl:fault name="SendMessageFault"/>
- <wsdl:fault name="DownloadMessageFault"/>

It must be generated and processed according to the [SOAP1.2] specification. In this case SOAP protocol is used and the binding is **<soap:binding>**. The transport is SOAP messages on top of HTTP protocol:

transport=*"http://schemas.xmlsoap.org/soap/http"*/>

3.1.2. The data model applicable to SendMessage from C1 to C2 (domibus-submission.xsd)

In this section the data model is explained.

3.1.2.1. *Messaging/UserMessage/MessageInfo:*

This Optional element occurs once, and contains the identifier of the current message, and (may) relate to other messages' identifiers.

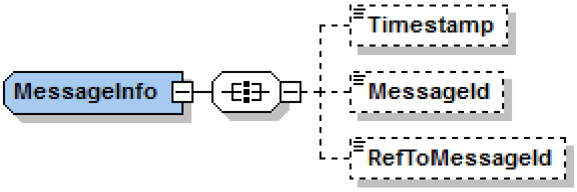


Figure 11 – MessageInfo type

- **Timestamp** element has a value representing the date at which the message header was created.

- **MessageId** has a value representing – for each message - a globally unique identifier.

- **RefToMessageId** contains the MessageId value of an ebMS Message to which this message relates, in a way that conforms to the MEP in use.

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
Timestamp	*[local-name()='schema']/*[local-name()='complexType' and @name='MessageInfo'] /*[local-name()='all']/*[local-name()='element' and @name='Timestamp']	**N**	**Max 1**	It MUST be expressed as *YYYY-MM-DDTHH:MM:SS.msmsmsZ*	*2016-03-31T09:00:44.418Z*
MessageId	*[local-name()='schema']/*[local-name()='complexType' and @name='MessageInfo'] /*[local-name()='all']/*[local-name()='element' and @name='MessageId']	**N**	**Max 1**	1. A globally unique identifier **2.** In the Message-Id and Content-Id MIME headers, values are always surrounded by angle brackets. However references in mid: or cid: scheme URI's and the MessageId and RefToMessageId elements MUST NOT include these delimiters. 3. Max length: value should not be more than 255 characters.	*346ea37f-7583-40b0-9ffc-3f4cfa88bf8b@domibus.eu*
RefToMessag	/*[local-	**N**	**Max 1**	1. A globally unique identifier.	*346ea37f-7583-*

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
eId	name()='schema']/*[local-name()='complexType' and @name='MessageInfo'] /*[local-name()='all']/*[local-name()='element' and @name='RefToMessageId']			**2.** In the Message-Id and Content-Id MIME headers, values are always surrounded by angle brackets. However references in mid: or cid: scheme URI's and the MessageId and RefToMessageId elements MUST NOT include these delimiters. 3. Max length: value should not be more than 255 characters.	*40b0-9ffc-3f4cfa88bf8b@domibus.eu*

3.1.2.2. *Messaging/UserMessage/PartyInfo*

This REQUIRED element occurs once, and contains data about originating and destination parties. This element has the following children elements:

- **From:** This REQUIRED element occurs once, and contains information describing the originating party. It can be either endpoint C1 or endpoint C2.
- **To:** This REQUIRED element occurs once, and contains information describing the destination party and it can be either endpoint C3 or endpoint C4.

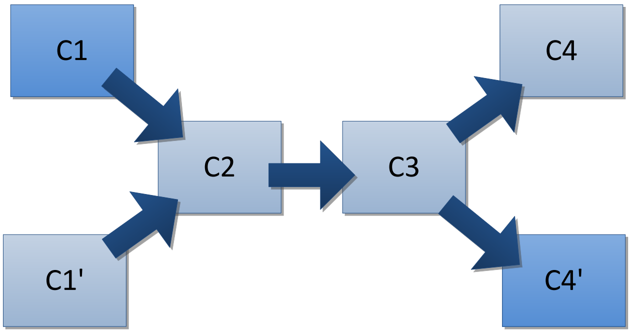


Figure 12 - The From - To PartyInfo

If the From and To are C1, C1' and C4, C4' respectively, the private keys of the certificates of C1 and C1' are stored in C2 and the public keys of the certificates of C4 and C4' are stored in C3. But if the From and To are C2 and C3, the private key of the certificate of C2 is stored in C2 and the public key of C3 is stored in C3.

From	To	Private key of	Private key stored in	Public key of	Public Key stored in
C1, C1'	C4, C4'	C1, C1'	C2	C4,C4'	C3
C2	C3	C2	C2	C3	C3

- **Role:** This REQUIRED element identifies the authorized role of the Party sending or receiving the message.
- **Type:** This element indicates the domain of names to which the string in the content of the PartyId element belongs.

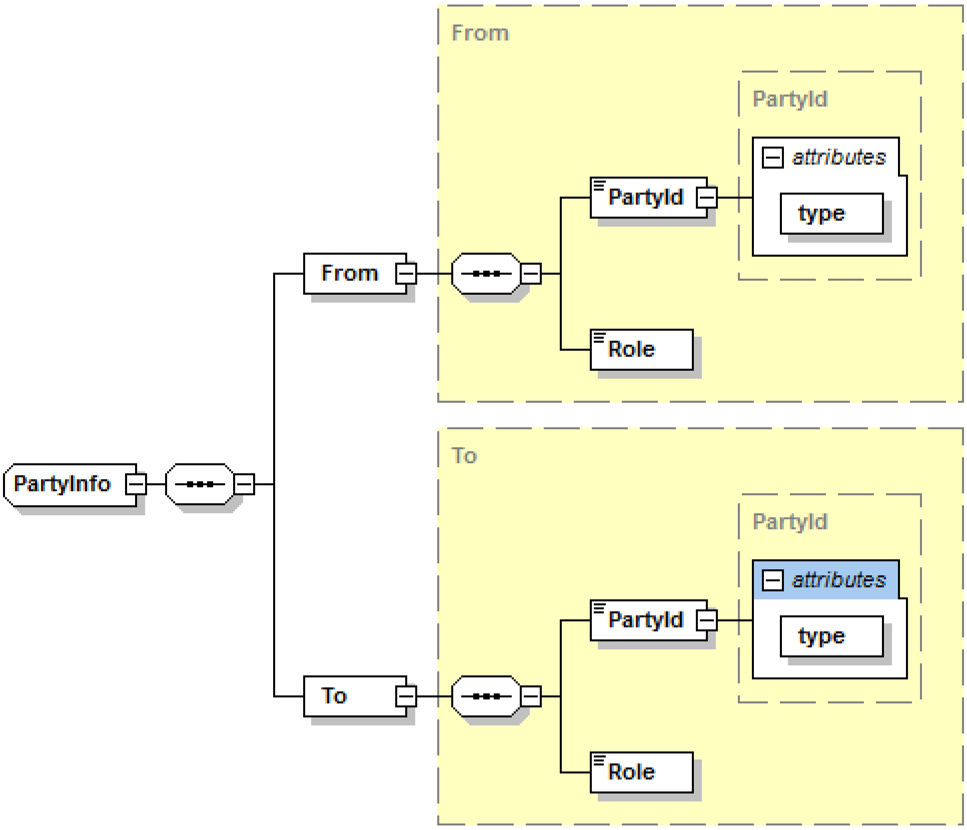


Figure 13 – PartyInfo type

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
From/PartyId	/*[local-name()='schema']/*[local-name()='complexType' and @name='From'] /*[local-name()='all'] /*[local-name()='element' and @name='PartyId']	Y	Max 1	1. The content of the PartyId element MUST be a URI if Type is not used. 2. The PartyID should be the same that is used in the PMode configuration: **3.** Max length:255 characters 4. Configuration in PMode: PMode.Initiator.Party	*C2*
From/Role	/*[local-name()='schema']/*[local-name()='complexType' and @name='From'] /*[local-name()='all'] /*[local-name()='element' and	Y	Max 1	1. It is a non-empty string, 2. Max length:255 characters 3. Configuration in PMode: PMode.Initiator.Role	*http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator*

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
	@name='Role']				
To/PartyId	/*[local-name()='schema']/*[local-name()='complexType' and @name='To'] /*[local-name()='all'] /*[local-name()='element' and @name='PartyId']	**Y**	**Max 1**	1. The content of the PartyId element MUST be a URI if PartyType is not used. 2. The PartyID should be the same that is used in the PMode configuration. **3.** Max length:255 characters 4. Configuration in PMode: PMode.Responder.Party	*C3*
To/Role	/*[local-name()='schema']/*[local-name()='complexType' and @name='To'] /*[local-name()='all'] /*[local-name()='element' and @name='Role']	**Y**	**Max 1**	1. It is a non-empty string, 2. Max length:255 characters 3. Configuration in PMode: PMode.Responder.Role	*http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder*
From-To/PartyType	/*[local-name()='schema']/*[local-name()='complexType' and @name='PartyId']//*[local-name()='attribute' and @name='type']	**Y**	**Max 1**	1. It is a non-empty string, 2. Max length:255 characters	*urn:oasis:names:tc:ebcore:partyid-type:unregistered*

*3.1.2.3. **Messaging/UserMessage/CollaborationInfo***

This REQUIRED element occurs once, and contains elements that facilitate collaboration between parties.

- The **AgreementRef** element is a string that identifies the entity or artifact governing the exchange of messages between the parties.

- **Service** SHOULD identify a set of related business transactions or other message exchanges in the context of a business process or use case.

- **Action** SHOULD identify the different types of business transactions or other message exchanges in the context of an identified Service.

