



EUROPEAN COMMISSION

DIGIT  
Connecting Europe Facility

## **Access Point**

# **Interface Control Document**

## **WS plugin**

Version [1.18]

Status [Final]

© European Union, 2018

Reuse of this document is authorised provided the source is acknowledged. The Commission's reuse policy is implemented by Commission Decision 2011/833/EU of 12 December 2011 on the reuse of Commission documents.

Date: 30/07/2018

## Document Approver(s):

| Approver Name  | Role          |
|----------------|---------------|
| Joao RODRIGUES | CEF eDelivery |

## Document Reviewers:

| Reviewer Name                    | Role              |
|----------------------------------|-------------------|
| Joao RODRIGUES                   | Programme Manager |
| Adrien FERAL                     | Product Owner     |
| Daniels MAARTEN                  | Tester            |
| Ioana DRAGUSANU                  | Developer         |
| Yves ADAM                        | Business Analyst  |
| Christian Koch                   | eCodex team       |
| Federico Martini                 | Developer         |
| VENUGOPAL Arun Raj               | Developer         |
| Cosmin BACIU                     | Developer         |
| Caroline AEBY and Chaouki BERRAH | Technical writers |

## Summary of Changes:

| Version | Date       | Created by                      | Short Description of Changes  |
|---------|------------|---------------------------------|---|
| 0.01    | 08/02/2016 | Pedro TAVARES                   | Initial version   |
| 0.02    | 18/02/2016 | Pedro TAVARES                   | Update document after revision with Adrien FERAL  |
| 0.03    | 02/03/2016 | Pedro TAVARES                   | Update document with the backend.wsdl   |
| 0.04    | 15/03/2016 | Pedro TAVARES                   | Update document with ecodex comments  |
| 0.05    | 29/03/2016 | Pedro TAVARES                   | Update document with Adrien FERAL comments  |
| 0.06    | 31/03/2016 | Pedro TAVARES                   | Update document with Daniels MAARTEN comments   |
| 0.07    | 13/04/2016 | Pedro TAVARES                   | Update document with the datamodel  |
| 0.08    | 22/04/2016 | Pedro TAVARES                   | Update document after revision with Joao RODRIGUES                                      |
| 0.09    | 29/04/2016 | Pedro TAVARES                   | Update document with Adrien FERAL comments  |
| 1.0     | 11/05/2016 | Pedro TAVARES                   | Update document with Yves ADAM and Ioana DRAGUSANU comments                             |
| 1.01    | 31/05/2016 | Pedro TAVARES                   | Update document with Federico MARTINI comments and updated schema                       |
| 1.02    | 05/09/2016 | Yves ADAM                       | Adapt changes for Domibus 3.2 upgrade, include additional XSD's and diagram source file |
| 1.03    | 15/09/2016 | Cosmin BACIU                    | Update the document for Domibus 3.2-RC1   |
| 1.04    | 30/09/2016 | Ioana DRAGUSANU                 | Update the document for Domibus 3.2   |
| 1.05    | 30/01/2017 | Arun Raj VENUGOPAL              | Add "Downloaded" state in C3  |
| 1.06    | 22/03/2017 | Ioana DRAGUSANU<br>Cosmin BACIU | Made "to:" element OPTIONAL   |

|      |            |                           |  |
|------|------------|---------------------------|--|
|      |            |                           | Upgrade from version 3.2.2 to 3.2.3  |
| 1.07 | 10/04/2017 | Cosmin BACIU              | Use a Domibus generic version instead of a specific version  |
| 1.08 | 10/07/2017 | Cosmin BACIU<br>Yves ADAM | Changes for 3.3  |
| 1.09 | 02/10/2017 | Ioana Dragusanu           | Replace the deprecated methods with new GetStatus, SubmitMessage and RetrieveMessage in SOAP examples and method description. The XSDs were already updated. |
| 1.10 | 09/10/2017 | CEF Support               | List of reviewers updated  |
| 1.11 | 30/10/2017 | Yves ADAM                 | Adjust deprecated methods/operations names in text   |
| 1.12 | 23/01/2018 | Yves ADAM                 | Alignments and corrections for Domibus version 3.3.2   |
| 1.13 | 20/03/2018 | CEF Support               | Reuse notice added, e-SENS AS4 profile replaced by eDelivery AS4 profile   |
| 1.14 | 21/06/2018 | CEF Support               | Removed reference to Domibus release number in document title.   |
| 1.15 | 05/07/2018 | CEF Support               | Updated for Domibus 4.0  |
| 1.16 | 24/07/2018 | Catalin ENACHE            | Remove deprecated operations   |
| 1.17 | 25/07/2018 | Ioana DRAGUSANU           | Update pMode example, multi-tenancy security, ws-plugin.properties file  |
| 1.18 | 27/07/2018 | Chaouki BERRAH            | References added   |

# Table of Contents

|   |           |
|---|-----------|
| <b>1. INTRODUCTION .....</b>  | <b>6</b>  |
| 1.1. Purpose of the Interface Control Document.....   | 6         |
| 1.2. Scope of the document .....  | 6         |
| 1.3. Audience.....  | 6         |
| 1.4. References.....  | 7         |
| <b>2. INTERFACE FUNCTIONAL SPECIFICATION .....</b>  | <b>8</b>  |
| 2.1. Purpose of the Access Point component .....  | 8         |
| 2.2. Use case overview .....  | 9         |
| 2.2.1. Actors.....  | 9         |
| 2.2.2. Use cases diagram .....  | 9         |
| 2.3. Detailed uses cases.....   | 10        |
| 2.3.1. UC01 – Submit Message.....   | 12        |
| 2.3.2. UC02 - Retrieve Message .....  | 23        |
| 2.3.3. UC03 - Get the status of the Message.....  | 29        |
| 2.3.4. UC04 – List Pending Messages .....   | 32        |
| <b>3. INTERFACE BEHAVIOURAL SPECIFICATION .....</b>   | <b>34</b> |
| 3.1. WS plugin configuration .....  | 34        |
| 3.2. WSDL model for Domibus 4.x.y .....   | 34        |
| 3.2.1. The WSDL schema .....  | 34        |
| 3.2.2. The data model applicable to SubmitMessage from C1 to C2 (domibus-submission.xsd) .. | 35        |
| 3.2.2.1. Messaging/UserMessage/MessageInfo: .....   | 35        |
| 3.2.2.2. Messaging/UserMessage/PartyInfo .....  | 36        |
| 3.2.2.3. Messaging/UserMessage/CollaborationInfo.....                                       | 39        |
| 3.2.2.4. MessageProperties.....   | 41        |
| 3.2.2.5. PayloadInfo.....   | 42        |
| <b>4. ANNEXES.....</b>  | <b>45</b> |
| 4.1. Interface Message standards .....  | 45        |
| 4.1.1. Errors codes table.....  | 45        |
| 4.1.2. The PMode configuration file.....  | 48        |
| 4.1.3. The PMode validation file (xsd).....   | 51        |
| 4.1.4. Web service WSDL.....  | 56        |
| 4.1.5. Web service Schema's .....   | 59        |
| 4.1.5.1. xmlmime.xsd .....  | 59        |
| 4.1.5.2. envelope.xsd.....  | 59        |
| 4.1.5.3. domibus-backend.xsd.....   | 60        |
| 4.1.5.4. domibus-header.xsd .....   | 64        |
| <b>5. ANNEXE 1 – DOCUMENT PARTS .....</b>   | <b>66</b> |

**6. LIST OF FIGURES ..... 67**

**7. CONTACT INFORMATION ..... 68**

## 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 4.x.y 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 4.x.y delivered by CEF.