- **ConversationId** element is a string identifying the set of related messages that make up a conversation between Parties. So, as defined in the eSens Specifications Library, it provides a more general way to associate a message with an ongoing conversation, without requiring a message to be a response to a single specific previous message, but allowing update messages to existing conversations from both Sender and Receiver of the original message.

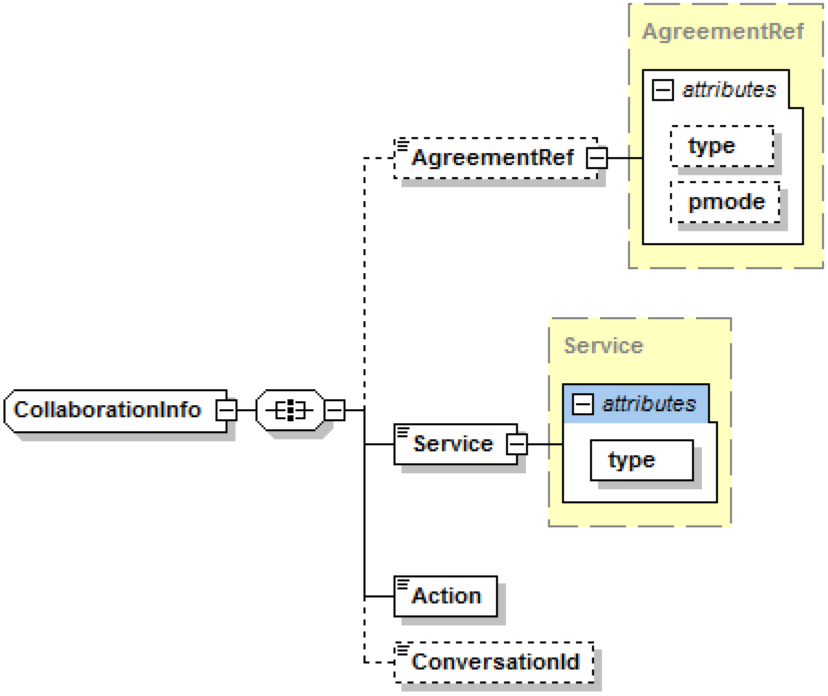


Figure 14 – CollaborationInfo type

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
AgreementRef	/*[local-name()='schema']/*[local-name()='complexType' and @name='CollaborationInfo'] /*[local-name()='all'] /*[local-name()='element' and @name='AgreementRef']	N	Max 1	1. It is a non-empty string. 2. The value of an AgreementRef element MUST be unique within a namespace mutually agreed by the two parties. This could be a concatenation of the From and To PartyId values, a URI containing the Internet domain name of one of the parties, or a namespace offered and managed by some other naming or registry service. It is RECOMMENDED that the AgreementRef be a URI. 3. AgreementRef is a string value that identifies the agreement that governs the exchange. The P-Mode under which the MSH operates for this message should be aligned with this agreement. 4. Max length:255 characters	*https://joinup.ec.europa.eu/*
agreementRef@type	/*[local-name()='schema']/*[local-name()='complexType' and @name='AgreementRef']/*[local-name()='simpleContent']	N	Max 1	1. It is a non-empty string. 2. Max length:255 characters 3. Indicates how the parties sending and receiving the message will interpret the value of the reference. There is no restriction on the value of the type attribute. 4. If the type attribute is not present, the content of the AgreementRef element MUST be a URI.	*MyServiceTypes*

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
	/*[local-name()='extension']/*[local-name()='attribute' and @name='type']				
agreementRef@pmode	/*[local-name()='schema']/*[local-name()='complexType' and @name='AgreementRef']/*[local-name()='simpleContent']/*[local-name()='extension']/*[local-name()='attribute' and @name='type']	**N**	**Max 1**	1. It is a non-empty string. 2. Max length:255 characters 3. Allows for explicit association of a message with a P-Mode. When used, its value contains the PMode.ID parameter (i.e. the identifier for the P-Mode. This identifier is user-defined and optional, for the convenience of P-Mode management. It must uniquely identify the P-Mode among all P-Modes deployed on the same AP, and may be absent if the P-Mode is identified by other means, e.g. embedded in a larger structure that is itself identified, or has parameter values distinct from other P-Modes used on the same AP. If the ID is specified, the AgreementRef/@pmode attribute value is also expected to be set in associated messages.).	*PurchaseOrderFromACME*
Service	/*[local-name()='schema']/*[local-name()='complexType' and @name='CollaborationInfo']/*[local-name()='all']/*[local-name()='element' and @name='Service']	**Y**	**Max 1**	4. It is a non-empty string, 5. Max length:255 characters 6. Configuration in PMode: PMode[1].BusinessInfo.Service	*SupplierOrderProcessing*
Service@Type	/*[local-name()='schema']/*[local-name()='complexType' and @name='Service']/*[local-name()='simpleContent']//*[local-name()='extension']/*[local-name()='attribute']	**Y**		1. Indicates how the parties sending and receiving the message will interpret the value of the element. 2. It is a non-empty string, 3. Max length:255 characters	
Action	/*[local-name()='schema']/*[loca	**Y**	**Max 1**	1. It is a non-empty string. 2. Action SHALL be unique within the Service in which it is defined.	*NewOrder*

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
	l-name()='complexType' and @name='CollaborationInfo'] /*[local-name()='all'] /*[local-name()='element' and @name='Action']			3. Max length:255 characters 4. Configuration in PMode: PMode[1].BusinessInfo.Action	
ConversationId	/*[local-name()='schema']/*[local-name()='complexType' and @name='CollaborationInfo'] /*[local-name()='all'] /*[local-name()='element' and @name='ConversationId']	**N**	**Max 1**	1. It is a non-empty string. Represents an immutable universally unique identifier (UUID). A UUID represents a 128-bit value. 2. Created randomly in the Receiving Access Point C2 3. Max length:36 characters	*06689621-428e-48a4-86e6-4a86539363f5*

3.1.2.4. *MessageProperties*

This REQUIRED element occurs at most once, and contains message properties that are implementation specific. As parts of the header such properties allow for more efficient monitoring, correlating, dispatching and validating functions (even if these are out of scope of ebMS specification) which would otherwise require payload access.

These elements hold a set of name-value properties that will hold for instance the identifiers for the 'originalSender' and 'finalRecipient', as in the example below:

```
<ns:MessageProperties>
   <ns:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C1</ns:Property>
   <ns:Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C4</ns:Property>
</ns:MessageProperties>
```

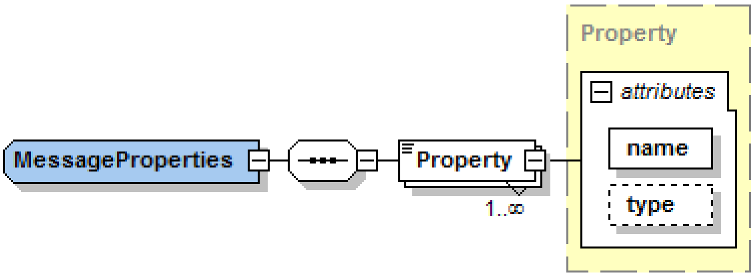


Figure 15 – MessageProperties type

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
Property name	/*[local-name()='schema']/*[local-name()='complexType' and @name='Property']//*[local-name()='attribute' and @name='name']	Y	Max 1	1. It is a non-empty string. 2. Max length:255 characters 3. Configuration in PMode: PMode[1].BusinessInfo.Properties	*originalSender*
Property type	/*[local-name()='schema']/*[local-name()='complexType' and @name='Property']//*[local-name()='attribute' and @name='type']	N	Max 1	1. It is a non-empty string	*String*

*3.1.2.5. **PayloadInfo***

This REQUIRED element occurs at most once, and identifies payload data associated with the message, whether included as part of the message as payload document(s) contained in a Payload Container, or remote resources accessible via a URL. The purpose of the PayloadInfo is to make it easier to directly extract a particular payload associated with this User message and to allow an application to determine whether it can process the payload without having to parse it. The PartInfo element is used to reference a MIME attachment, an XML element within the SOAP Body, or another resource which may be obtained by resolving a URL, according to the value of the href attribute.

- **href:** This OPTIONAL attribute has a value that is the Content-ID URI of the payload object referenced, an xml:id fragment identifier, or the URL of an externally referenced resource; The absence of the attribute href in the element PartInfo indicates that the payload part being referenced is the SOAP Body element itself.
- **Schema:** This OPTIONAL attribute refers to schema(s) that define the instance document identified in the parent PartInfo element. If the item being referenced has schema(s) of some kind that describe it (e.g. an XML Schema, DTD and/or a database schema), then the Schema

element SHOULD be present as a child of the PartInfo element. It provides a means of identifying the schema and its version defining the payload object identified by the parent PartInfo element. This metadata MAY be used to validate the Payload Part to which it refers, but the AP is NOT REQUIRED to do so.

- **Description:** This OPTIONAL attribute describes the Partinfo,
- **PartProperties:** has a REQUIRED @name attribute, value which must be agreed per implementation.

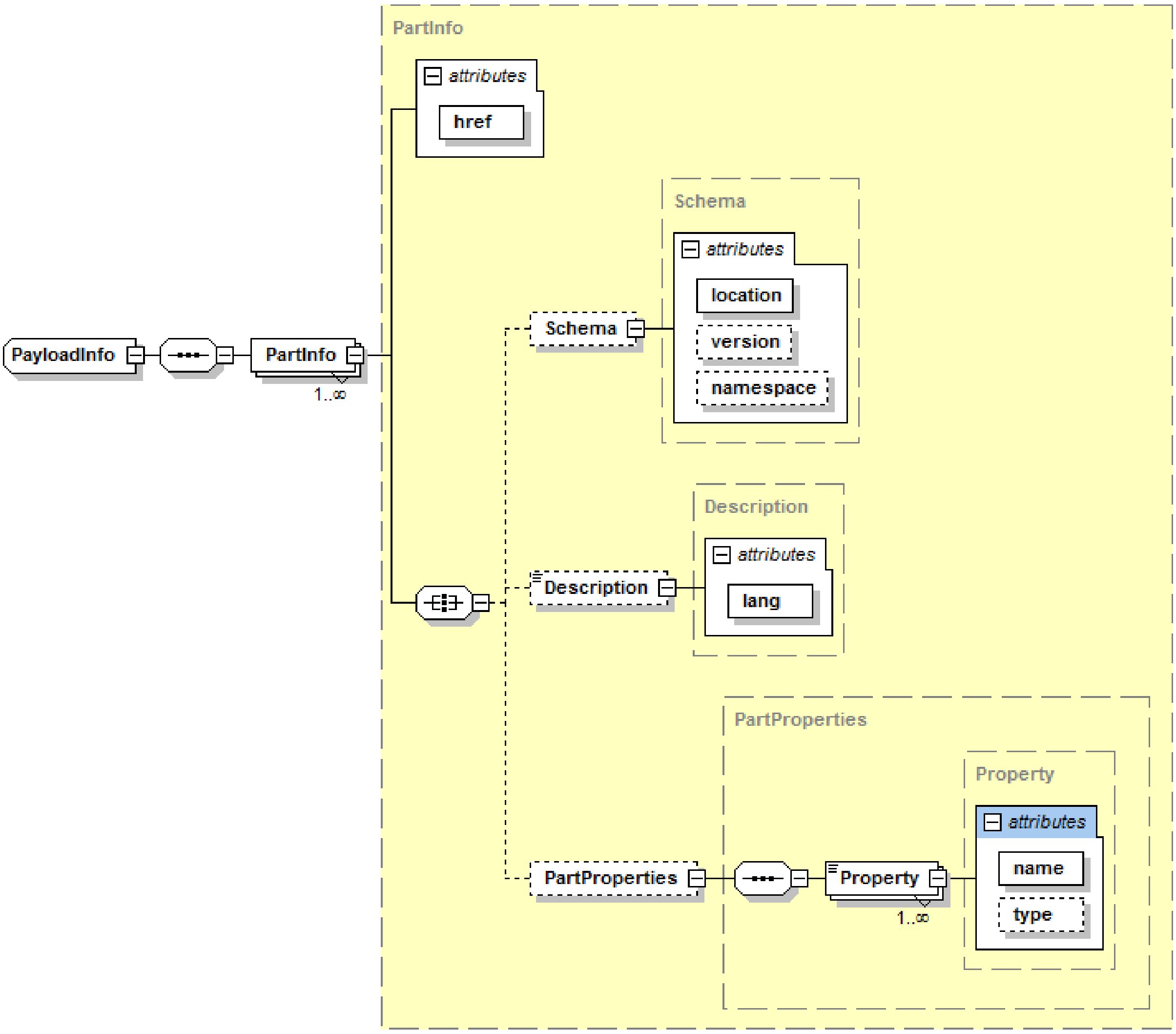


Figure 16 – PayloadInfo type

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
href	/*[local-name()='schema']/*[local-name()='complexType' and @name='PartInfo'] /*[local-name()='attribute' and	N	Max 1	1. Max length:255 characters	*cid:message*

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
	@name='href']				
Schema/location	/*[local-name()='schema']/*[local-name()='complexType' and @name='Schema']/*[local-name()='attribute' and @name='location']	**Y**	**Max 1**	1. URI	*http://localhost:8080/domibus/services/backend*
Schema/version	/*[local-name()='schema']/*[local-name()='complexType' and @name='Schema']/*[local-name()='attribute' and @name='version']	**N**	**Max 1**	1. Max length:255 characters	*1.0*
Schema/namespace	/*[local-name()='schema']/*[local-name()='complexType' and @name='Schema']/*[local-name()='attribute' and @name='version']	**N**	**Max 1**	2. Max length:255 characters	*http://www.w3.org/XML/1998/namespace*
Description	/*[local-name()='schema']/*[local-name()='complexType' and @name='PartInfo']/xsd:all/*[local-name()='element' and @name='Schema']	**N**	**Max 1**	1. Max length:255 characters	*Any short text of maximum 255 characters in the language specified by the lan'g attribute*
PartProperties/ Property/MimeType	/*[local-name()='schema']/*[local-name()='complexType' and @name='Pa	**N**	**Max 1**	1. Max length:255 characters 2. If the PMode compression is enabled this field is mandatory	*application/xml*

Description	Field (xpath)	Mandatory	Occurrences	Constraints	Valid example
	rtProperties'] /*[local-name()='sequence']/*[local-name()='element' and @name='Property']				
PartProperties/ Property/Description	/*[local-name()='schema']/*[local-name()='complexType' and @name='PartProperties'] /*[local-name()='sequence']/*[local-name()='element' and @name='Property']	**N**	**Max 1**	1. Max length:255 characters	*Message Payload*

4. ANNEXES

4.1. Interface Message standards

AS4 does not define a maximum message size, though implementations will have practical limits based on available memory, disk or database storage etc.

4.1.1. Errors codes table

Ebms error codes contained in the backend.wsdl:

Example:

```
<eb:Error origin="ebMS" category="Unpackaging"
shortDescription="InvalidHeader"
errorCode="EBMS:0009" severity="fatal">
<eb:Description xml:lang="en"> … </eb:Description>
</eb:Error>
```

Error Code	Short Description	Recommended Severity	Category Value	Description or Semantics
EBMS_0001	ValueNotRecognized	failure	Content	Although the message document is well formed and schema valid, some element/attribute contains a value that could not be recognized and therefore could not be used by the MSH.
EBMS_0002	FeatureNotSupported	warning	Content	Although the message document is well formed and schema valid, some element/attribute value cannot be processed as expected because the related feature is not supported by the MSH.
EBMS_0003	ValueInconsistent	failure	Content	Although the message document is well formed and schema valid, some element/attribute value is inconsistent either with the content of other element/attribute, or with the processing mode of the MSH, or with the normative requirements of the ebMS specification.
EBMS_0004	Other	failure	Content	
EBMS_0005	ConnectionFailure	failure	Communication	The MSH is experiencing temporary or permanent failure in trying to open a transport connection with a remote MSH.
EBMS_0006	EmptyMessagePartitionChannel	warning	Communication	There is no message available for pulling from this MPC at this moment.
EBMS_0007	MimeInconsistency	failure	Unpackaging	The use of MIME is not consistent with the required usage in this specification.
EBMS_0008	FeatureNotSupported	failure	Unpackaging	Although the message document is well formed and schema valid, the presence or absence of some element/ attribute is not consistent with the capability of the MSH, with respect to supported features.
EBMS_0009	InvalidHeader	failure	Unpackaging	The ebMS header is either not well formed as an XML document, or does not conform to the ebMS packaging rules.
EBMS_0010	ProcessingModeMismatch	failure	Processing	The ebMS header or another header (e.g. reliability, security) expected by the MSH is not compatible with the expected content, based on the associated P-Mode.

Error Code	Short Description	Recommended Severity	Category Value	Description or Semantics
EBMS_0011	ExternalPayloadError	failure	Content	The MSH is unable to resolve an external payload reference (i.e. a Part that is not contained within the ebMS Message, as identified by a PartInfo/href URI).
EBMS_0101	FailedAuthentication	failure	Processing	The signature in the Security header intended for the "ebms" SOAP actor, could not be validated by the Security module.
EBMS_0102	FailedDecryption	failure	Processing	The encrypted data reference the Security header intended for the "ebms" SOAP actor could not be decrypted by the Security Module.
EBMS_0103	PolicyNoncompliance	failure	Processing	The processor determined that the message's security methods, parameters, scope or other security policy-level requirements or agreements were not satisfied.
EBMS_0201	DysfunctionalReliability	failure	Processing	Some reliability function as implemented by the Reliability module, is not operational, or the reliability state associated with this message sequence is not valid.
EBMS_0202	DeliveryFailure	failure	Communication	Although the message was sent under Guaranteed delivery requirement, the Reliability module could not get assurance that the message was properly delivered, in spite of resending efforts.
EBMS_0301	MissingReceipt	failure	Communication	A Receipt has not been received for a message that was previously sent by the MSH generating this error
EBMS_0302	InvalidReceipt	failure	Communication	A Receipt has been received for a message that was previously sent by the MSH generating this error, but the content does not match the message content (e.g. some part has not been acknowledged, or the digest associated does not match the signature digest, for NRR).
EBMS_0303	DecompressionFailure	failure	Communication	An error occurred during the decompression
EBMS_0020	RoutingFailure	failure	Processing	An Intermediary MSH was unable to route an ebMS message and stopped processing the message.
EBMS_0021	MPCCapacityExceeded	failure	Processing	An entry in the routing function is matched that assigns the message to an MPC for pulling, but the intermediary MSH is unable to store the message with this MPC
EBMS_0022	MessagePersistenceTimeout	failure	Processing	An intermediary MSH has assigned the message to an MPC for pulling and has successfully stored it. However the intermediary set a limit on the time it was prepared to wait for the message to be pulled, and that limit has been reached.
EBMS_0023	MessageExpired	warning	Processing	An MSH has determined that the message is expired and will not attempt to forward or deliver it.
EBMS_0030	BundlingError	failure	Content	The structure of a received bundle is not in accordance with the bundling rules.
EBMS_0031	RelatedMessageFailed	failure	Processing	A message unit in a bundle was not processed because a related message unit in the bundle caused an error.
EBMS_0040	BadFragmentGroup	failure	Content	A fragment is received that relates to a group that was previously rejected.
EBMS_0041	DuplicateMessageSize	failure	Content	A fragment is received but more than one fragment message in a group of fragments specifies a value for this element.
EBMS_0042	DuplicateFragmentCount	failure	Content	A fragment is received but more than one fragment message in a group of fragments specifies a value for this element.