There is 1 interface described in this document:

| Interface                      | Description                        | Version |
|--------------------------------|------------------------------------|---------|
| <b>BackendService_1_1.wsdl</b> | The backend webservice for Domibus | 4.x.y   |

### 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<br>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] | eDelivery AS4 profile   | <a href="https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4">https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eDelivery+AS4</a>  |
| [REF12] | eDelivery – Pmode Configuration   | eDelivery – Pmode Configuration<br><i>(will be available at a later stage)</i>   |
| [REF13] | <a href="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/">http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/</a>   | XSDs for ebms3   |
| [REF14] | <a href="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/cs02/ebms_core-3.0-spec-cs-02.pdf">http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/cs02/ebms_core-3.0-spec-cs-02.pdf</a> | 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.

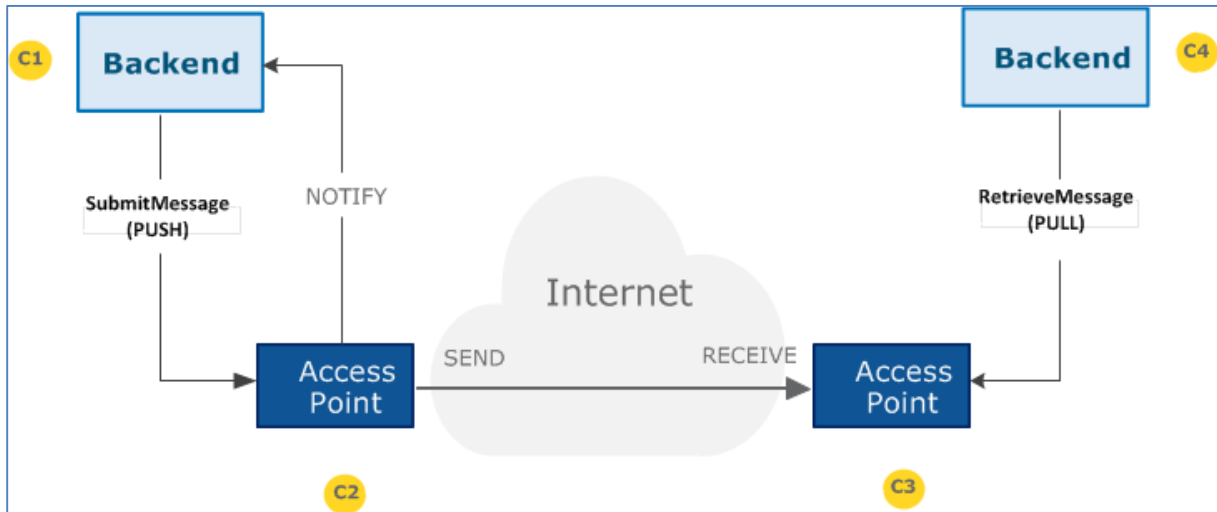


Figure 1 - The four corner model

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. Purpose of the Access Point component

The Access Point provides the functionality supporting Corner Two and Corner Three components.



## 2.2. Use case overview

### 2.2.1. Actors

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

Table 1 - Actor list

**Note:** greyed use cases in this paragraph show deprecated operations in the WSDL (in these diagrams, the use cases below these replace them). Since deprecated and replacing operations have the same functionality (only technical changes), in each case only one use case is presented for both.