Error Code	Short Description	Recommended Severity	Category Value	Description or Semantics
EBMS_0043	DuplicateMessageHeader	failure	Content	A fragment is received but more than one fragment message in a group of fragments specifies a value for this element.
EBMS_0044	DuplicateAction	failure	Content	A fragment is received but more than one fragment message in a group of fragments specifies a value for this element.
EBMS_0045	DuplicateCompressionInfo	failure	Content	A fragment is received but more than one fragment message in a group of fragments specifies a value for a compression element.
EBMS_0046	DuplicateFragment	failure	Content	A fragment is received but a previously received fragment message had the same values for GroupId and FragmentNum
EBMS_0047	BadFragmentStructure	failure	Unpackaging	The href attribute does not reference a valid MIME data part, MIME parts other than the fragment header and a data part are in the message. are added or the SOAP Body is not empty.
EBMS_0048	BadFragmentNum	failure	Content	An incoming message fragment has a a value greater than the known FragmentCount.
EBMS_0049	BadFragmentCount	failure	Content	A value is set for FragmentCount, but a previously received fragment had a greature value.
EBMS_0050	FragmentSizeExceeded	warning	Unpackaging	The size of the data part in a fragment message is greater than **Pmode[].Splitting.FragmentSize**
EBMS_0051	ReceiveIntervalExceeded	failure	Unpackaging	More time than **Pmode[].Splitting.JoinInterval** has passed since the first fragment was received but not all other fragments are received.
EBMS_0052	BadProperties	warning	Unpackaging	Message properties were present in the fragment SOAP header that were not specified in **Pmode[].Splitting.RoutingProperties**
EBMS_0053	HeaderMismatch	failure	Unpackaging	The eb3:Message header copied to the fragment header does not match the eb3:Message header in the reassembled source message.
EBMS_0054	OutOfStorageSpace	failure	Unpackaging	Not enough disk space available to store all (expected) fragments of the group.
EBMS_0055	DecompressionError	failure	Processing	An error occurred while decompressing the reassembled message.
EBMS_0060	ResponseUsingAlternateMEP	Warning	Processing	A responding MSH indicates that it applies the alternate MEP binding to the response message.
EBMS_0065	InvalidXML	failure	Content	The XML could not be validated agains the corresponding xsd.

4.1.2. The PMode configuration file

```
<?xml version="1.0" encoding="UTF-8"?>
<db:configuration xmlns:db="http://domibus.eu/configuration" party="blue_gw">

    <mpcs>
        <mpc name="defaultMpc"
             qualifiedName="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/defaultMPC"
             enabled="true"
             default="true"
             retention_downloaded="0"
             retention_undownloaded="0"/>
    </mpcs>
    <businessProcesses>
        <roles>
            <role name="defaultInitiatorRole"
                    value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator"/>
            <role name="defaultResponderRole"
                    value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder"/>
        </roles>
        <parties>
            <partyIdTypes>
                <partyIdType name="partyTypeUrn" value="urn:oasis:names:tc:ebcore:partyid-type:unregistered"/>
            </partyIdTypes>
            <party name="red_gw"
                  endpoint="http://<red_hostname>:8080/domibus/services/msh"
                    allowChunking="false"
>
                <identifier partyId="domibus-red" partyIdType="partyTypeUrn"/>
            </party>
            <party name="blue_gw"
                    endpoint="http://<blue_hostname>:8080/domibus/services/msh"
                    allowChunking="false"
>
                <identifier partyId="domibus-blue" partyIdType="partyTypeUrn"/>
            </party>
        </parties>
        <meps>
            <mep name="oneway" value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/oneWay"/>
            <mep name="twoway" value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/twoWay"/>
            <binding name="push" value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push"/>
            <binding name="pushAndPush" value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/push-and-push"/>
        </meps>
        <properties>
            <property name="originalSenderProperty"
                    key="originalSender"
                    datatype="string"
                    required="true"/>
            <property name="finalRecipientProperty"
                    key="finalRecipient"
                    datatype="string"
                    required="true"/>
            <propertySet name="ecodexPropertySet">
                <propertyRef property="finalRecipientProperty"/>
                <propertyRef property="originalSenderProperty"/>
            </propertySet>
        </properties>
        <payloadProfiles>
            <payload name="businessContentPayload"
                    cid="cid:message"
                    required="true"
                    mimeType="text/xml"/>
            <payload name="businessContentAttachment"
                    cid="cid:attachment"
                    required="false"
                    mimeType="application/octet-stream"/>
            <payloadProfile name="MessageProfile"
                    maxSize="40894464">
                <attachment name="businessContentPayload"/>
                <attachment name="businessContentAttachment"/>
            </payloadProfile>
        </payloadProfiles>
        <securities>
            <security name="eDeliveryPolicy"
                    policy="eDeliveryPolicy.xml"
                    signatureMethod="RSA_SHA256" />
            <security name="noSigNoEnc"
                    policy="doNothingPolicy.xml"
                    signatureMethod="RSA_SHA256" />
        </securities>
        <errorHandlings>
            <errorHandling name="demoErrorHandling"
                    errorAsResponse="true"
                    businessErrorNotifyProducer="false"
                    businessErrorNotifyConsumer="false"
                    deliveryFailureNotifyProducer="false"/>
        </errorHandlings>
        <agreements>
            <agreement name="agreementEmpty" value="" type=""/>
        </agreements>
        <services>
            <service name="testService1" value="bdx:noprocess" type="tc1"/>
        </services>
        <actions>
            <action name="tc1Action" value="TC1Leg1"/>
        </actions>
        <as4>
            <receptionAwareness name="receptionAwareness" retry="12;4;CONSTANT" duplicateDetection="true"/>
            <reliability name="AS4Reliability" nonRepudiation="true" replyPattern="response"/>
            <reliability name="noReliability" nonRepudiation="false" replyPattern="response"/>
        </as4>
        <legConfigurations>
            <legConfiguration name="pushTestcase1tc1Action"
                    service="testService1"
                    action="tc1Action"
                    defaultMpc="defaultMpc"
                    reliability="AS4Reliability"
                    security="eDeliveryPolicy"
                    receptionAwareness="receptionAwareness"
                    propertySet="ecodexPropertySet"
                    payloadProfile="MessageProfile"
                    errorHandling="demoErrorHandling"
                    compressPayloads="true"/>
        </legConfigurations>
<process name="tc1Process"
        agreement="agreementEmpty"
        mep="oneway"
        binding="push"
        initiatorRole="defaultInitiatorRole"
        responderRole="defaultResponderRole">
    <initiatorParties>
        <initiatorParty name="blue_gw"/>
        <initiatorParty name="red_gw"/>
    </initiatorParties>
    <responderParties>
        <responderParty name="blue_gw"/>
        <responderParty name="red_gw"/>
    </responderParties>
    <legs>
        <leg name="pushTestcase1tc1Action"/>
    </legs>
</process>
    </businessProcesses>
</db:configuration>
```

4.1.3. The PMode validation file (xsd)

This file validates the domibus configuration files:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!--
 ~ Copyright 2015 e-CODEX Project
 ~
 ~ Licensed under the EUPL, Version 1.1 or – as soon they
 ~ will be approved by the European Commission - subsequent
 ~ versions of the EUPL (the "Licence");
 ~ You may not use this work except in compliance with the
 ~ Licence.
 ~ You may obtain a copy of the Licence at:
 ~ http://ec.europa.eu/idabc/eupl5
 ~ Unless required by applicable law or agreed to in
 ~ writing, software distributed under the Licence is
 ~ distributed on an "AS IS" basis,
 ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
 ~ express or implied.
 ~ See the Licence for the specific language governing
 ~ permissions and limitations under the Licence.
 -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://domibus.eu/configuration"
targetNamespace="http://domibus.eu/configuration">
    <xs:element name="configuration">
        <xs:complexType>
            <xs:all>
                <xs:element name="mpcs">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="mpc" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:attribute name="name" type="xs:string" use="required"/>
                                    <xs:attribute name="retention_downloaded" type="xs:integer" use="required"/>
                                    <xs:attribute name="retention_undownloaded" type="xs:integer" use="required"/>
                                    <xs:attribute name="default" type="xs:boolean" use="required"/>
                                    <xs:attribute name="enabled" type="xs:boolean" use="required"/>
                                    <xs:attribute name="qualifiedName" type="xs:string" use="required"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                    <xs:unique name="uniqueMpcName">
                        <xs:selector xpath="mpc"/>
                        <xs:field xpath="@name"/>
                    </xs:unique>
                    <xs:unique name="newUnique1">
                        <xs:selector xpath="mpc"/>
                        <xs:field xpath="@qualifiedName"/>
                    </xs:unique>
                </xs:element>
                <xs:element name="businessProcesses">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element name="roles">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="role" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="value" type="xs:string" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                                <xs:unique name="uniqueRoleName">
                                    <xs:selector xpath="role"/>
                                    <xs:field xpath="@name"/>
                                </xs:unique>
                            </xs:element>
                            <xs:element name="parties">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="partyIdTypes">
```