### 2.2.2. Use cases diagram

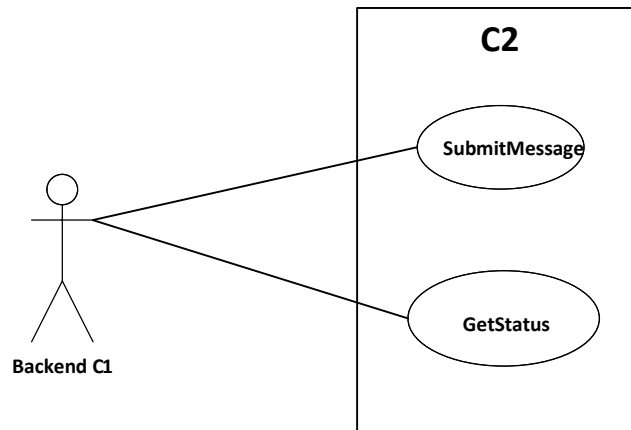


Figure 1 - Backend C1 Use cases diagram

| ID   | UC                        | Short description  | Oper.         | System        |
|------|---------------------------|--|---------------|---------------|
| UC01 | Submit Message            | Submit any type of document from an Backend C1 to a Backend C4 | submitMessage | Domibus 4.x.y |
| UC03 | Get Status of the Message | Get the status of the Message                                  | getStatus     | Domibus 4.x.y |

Table 2 - C2 Use cases

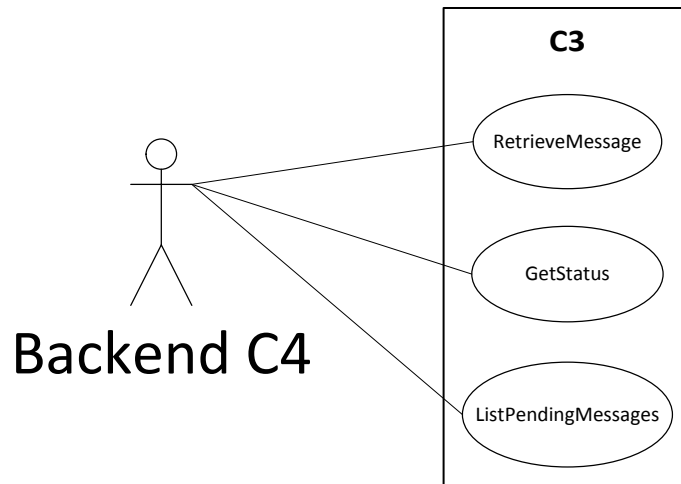


Figure 2 - backend C4 Use cases diagram

| ID   | UC                        | Short description  | Oper.               | System        |
|------|---------------------------|--|---------------------|---------------|
| UC02 | Download Message          | Retrieve the message from the Receiving AP C3                        | retrieveMessage     | Domibus 4.x.y |
| UC03 | Get Status of the Message | Get the status of the Message  | getStatus           | Domibus 4.x.y |
| UC04 | ListPending Messages      | Check the pending messages to be retrieved by the Backend C4 from C3 | listPendingMessages | Domibus 4.x.y |

Table 3 - C3 Use cases

## 2.3. Detailed uses 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 [eDelivery 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:

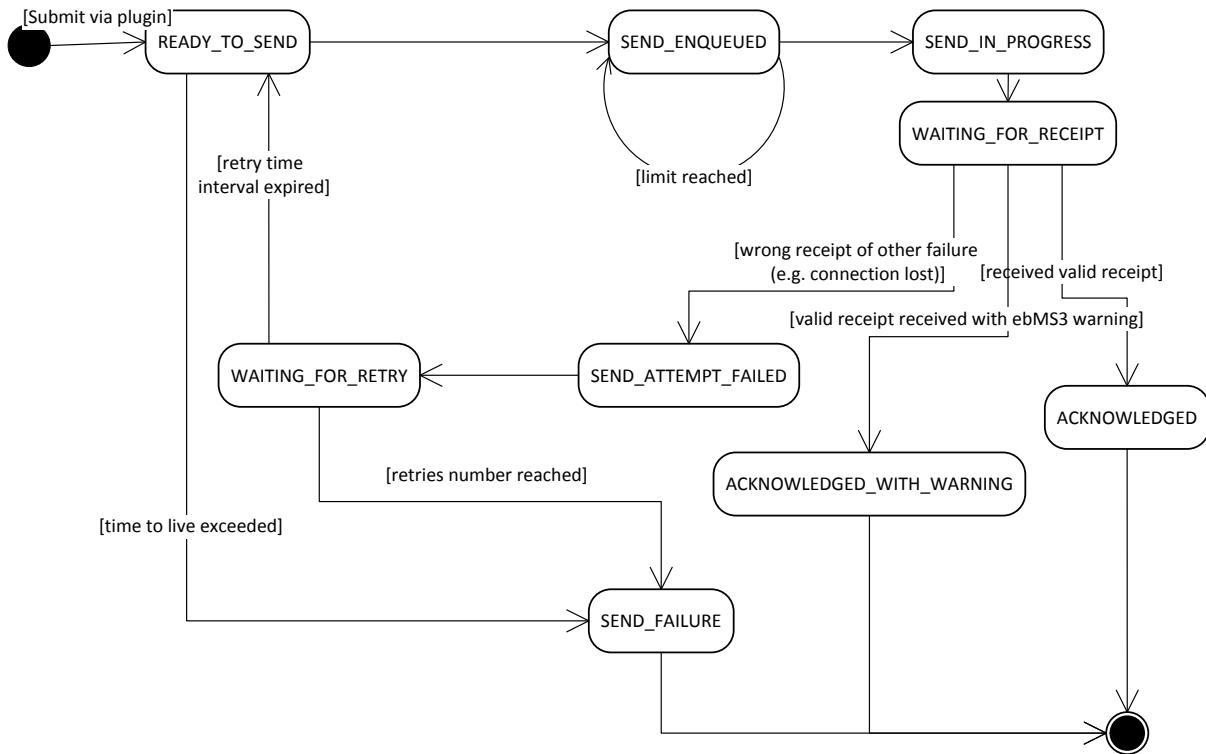


Figure 3 – State machine of C2

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

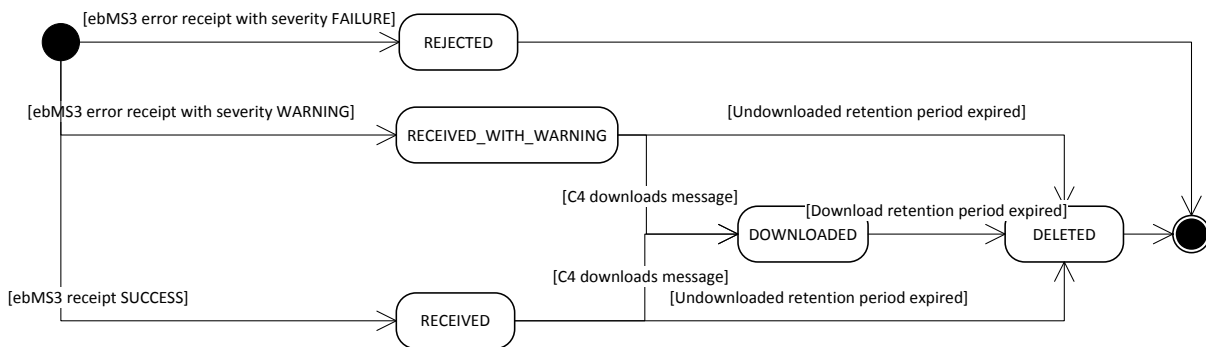


Figure 4 - State machine of C3

### 2.3.1. UC01 – Submit Message

#### Brief description

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

MTOM feature is required when sending large files.