```
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="partyIdType" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:attribute name="name" type="xs:string" use="required"
id="partyIdType"/>
                                                            <xs:attribute name="value" type="xs:anyURI" use="required"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                            <xs:unique name="uniquePartyIdType">
                                                <xs:selector xpath="partyIdType"/>
                                                <xs:field xpath="name"/>
                                            </xs:unique>
                                            <xs:unique name="uniquePartyIdTypeValue">
                                                <xs:selector xpath="partyIdType"/>
                                                <xs:field xpath="value"/>
                                            </xs:unique>
                                        </xs:element>
                                        <xs:element name="party" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="identifier" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:attribute name="partyId" type="xs:string" use="required"/>
                                                            <xs:attribute name="partyIdType" type="xs:string" use="required"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="userName" type="xs:string"/>
                                                <xs:attribute name="password" type="xs:string"/>
                                                <xs:attribute name="endpoint" type="xs:anyURI" use="required"/>
                                                <xs:attribute name="allowChunking" type="xs:string" use="optional"/>
                                                <!-- NOT SUPPORTED YET -->
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="meps">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="mep" form="unqualified" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="value" type="xs:anyURI" use="required"/>
                                                <xs:attribute name="legs" type="xs:integer" default="1"/>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="binding" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="value" type="xs:anyURI" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="properties">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="property" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="key" type="xs:string" use="required"/>
                                                <xs:attribute name="datatype" type="xs:string" use="required" form="unqualified"/>
                                                <xs:attribute name="required" type="xs:boolean" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
```

```
                    <xs:element name="propertySet" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="propertyRef" maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:attribute name="property" type="xs:string" use="required"/>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="payloadProfiles">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="payload" minOccurs="0" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="cid" type="xs:string" use="required"/>
                        <xs:attribute name="mimeType" type="xs:string"/>
                        <xs:attribute name="inBody" type="xs:string"/>
                        <xs:attribute name="schemaFile" type="xs:anyURI" form="unqualified"/>
                        <xs:attribute name="maxSize" type="xs:integer"/>
                        <xs:attribute name="required" type="xs:boolean" use="required"/>
                      </xs:complexType>
                    </xs:element>
                    <xs:element name="payloadProfile" minOccurs="0" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:sequence>
                          <xs:element name="attachment" maxOccurs="unbounded">
                            <xs:complexType>
                              <xs:attribute name="name" type="xs:string" use="required"/>
                            </xs:complexType>
                          </xs:element>
                        </xs:sequence>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="maxSize" type="xs:integer" use="required"/>
                      </xs:complexType>
                      <xs:unique name="uniqueAttachment">
                        <xs:selector xpath="attachment"/>
                        <xs:field xpath="@name"/>
                      </xs:unique>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="securities">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="security" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="policy" type="xs:string" use="required"/>
                        <xs:attribute name="signatureMethod" type="xs:string" use="required"/>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element name="errorHandlings">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="errorHandling" maxOccurs="unbounded">
                      <xs:complexType>
                        <xs:attribute name="name" type="xs:string" use="required"/>
                        <xs:attribute name="errorAsResponse" type="xs:boolean" use="required"/>
                        <xs:attribute name="businessErrorNotifyProducer" type="xs:boolean"
use="required"/>
```

```
                                                <xs:attribute name="businessErrorNotifyConsumer" type="xs:boolean"
use="required"/>
                                                <xs:attribute name="deliveryFailureNotifyProducer" type="xs:boolean"
use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="agreements">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="agreement" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="value" type="xs:string" use="required"/>
                                                <xs:attribute name="type" type="xs:string" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="services">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="service" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="value" type="xs:string" use="required"/>
                                                <xs:attribute name="type" type="xs:string" use="optional"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="actions">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="action" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="value" type="xs:string" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="as4">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="receptionAwareness" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="retry" type="xs:string"/>
                                                <xs:attribute name="duplicateDetection" type="xs:string"/>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="reliability" maxOccurs="unbounded">
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="replyPattern" type="xs:string" use="required"/>
                                                <xs:attribute name="nonRepudiation" type="xs:string" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="legConfigurations">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="legConfiguration" maxOccurs="unbounded">
```

```
                                            <xs:complexType>
                                                <xs:attribute name="name" type="xs:string" use="required"/>
                                                <xs:attribute name="service" type="xs:string" use="required"/>
                                                <xs:attribute name="action" type="xs:string" use="required"/>
                                                <xs:attribute name="security" type="xs:string" use="required"/>
                                                <xs:attribute name="defaultMpc" type="xs:string" use="required"/>
                                                <xs:attribute name="receptionAwareness" type="xs:string" use="required"/>
                                                <xs:attribute name="reliability" type="xs:string" use="required"/>
                                                <xs:attribute name="propertySet" type="xs:string"/>
                                                <xs:attribute name="payloadProfile" type="xs:string"/>
                                                <xs:attribute name="errorHandling" type="xs:string" use="required"/>
                                                <xs:attribute name="compressPayloads" type="xs:string" use="required"/>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                </xs:complexType>
                            </xs:element>
                            <xs:element name="process" maxOccurs="unbounded">
                                <xs:complexType>
                                    <xs:sequence>
                                        <xs:element name="initiatorParties">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="initiatorParty" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:attribute name="name" type="xs:string" use="required"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="responderParties">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="responderParty" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:attribute name="name" type="xs:string" use="required"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                        <xs:element name="legs">
                                            <xs:complexType>
                                                <xs:sequence>
                                                    <xs:element name="leg" maxOccurs="unbounded">
                                                        <xs:complexType>
                                                            <xs:attribute name="name" type="xs:string" use="required"/>
                                                        </xs:complexType>
                                                    </xs:element>
                                                </xs:sequence>
                                            </xs:complexType>
                                        </xs:element>
                                    </xs:sequence>
                                    <xs:attribute name="name" type="xs:string" use="required"/>
                                    <xs:attribute name="responderRole" type="xs:string" use="required"/>
                                    <xs:attribute name="agreement" type="xs:string"/>
                                    <xs:attribute name="binding" type="xs:string" use="required"/>
                                    <xs:attribute name="mep" type="xs:string" use="required"/>
                                    <xs:attribute name="initiatorRole" type="xs:string" use="required"/>
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                    </xs:complexType>
                </xs:element>
            </xs:all>
            <xs:attribute name="party" type="xs:string" use="required"/>
        </xs:complexType>
    </xs:element>
</xs:schema>
```

4.1.4. Domibus-submission.xsd

This XSD validates the request messages (UC01):

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
targetNamespace="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/" elementFormDefault="qualified"
attributeFormDefault="unqualified">
    <xsd:import namespace="http://www.w3.org/XML/1998/namespace"/>
    <xsd:annotation>
        <xsd:appinfo>Schema for Domibus messages' headers submission</xsd:appinfo>
        <xsd:documentation xml:lang="en">
  This schema defines an XML subset of ebMS-3 headers which is used to validate messages submitted to Domibus
  through WS plugin.
  </xsd:documentation>
    </xsd:annotation>
    <xsd:element name="Messaging" type="Messaging"/>
    <xsd:complexType name="Messaging">
        <xsd:sequence>
            <xsd:element name="UserMessage" type="UserMessage"/>
        </xsd:sequence>
        <xsd:attribute name="mustUnderstand" type="xsd:boolean" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="UserMessage">
        <xsd:all>
            <xsd:element name="MessageInfo" type="MessageInfo" minOccurs="0"/>
            <xsd:element name="PartyInfo" type="PartyInfo"/>
            <xsd:element name="CollaborationInfo" type="CollaborationInfo"/>
            <xsd:element name="MessageProperties" type="tns:MessageProperties"/>
            <xsd:element name="PayloadInfo" type="tns:PayloadInfo"/>
        </xsd:all>
        <xsd:attribute name="mpc" type="xsd:anyURI" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="MessageInfo">
        <xsd:all>
            <xsd:element name="Timestamp" type="xsd:dateTime" minOccurs="0"/>
            <xsd:element name="MessageId" type="tns:max255-non-empty-string" minOccurs="0"/>
            <xsd:element name="RefToMessageId" type="tns:max255-non-empty-string" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="PartyInfo">
        <xsd:all>
            <xsd:element name="From" type="tns:From"/>
            <xsd:element name="To" type="tns:To"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="PartyId">
        <xsd:simpleContent>
            <xsd:extension base="tns:max255-non-empty-string">
                <xsd:attribute name="type" type="tns:max255-non-empty-string" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="From">
        <xsd:all>
            <xsd:element name="PartyId" type="tns:PartyId"/>
            <xsd:element name="Role" type="tns:max255-non-empty-string"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="To">
        <xsd:all>
            <xsd:element name="PartyId" type="tns:PartyId"/>
            <xsd:element name="Role" type="tns:max255-non-empty-string"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="CollaborationInfo">
        <xsd:all>
            <xsd:element name="AgreementRef" type="tns:AgreementRef" minOccurs="0"/>
            <xsd:element name="Service" type="tns:Service"/>
            <xsd:element name="Action" type="xsd:token"/>
            <xsd:element name="ConversationId" type="xsd:token" minOccurs="0"/>
        </xsd:all>
    </xsd:complexType>
    <xsd:complexType name="Service">
        <xsd:simpleContent>
            <xsd:extension base="tns:max255-non-empty-string">
                <xsd:attribute name="type" type="tns:max255-non-empty-string" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="AgreementRef">
        <xsd:simpleContent>
            <xsd:extension base="tns:max255-non-empty-string">
                <xsd:attribute name="type" type="tns:max255-non-empty-string" use="optional"/>
                <xsd:attribute name="pmode" type="tns:max255-non-empty-string" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="PayloadInfo">
        <xsd:sequence>
            <xsd:element name="PartInfo" type="tns:PartInfo" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="PartInfo">
        <xsd:all>
            <xsd:element name="Schema" type="Schema" minOccurs="0"/>
            <xsd:element name="Description" type="tns:Description" minOccurs="0"/>
            <xsd:element name="PartProperties" type="tns:PartProperties" minOccurs="0"/>
        </xsd:all>
        <xsd:attribute name="href" type="xsd:token" use="required"/>
    </xsd:complexType>
    <xsd:complexType name="Property">
        <xsd:simpleContent>
            <xsd:extension base="tns:max255-non-empty-string">
                <xsd:attribute name="name" type="tns:max255-non-empty-string" use="required"/>
                <xsd:attribute name="type" type="tns:max255-non-empty-string" use="optional"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:complexType name="PartProperties">
        <xsd:sequence>
            <xsd:element name="Property" type="tns:Property" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="MessageProperties">
        <xsd:sequence>
            <xsd:element name="Property" type="Property" maxOccurs="unbounded"/>
        </xsd:sequence>
    </xsd:complexType>
    <xsd:complexType name="Schema">
        <xsd:attribute name="location" type="xsd:anyURI" use="required"/>
        <xsd:attribute name="version" type="tns:max255-non-empty-string" use="optional"/>
        <xsd:attribute name="namespace" type="tns:max255-non-empty-string" use="optional"/>
    </xsd:complexType>
    <xsd:complexType name="Description">
        <xsd:simpleContent>
            <xsd:extension base="tns:max255-non-empty-string">
                <xsd:attribute ref="xml:lang" use="required"/>
            </xsd:extension>
        </xsd:simpleContent>
    </xsd:complexType>
    <xsd:simpleType name="max255-non-empty-string">
        <xsd:restriction base="xsd:string">
            <xsd:minLength value="1"/>
            <xsd:maxLength value="255"/>
        </xsd:restriction>
    </xsd:simpleType>
</xsd:schema>
```

4.1.5. Web service WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:tns="http://org.ecodex.backend/1_1/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/" xmlns:ns1="http://docs.oasis-
open.org/ebxml-msg/ebms/v3.0/ns/core/200704/" name="BackendService_1_1" targetNamespace="http://org.ecodex.backend/1_1/">
    <wsdl:types>
        <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/" xmlns="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
attributeFormDefault="unqualified" elementFormDefault="qualified" targetNamespace="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/">
            <xsd:import namespace="http://www.w3.org/XML/1998/namespace"/>
            <xsd:annotation>
                <xsd:appinfo>Schema for Domibus messages' headers submission</xsd:appinfo>
                <xsd:documentation xmlns:xml="http://www.w3.org/XML/1998/namespace" xml:lang="en">
          This schema defines an XML subset of ebMS-3 headers which is used to validate messages submitted to
          Domibus
          through WS plugin.
        </xsd:documentation>
            </xsd:annotation>
            <xsd:element name="Messaging" type="Messaging"/>
            <xsd:complexType name="Messaging">
                <xsd:sequence>
                    <xsd:element maxOccurs="1" minOccurs="1" name="UserMessage" type="UserMessage"/>
                </xsd:sequence>
                <xsd:attribute name="mustUnderstand" type="xsd:boolean" use="optional"/>
            </xsd:complexType>
            <xsd:complexType name="UserMessage">
                <xsd:all>
                    <xsd:element maxOccurs="1" minOccurs="0" name="MessageInfo" type="MessageInfo"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="PartyInfo" type="PartyInfo"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="CollaborationInfo" type="CollaborationInfo"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="MessageProperties" type="tns:MessageProperties"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="PayloadInfo" type="tns:PayloadInfo"/>
                </xsd:all>
                <xsd:attribute name="mpc" type="xsd:anyURI" use="optional"/>
            </xsd:complexType>
            <xsd:complexType name="MessageInfo">
                <xsd:all>
                    <xsd:element maxOccurs="1" minOccurs="0" name="Timestamp" type="xsd:dateTime"/>
                    <xsd:element maxOccurs="1" minOccurs="0" name="MessageId" type="tns:max255-non-empty-string"/>
                    <xsd:element maxOccurs="1" minOccurs="0" name="RefToMessageId" type="tns:max255-non-empty-string"/>
                </xsd:all>
            </xsd:complexType>
            <xsd:complexType name="PartyInfo">
                <xsd:all>
                    <xsd:element name="From" type="tns:From"/>
                    <xsd:element name="To" type="tns:To"/>
                </xsd:all>
            </xsd:complexType>
            <xsd:complexType name="PartyId">
                <xsd:simpleContent>
                    <xsd:extension base="tns:max255-non-empty-string">
                        <xsd:attribute name="type" type="tns:max255-non-empty-string" use="required"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
            <xsd:complexType name="From">
                <xsd:all>
                    <xsd:element maxOccurs="1" minOccurs="1" name="PartyId" type="tns:PartyId"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="Role" type="tns:max255-non-empty-string"/>
                </xsd:all>
            </xsd:complexType>
            <xsd:complexType name="To">
                <xsd:all>
                    <xsd:element maxOccurs="1" minOccurs="1" name="PartyId" type="tns:PartyId"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="Role" type="tns:max255-non-empty-string"/>
                </xsd:all>
            </xsd:complexType>
            <xsd:complexType name="CollaborationInfo">
                <xsd:all>
                    <xsd:element minOccurs="0" name="AgreementRef" type="tns:AgreementRef"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="Service" type="tns:Service"/>
                    <xsd:element maxOccurs="1" minOccurs="1" name="Action" type="xsd:token"/>
                    <xsd:element maxOccurs="1" minOccurs="0" name="ConversationId" type="xsd:token"/>
                </xsd:all>
            </xsd:complexType>
            <xsd:complexType name="Service">
                <xsd:simpleContent>
                    <xsd:extension base="tns:max255-non-empty-string">
                        <xsd:attribute name="type" type="tns:max255-non-empty-string" use="required"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
            <xsd:complexType name="AgreementRef">
                <xsd:simpleContent>
                    <xsd:extension base="tns:max255-non-empty-string">
                        <xsd:attribute name="type" type="tns:max255-non-empty-string" use="optional"/>
                        <xsd:attribute name="pmode" type="tns:max255-non-empty-string" use="optional"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
            <xsd:complexType name="PayloadInfo">
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" minOccurs="1" name="PartInfo" type="tns:PartInfo"/>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="PartInfo">
                <xsd:all>
                    <xsd:element minOccurs="0" name="Schema" type="Schema"/>
                    <xsd:element maxOccurs="1" minOccurs="0" name="Description" type="tns:Description"/>
                    <xsd:element minOccurs="0" name="PartProperties" type="tns:PartProperties"/>
                </xsd:all>
                <xsd:attribute name="href" type="xsd:token" use="required"/>
            </xsd:complexType>
            <xsd:complexType name="Property">
                <xsd:simpleContent>
                    <xsd:extension base="tns:max255-non-empty-string">
                        <xsd:attribute name="name" type="tns:max255-non-empty-string" use="required"/>
                        <xsd:attribute name="type" type="tns:max255-non-empty-string" use="optional"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
            <xsd:complexType name="PartProperties">
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" name="Property" type="tns:Property"/>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="MessageProperties">
                <xsd:sequence>
                    <xsd:element maxOccurs="unbounded" name="Property" type="Property"/>
                </xsd:sequence>
            </xsd:complexType>
            <xsd:complexType name="Schema">
                <xsd:attribute name="location" type="xsd:anyURI" use="required"/>
                <xsd:attribute name="version" type="tns:max255-non-empty-string" use="optional"/>
                <xsd:attribute name="namespace" type="tns:max255-non-empty-string" use="optional"/>
            </xsd:complexType>
            <xsd:complexType name="Description">
                <xsd:simpleContent>
                    <xsd:extension base="tns:max255-non-empty-string">
                        <xsd:attribute ref="xml:lang" use="required"/>
                    </xsd:extension>
                </xsd:simpleContent>
            </xsd:complexType>
            <xsd:simpleType name="max255-non-empty-string">
                <xsd:restriction base="xsd:string">
                    <xsd:minLength value="1"/>
                    <xsd:maxLength value="255"/>
                </xsd:restriction>
            </xsd:simpleType>
        </xsd:schema>
```

```
        <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.w3.org/XML/1998/namespace"
xml:lang="en">
            <xs:annotation>
                <xs:documentation>
          See http://www.w3.org/XML/1998/namespace.html and
          http://www.w3.org/TR/REC-xml for information about this namespace.

          This schema document describes the XML namespace, in a form
          suitable for import by other schema documents.

          Note that local names in this namespace are intended to be defined
          only by the World Wide Web Consortium or its subgroups. The
          following names are currently defined in this namespace and should
          not be used with conflicting semantics by any Working Group,
          specification, or document instance:

          base (as an attribute name): denotes an attribute whose value
          provides a URI to be used as the base for interpreting any
          relative URIs in the scope of the element on which it
          appears; its value is inherited. This name is reserved
          by virtue of its definition in the XML Base specification.

          id (as an attribute name): denotes an attribute whose value
          should be interpreted as if declared to be of type ID.
          The xml:id specification is not yet a W3C Recommendation,
          but this attribute is included here to facilitate experimentation
          with the mechanisms it proposes. Note that it is _not_ included
          in the specialAttrs attribute group.

          lang (as an attribute name): denotes an attribute whose value
          is a language code for the natural language of the content of
          any element; its value is inherited. This name is reserved
          by virtue of its definition in the XML specification.

          space (as an attribute name): denotes an attribute whose
          value is a keyword indicating what whitespace processing
          discipline is intended for the content of the element; its
          value is inherited. This name is reserved by virtue of its
          definition in the XML specification.