The state machine of the outgoing messages is the following:

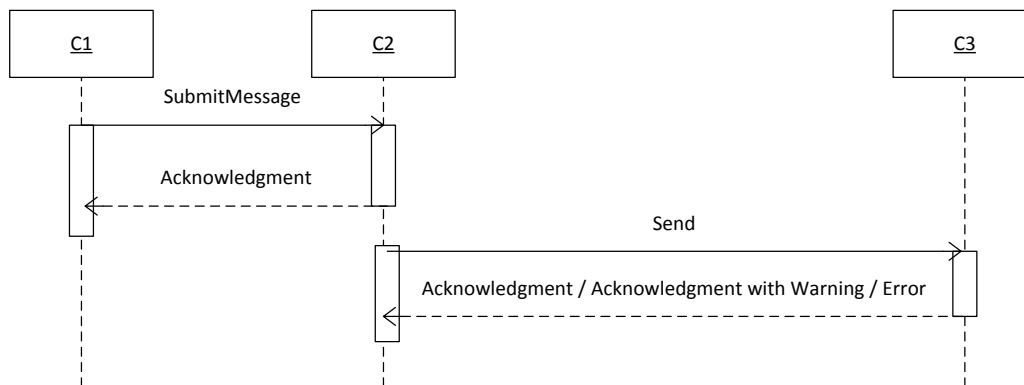


Figure 5 - Sequence Diagram C1 to C2 – SubmitMessage

## Actors

|    |                 |
|----|-----------------|
| C1 | Backend C1      |
| C2 | Access Point C2 |
| C3 | Access Point C3 |

## Preconditions

|       |   |
|-------|---|
| C1    | Backend C1 has a message to submit.   |
| 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 Message State    | C3 Message 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) submits the message to the Receiving AP (C3). The message status is updated to SEND_IN_PROGRESS whilst the message is being sent.    | SEND_IN_PROGRESS    |                  |
| C3         | 7 The Receiving AP (C3) receives the message from the Sending AP (C2)  | SEND_IN_PROGRESS    |                  |
| C2         | 9 Once the sending process finishes the status changes to <u>WAITING FOR RECEIPT</u> .   | 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. ACKNOWLEDGED RECEIVED  
If configured in the PMode for non-repudiation, the receipt SHOULD contain a single  
ebbsig:NonRepudiationInformation child element. The value of  
eb:MessageInfo/eb:RefToMessageId MUST refer to the message for which this signal is  
a receipt.
- 12 Use case ends in successful condition.

| Alternative flow |        |   | C2 Message State          | C3 Message state       |
|------------------|--------|---|---------------------------|------------------------|
| C2               | A2.1   | <b>The Backend C1 has not provided a messageID (the field is missing)</b>   |                           |                        |
| 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   | <b>Undefined condition</b>  | 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 Message State    | C3 Message state |
|----------------|------------------|--|---------------------|------------------|
| <b>C2</b>      | <b>E2.1</b>      | <b>The ID provided by C1 already exists</b>  |                     |                  |
|                | E2.1.1           | The parameter PMode[1].ReceptionAwareness.DuplicateDetection must be set to true                     |                     |                  |
|                | E2.1.2           | The status of the message changes to <u>SEND_FAILURE</u> .   | SEND_FAILURE        |                  |
| <b>C2</b>      | <b>E11.1</b>     | <b>Wrong receipt received or any other failure (e.g connection lost)</b>                             |                     |                  |
| 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   |                     |                  |
| <b>C2</b>      | <b>E11.1.3.1</b> | <b>The maximum number of retries (Configurable via PMode on a by-usecase basis) has been reached</b> |                     |                  |
| C2             | E11.1.3.1.1      | The status of the message changes to <u>SEND_FAILURE</u> .   | 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 getStatus 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 getStatus 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/">
  <soap:Header>
    <ns:Messaging>
      <ns:UserMessage>
        <ns:PartyInfo>
          <ns:From>
            <ns:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-blue</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">domibus-red</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="tc1">bdx:noprocess</ns:Service>
          <ns:Action>TC1Leg1</ns:Action>
        </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">text/xml</ns:Property>
            </ns:PartProperties>
          </ns:PartInfo>
        </ns:PayloadInfo>
      </ns:UserMessage>
    </ns:Messaging>
  </soap:Header>
  <soap:Body>
    <_1:submitRequest>
      <payload payloadId="cid:message" contentType="text/xml">
        <value>PD94bWwgdmVyc2lvbj0iMS4wliBlbmNvZGluc2VzOjVVRGLTgiPz4KPz4KPHBvZGVmcmxkPC9oZWxsZ4=</value>
      </payload>
    </_1:submitRequest>
  </soap:Body>
</soap:Envelope>

```

</soap:Envelope>

**Example of the response**