          Father (in any context at all): denotes Jon Bosak, the chair of
          the original XML Working Group. This name is reserved by
          the following decision of the W3C XML Plenary and
          XML Coordination groups:

          In appreciation for his vision, leadership and dedication
          the W3C XML Plenary on this 10th day of February, 2000
          reserves for Jon Bosak in perpetuity the XML name
          xml:Father
        </xs:documentation>
            </xs:annotation>
            <xs:annotation>
                <xs:documentation>This schema defines attributes and an attribute group
          suitable for use by
          schemas wishing to allow xml:base, xml:lang, xml:space or xml:id
          attributes on elements they define.

          To enable this, such a schema must import this schema
          for the XML namespace, e.g. as follows:
          &lt;schema . . .>
          . . .
          &lt;import namespace="http://www.w3.org/XML/1998/namespace"
          schemaLocation="http://www.w3.org/2001/xml.xsd"/>

          Subsequently, qualified reference to any of the attributes
          or the group defined below will have the desired effect, e.g.

          &lt;type . . .>
          . . .
          &lt;attributeGroup ref="xml:specialAttrs"/>
```

```
 will define a type which will schema-validate an instance
 element with any of those attributes
</xs:documentation>
  </xs:annotation>
  <xs:annotation>
    <xs:documentation>In keeping with the XML Schema WG's standard versioning
 policy, this schema document will persist at
 http://www.w3.org/2005/08/xml.xsd.
 At the date of issue it can also be found at
 http://www.w3.org/2001/xml.xsd.
 The schema document at that URI may however change in the future,
 in order to remain compatible with the latest version of XML Schema
 itself, or with the XML namespace itself. In other words, if the XML
 Schema or XML namespaces change, the version of this document at
 http://www.w3.org/2001/xml.xsd will change
 accordingly; the version at
 http://www.w3.org/2005/08/xml.xsd will not change.
</xs:documentation>
  </xs:annotation>
  <xs:attribute name="lang">
    <xs:annotation>
      <xs:documentation>Attempting to install the relevant ISO 2- and 3-letter
  codes as the enumerated possible values is probably never
  going to be a realistic possibility. See
  RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry
  at http://www.iana.org/assignments/lang-tag-apps.htm for
  further information.

  The union allows for the 'un-declaration' of xml:lang with
  the empty string.
 </xs:documentation>
    </xs:annotation>
    <xs:simpleType>
      <xs:union memberTypes="xs:language">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value=""/>
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="space">
    <xs:simpleType>
      <xs:restriction base="xs:NCName">
        <xs:enumeration value="default"/>
        <xs:enumeration value="preserve"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
  <xs:attribute name="base" type="xs:anyURI">
    <xs:annotation>
      <xs:documentation>See http://www.w3.org/TR/xmlbase/ for
  information about this attribute.
 </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="id" type="xs:ID">
    <xs:annotation>
      <xs:documentation>See http://www.w3.org/TR/xml-id/ for
  information about this attribute.
 </xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attributeGroup name="specialAttrs">
    <xs:attribute ref="xml:base"/>
    <xs:attribute ref="xml:lang"/>
    <xs:attribute ref="xml:space"/>
  </xs:attributeGroup>
</xs:schema>
```

```
        <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://org.ecodex.backend/1_1/"
xmlns:ns1="http://www.w3.org/2005/05/xmlmime" attributeFormDefault="unqualified" elementFormDefault="unqualified"
targetNamespace="http://org.ecodex.backend/1_1/">
            <xs:import namespace="http://www.w3.org/2005/05/xmlmime"/>
            <xs:simpleType name="max255-non-empty-string">
                <xs:restriction base="xs:string">
                    <xs:minLength value="1"/>
                    <xs:maxLength value="255"/>
                </xs:restriction>
            </xs:simpleType>
            <xs:element name="FaultDetail">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="code" type="xs:string"/>
                        <xs:element name="message" nillable="true" type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="downloadMessageRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="messageID" type="tns:max255-non-empty-string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="downloadMessageResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="bodyload" type="tns:PayloadType"/>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="payload" nillable="true"
type="tns:PayloadType"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="listPendingMessagesRequest" type="xs:anyType"/>
            <xs:element name="listPendingMessagesResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="messageID" nillable="true" type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="messageErrorsRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="messageID" type="tns:max255-non-empty-string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="messageStatusRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="messageID" type="tns:max255-non-empty-string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="sendRequest">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="bodyload" type="tns:PayloadType"/>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="payload" nillable="true"
type="tns:PayloadType"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="sendRequestURL">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="bodyload" type="tns:PayloadURLType"/>
                        <xs:element maxOccurs="unbounded" minOccurs="0" name="payload" nillable="true"
type="tns:PayloadURLType"/>
```

```xml
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="sendResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element maxOccurs="unbounded" minOccurs="0" name="messageID" nillable="true" type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:complexType name="PayloadType">
        <xs:simpleContent>
            <xs:extension base="ns1:base64Binary">
                <xs:attribute name="payloadId" type="xs:token" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:complexType name="errorResultImpl">
        <xs:sequence>
            <xs:element minOccurs="0" name="errorCode" type="tns:errorCode"/>
            <xs:element minOccurs="0" name="errorDetail" type="xs:string"/>
            <xs:element minOccurs="0" name="messageInErrorId" type="xs:string"/>
            <xs:element minOccurs="0" name="mshRole" type="tns:mshRole"/>
            <xs:element minOccurs="0" name="notified" type="xs:dateTime"/>
            <xs:element minOccurs="0" name="timestamp" type="xs:dateTime"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="PayloadURLType">
        <xs:simpleContent>
            <xs:extension base="xs:string">
                <xs:attribute name="payloadId" type="xs:token" use="required"/>
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
    <xs:simpleType name="messageStatus">
        <xs:restriction base="xs:string">
            <xs:enumeration value="READY_TO_SEND"/>
            <xs:enumeration value="SEND_ENQUEUED"/>
            <xs:enumeration value="SEND_IN_PROGRESS"/>
            <xs:enumeration value="WAITING_FOR_RECEIPT"/>
            <xs:enumeration value="ACKNOWLEDGED"/>
            <xs:enumeration value="ACKNOWLEDGED_WITH_WARNING"/>
            <xs:enumeration value="SEND_ATTEMPT_FAILED"/>
            <xs:enumeration value="SEND_FAILURE"/>
            <xs:enumeration value="NOT_FOUND"/>
            <xs:enumeration value="WAITING_FOR_RETRY"/>
            <xs:enumeration value="RECEIVED"/>
            <xs:enumeration value="RECEIVED_WITH_WARNINGS"/>
            <xs:enumeration value="DELETED"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="errorCode">
        <xs:restriction base="xs:string">
            <xs:enumeration value="EBMS_0001"/>
            <xs:enumeration value="EBMS_0002"/>
            <xs:enumeration value="EBMS_0003"/>
            <xs:enumeration value="EBMS_0004"/>
            <xs:enumeration value="EBMS_0005"/>
            <xs:enumeration value="EBMS_0006"/>
            <xs:enumeration value="EBMS_0007"/>
            <xs:enumeration value="EBMS_0008"/>
            <xs:enumeration value="EBMS_0009"/>
            <xs:enumeration value="EBMS_0010"/>
            <xs:enumeration value="EBMS_0011"/>
            <xs:enumeration value="EBMS_0101"/>
            <xs:enumeration value="EBMS_0102"/>
            <xs:enumeration value="EBMS_0103"/>
            <xs:enumeration value="EBMS_0201"/>
            <xs:enumeration value="EBMS_0202"/>
            <xs:enumeration value="EBMS_0301"/>
            <xs:enumeration value="EBMS_0302"/>
```