```

<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <ns5:submitResponse xmlns:ns6="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
xmlns:ns5="http://org.ecodex.backend/1_1/" xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
      <messageID>23dce7d9-2781-4623-beeb-6b43ab9e7d37@domibus.eu</messageID>
    </ns5:submitResponse>
  </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:SignalMessage>
    </eb3:Messaging>
  </S12:Header>

```

```

</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>VExWpf0Knd9Wsd2sqrnbnbuSW99V25zYqFqFJz/tZwM=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>MpbhEgDJBHtwyyl6ecTHOVlVtGgMI7I5rHWtbeBXnLGtMBrpEEziPIJ+FYy1Tk0uDIGTqsJKhqegTAI9wbj4o/osd2fKmpCsXH3GjAbDnlfwRFpNbfwG874EtXi2Jy/qwY9pv9JMtHCP/Lloj7zKD+8gbiRMMOvGh2z+7lcpOby/drVdpgWia7by/qZtQP9MTInRwxatscuF/WvNaRudNfXusHuW1FEjsNDhBYtQ/U1mRDklhF9rli1/YSFmJ4hSwnJ6jTzaAwDiyClxsmY+Z19VwzDdeEGXsKCWakp64zRapj8mT9ED7ZcmAuMtrhrXp1+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-7375525e68f">
        <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+KPctatpQhkh6S9qEltb4=</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"
refToMessageId="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/xmldsig-more#sha256"/>
            <ds:DigestValue>h3h/rjVfxae7jN/mtVa9UGr9Rn7+KUXTZDirZagcp3Y=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>mIOW1IFVzYXkdC79xIE8leKkMEGn15GFMkS9X0W+aPHWW6niNwgZ01zAsHvVQKFPjNAFmVFya4DaXn3Adv9+/bis1
HzwWf/ccVQokOgMKFkyUb5KpFVWWBa3TPXUxzUcjFtZ1a0dG3eVmWa4lg9+eOBGNvnbkpsPljkazGPKvqV+cAXg0vjewtWQklakOD1noB
UE1ETaBj9aKMKFviteELoG+pRMB5IMWt7zvQ3J7WfStupLBbqyxfhKCFI7nckV27mPR0fDBryX1grR/LseeQqLuwHz1j+ppPdVm2owPVabAI
TVYd3xBrhipdZ7ZxQ1GOM+kL000hW4snHArWZA==</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+KPctatpQhhk6S9qEltb4=</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

In Multi-tenancy mode, the general property `domibus.auth.unsecureLoginAllowed` is ignored and authentication is always required. More about authentication options can be read in the [Administration Guide](#), section 6.1.2.1 Domibus Authentication.

### 2.3.2. UC02 - Retrieve Message

## Brief description

Retrieve 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.

Please note that RetrieveMessage method replaces the deprecated method retrieveMessage to support the retrieval of large files. MTOM feature is required when retrieving large files.

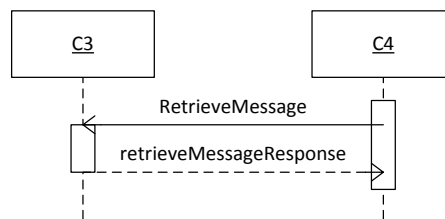


Figure 6 - Sequence Diagram C4 to C3 – RetrieveMessage

## 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 Message State |
|-------|---|--|--|------------------|
| C4    | 1 | Requests, to the Receiving AP C3, the service retrieveMessage by providing the messageID   |  | RECEIVED         |
| C3    | 2 | Receiving AP C3 retrieves and sends the information retrieveMessageResponse 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. The status of the message changes to <u>DOWNLOADED</u> when the message is retrieved by the backend C4. |  | DOWNLOADED       |
| C4    | 3 | Receives the message as sent by C1   |  | DOWNLOADED       |