```
                    <xs:enumeration value="EBMS_0303"/>
                    <xs:enumeration value="EBMS_0020"/>
                    <xs:enumeration value="EBMS_0021"/>
                    <xs:enumeration value="EBMS_0022"/>
                    <xs:enumeration value="EBMS_0023"/>
                    <xs:enumeration value="EBMS_0030"/>
                    <xs:enumeration value="EBMS_0031"/>
                    <xs:enumeration value="EBMS_0040"/>
                    <xs:enumeration value="EBMS_0041"/>
                    <xs:enumeration value="EBMS_0042"/>
                    <xs:enumeration value="EBMS_0043"/>
                    <xs:enumeration value="EBMS_0044"/>
                    <xs:enumeration value="EBMS_0045"/>
                    <xs:enumeration value="EBMS_0046"/>
                    <xs:enumeration value="EBMS_0047"/>
                    <xs:enumeration value="EBMS_0048"/>
                    <xs:enumeration value="EBMS_0049"/>
                    <xs:enumeration value="EBMS_0050"/>
                    <xs:enumeration value="EBMS_0051"/>
                    <xs:enumeration value="EBMS_0052"/>
                    <xs:enumeration value="EBMS_0053"/>
                    <xs:enumeration value="EBMS_0054"/>
                    <xs:enumeration value="EBMS_0055"/>
                    <xs:enumeration value="EBMS_0060"/>
                    <xs:enumeration value="EBMS_0065"/>
                </xs:restriction>
            </xs:simpleType>
            <xs:simpleType name="mshRole">
                <xs:restriction base="xs:string">
                    <xs:enumeration value="SENDING"/>
                    <xs:enumeration value="RECEIVING"/>
                </xs:restriction>
            </xs:simpleType>
            <xs:complexType final="#all" name="errorResultImplArray">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" minOccurs="0" name="item" nillable="true" type="tns:errorResultImpl"/>
                </xs:sequence>
            </xs:complexType>
            <xs:element name="getStatusRequest" nillable="true">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="messageID" type="tns:max255-non-empty-string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getMessageStatusResponse" nillable="true" type="tns:messageStatus"/>
            <xs:element name="getErrorsRequest" nillable="true">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="messageID" type="tns:max255-non-empty-string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="getMessageErrorsResponse" nillable="true" type="tns:errorResultImplArray"/>
        </xs:schema>
        <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xmime="http://www.w3.org/2005/05/xmlmime"
targetNamespace="http://www.w3.org/2005/05/xmlmime">
            <xs:attribute name="contentType">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:minLength value="3"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:attribute>
            <xs:attribute name="expectedContentTypes" type="xs:string"/>
            <xs:complexType name="base64Binary">
                <xs:simpleContent>
                    <xs:extension base="xs:base64Binary">
                        <xs:attribute ref="xmime:contentType"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
            <xs:complexType name="hexBinary">
                <xs:simpleContent>
                    <xs:extension base="xs:hexBinary">
                        <xs:attribute ref="xmime:contentType"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
        </xs:schema>
        <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://www.w3.org/2003/05/soap-envelope"
elementFormDefault="qualified" targetNamespace="http://www.w3.org/2003/05/soap-envelope" version="1.0">
            <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
            <xs:element name="Body" type="tns:Body"/>
            <xs:element name="Envelope" type="tns:Envelope"/>
            <xs:element name="Fault" type="tns:Fault"/>
            <xs:element name="Header" type="tns:Header"/>
            <xs:element name="NotUnderstood" type="tns:NotUnderstoodType"/>
            <xs:element name="Upgrade" type="tns:UpgradeType"/>
            <xs:complexType name="Fault">
                <xs:sequence>
                    <xs:element name="Code" type="tns:faultcode"/>
                    <xs:element name="Reason" type="tns:faultreason"/>
                    <xs:element minOccurs="0" name="Node" type="xs:anyURI"/>
                    <xs:element minOccurs="0" name="Role" type="xs:anyURI"/>
                    <xs:element minOccurs="0" name="Detail" type="tns:detail"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="faultcode">
                <xs:sequence>
                    <xs:element name="Value" type="xs:QName"/>
                    <xs:element minOccurs="0" name="Subcode" type="tns:subcode"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="subcode">
                <xs:sequence>
                    <xs:element name="Value" type="xs:QName"/>
                    <xs:element minOccurs="0" name="Subcode" type="tns:subcode"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="faultreason">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="Text" type="tns:reasontext"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="reasontext">
                <xs:simpleContent>
                    <xs:extension base="xs:string">
                        <xs:attribute ref="xml:lang" use="required"/>
                    </xs:extension>
                </xs:simpleContent>
            </xs:complexType>
            <xs:complexType name="detail">
                <xs:sequence>
                    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other" processContents="lax"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##other" processContents="skip"/>
            </xs:complexType>
            <xs:complexType name="Envelope">
                <xs:sequence>
                    <xs:element minOccurs="0" name="Header" type="tns:Header"/>
                    <xs:element name="Body" type="tns:Body"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##other" processContents="skip"/>
            </xs:complexType>
            <xs:complexType name="Header">
                <xs:sequence>
                    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other" processContents="lax"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##other" processContents="skip"/>
            </xs:complexType>
            <xs:complexType name="Body">
                <xs:sequence>
                    <xs:any maxOccurs="unbounded" minOccurs="0" namespace="##other" processContents="lax"/>
                </xs:sequence>
                <xs:anyAttribute namespace="##other" processContents="skip"/>
            </xs:complexType>
            <xs:complexType name="NotUnderstoodType">
                <xs:sequence/>
                <xs:attribute name="qname" type="xs:QName" use="required"/>
            </xs:complexType>
            <xs:complexType name="UpgradeType">
                <xs:sequence>
                    <xs:element maxOccurs="unbounded" name="SupportedEnvelope" type="tns:SupportedEnvType"/>
                </xs:sequence>
            </xs:complexType>
            <xs:complexType name="SupportedEnvType">
                <xs:sequence/>
                <xs:attribute name="qname" type="xs:QName" use="required"/>
            </xs:complexType>
            <xs:attribute name="mustUnderstand" type="xs:boolean"/>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="getMessageErrors">
        <wsdl:part name="getErrorsRequest" element="tns:getErrorsRequest">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="downloadMessage">
        <wsdl:part name="downloadMessageRequest" element="tns:downloadMessageRequest">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="DownloadMessageFault">
        <wsdl:part name="DownloadMessageFault" element="tns:FaultDetail">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="downloadMessageResponse">
        <wsdl:part name="downloadMessageResponse" element="tns:downloadMessageResponse">
    </wsdl:part>
        <wsdl:part name="ebMSHeaderInfo" element="ns1:Messaging">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="listPendingMessagesResponse">
        <wsdl:part name="listPendingMessagesResponse" element="tns:listPendingMessagesResponse">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="getMessageStatusResponse">
        <wsdl:part name="getMessageStatusResponse" element="tns:getMessageStatusResponse">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="listPendingMessages">
        <wsdl:part name="listPendingMessagesRequest" element="tns:listPendingMessagesRequest">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="getMessageStatus">
        <wsdl:part name="getStatusRequest" element="tns:getStatusRequest">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="sendMessageResponse">
        <wsdl:part name="sendResponse" element="tns:sendResponse">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="sendMessage">
        <wsdl:part name="sendRequest" element="tns:sendRequest">
    </wsdl:part>
        <wsdl:part name="ebMSHeaderInfo" element="ns1:Messaging">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="SendMessageFault">
        <wsdl:part name="SendMessageFault" element="tns:FaultDetail">
    </wsdl:part>
    </wsdl:message>
    <wsdl:message name="getMessageErrorsResponse">
        <wsdl:part name="getMessageErrorsResponse" element="tns:getMessageErrorsResponse">
    </wsdl:part>
    </wsdl:message>
    <wsdl:portType name="BackendInterface">
        <wsdl:operation name="sendMessage">
            <wsdl:input name="sendMessage" message="tns:sendMessage">
      </wsdl:input>
            <wsdl:output name="sendMessageResponse" message="tns:sendMessageResponse">
      </wsdl:output>
            <wsdl:fault name="SendMessageFault" message="tns:SendMessageFault">
      </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="getMessageStatus">
            <wsdl:input name="getMessageStatus" message="tns:getMessageStatus">
      </wsdl:input>
            <wsdl:output name="getMessageStatusResponse" message="tns:getMessageStatusResponse">
      </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="listPendingMessages">
            <wsdl:input name="listPendingMessages" message="tns:listPendingMessages">
      </wsdl:input>
            <wsdl:output name="listPendingMessagesResponse" message="tns:listPendingMessagesResponse">
      </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getMessageErrors">
            <wsdl:input name="getMessageErrors" message="tns:getMessageErrors">
      </wsdl:input>
            <wsdl:output name="getMessageErrorsResponse" message="tns:getMessageErrorsResponse">
      </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="downloadMessage" parameterOrder="downloadMessageRequest downloadMessageResponse
ebMSHeaderInfo">
            <wsdl:input name="downloadMessage" message="tns:downloadMessage">
      </wsdl:input>
            <wsdl:output name="downloadMessageResponse" message="tns:downloadMessageResponse">
      </wsdl:output>
            <wsdl:fault name="DownloadMessageFault" message="tns:DownloadMessageFault">
      </wsdl:fault>
        </wsdl:operation>
    </wsdl:portType>
    <wsdl:binding name="BackendService_1_1SoapBinding" type="tns:BackendInterface">
        <soap12:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
        <wsdl:operation name="sendMessage">
            <soap12:operation soapAction="" style="document"/>
            <wsdl:input name="sendMessage">
                <soap12:header message="tns:sendMessage" part="ebMSHeaderInfo" use="literal">
        </soap12:header>
                <soap12:body parts="sendRequest" use="literal"/>
            </wsdl:input>
            <wsdl:output name="sendMessageResponse">
                <soap12:body use="literal"/>
            </wsdl:output>
            <wsdl:fault name="SendMessageFault">
                <soap12:fault name="SendMessageFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
        <wsdl:operation name="getMessageStatus">
            <soap12:operation soapAction="" style="document"/>
            <wsdl:input name="getMessageStatus">
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getMessageStatusResponse">
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="getMessageErrors">
            <soap12:operation soapAction="" style="document"/>
            <wsdl:input name="getMessageErrors">
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="getMessageErrorsResponse">
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="listPendingMessages">
            <soap12:operation soapAction="" style="document"/>
            <wsdl:input name="listPendingMessages">
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="listPendingMessagesResponse">
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="downloadMessage">
            <soap12:operation soapAction="" style="document"/>
            <wsdl:input name="downloadMessage">
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output name="downloadMessageResponse">
                <soap12:header message="tns:downloadMessageResponse" part="ebMSHeaderInfo" use="literal">
        </soap12:header>
                <soap12:body parts="downloadMessageResponse" use="literal"/>
            </wsdl:output>
            <wsdl:fault name="DownloadMessageFault">
                <soap12:fault name="DownloadMessageFault" use="literal"/>
            </wsdl:fault>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="BackendService_1_1">
        <wsdl:port name="BACKEND_PORT" binding="tns:BackendService_1_1SoapBinding">
            <soap12:address location="http://localhost:8080/domibus/services/backend"/>
        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

4.2. Diagrams source file

These file(s) contain diagrams of the present document:

Sequence diagrams.vsd