|    |   |  |         |
|----|---|--|---------|
| C3 | 4 | Deletes the payload of the message from the database if retention timeout for downloaded messages = 0.<br>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 <i>DELETED</i> when the message is deleted by C3 after the configured retention timeout for downloaded messages ( <i>retention_downloaded</i> ) expired. Note: If <i>retention_downloaded</i> has a negative value, the message will never be deleted from C3. If the message was not downloaded, the <i>retention_undownloaded</i> value will be used as timeout for deletion. | DELETED |
|    | 6 | Use case ends  | DELETED |

| <b>Alternative flow</b> |        |   | <b>C3 Message State</b> |
|-------------------------|--------|---|-------------------------|
| C3                      | A1.1   | <b>Configured retention time has passed</b> | RECEIVED                |
| C3                      | A1.1.1 | Go directly to step 4                       | RECEIVED                |

| Exception flow |        |   | C3 Message State |
|----------------|--------|---|------------------|
| C3             | E2.1   | <b>Wrong messageID (malformed or missing), this is a condition of failure.</b>  | (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   | <b>Message status is REJECTED</b>   |                  |
| 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 ACKNOWLEDGED or ACKNOWLEDGED WITH WARNINGS on C2 and DOWNLOADED or DELETED on C3.

Payload of the message may be 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 GetStatus 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 retrieveMessage method 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**

```

2.3.2.1.1.1.1. <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:_1="http://org.ecodex.backend/1_1/">
2.3.2.1.1.1.2. <soap:Header/>
2.3.2.1.1.1.3. <soap:Body>
  <_1:retrieveMessageRequest>
2.3.2.1.1.1.4. <messageID>${ResponseParameters#messageID}</messageID>
2.3.2.1.1.1.5. </_1:retrieveMessageRequest>
2.3.2.1.1.1.6. </soap:Body>
</soap:Envelope>

```

**Example of the response**

```

2.3.2.1.1.1.7. <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
2.3.2.1.1.1.8. <soap:Header>
2.3.2.1.1.1.9. <ns6:Messaging mustUnderstand="false" xmlns:ns6="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/" xmlns:ns5="http://org.ecodex.backend/1_1/" xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
2.3.2.1.1.1.10. <ns6:UserMessage>
2.3.2.1.1.1.11. <ns6:MessageInfo>
2.3.2.1.1.1.12. <ns6:Timestamp>2017-10-02T17:32:14.956+02:00</ns6:Timestamp>
2.3.2.1.1.1.13. <ns6:MessageId>d05051c6-951c-4f40-90b5-459eca9d8302@domibus.eu</ns6:MessageId>
2.3.2.1.1.1.14. </ns6:MessageInfo>
2.3.2.1.1.1.15. <ns6:PartyInfo>
2.3.2.1.1.1.16. <ns6:From>
2.3.2.1.1.1.17. <ns6:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-blue</ns6:PartyId>
2.3.2.1.1.1.18. <ns6:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</ns6:Role>
2.3.2.1.1.1.19. </ns6:From>
2.3.2.1.1.1.20. <ns6:To>
2.3.2.1.1.1.21. <ns6:PartyId type="urn:oasis:names:tc:ebcore:partyid-type:unregistered">domibus-red</ns6:PartyId>
2.3.2.1.1.1.22. <ns6:Role>http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</ns6:Role>
2.3.2.1.1.1.23. </ns6:To>
2.3.2.1.1.1.24. </ns6:PartyInfo>
2.3.2.1.1.1.25. <ns6:CollaborationInfo>
2.3.2.1.1.1.26. <ns6:Service type="tc1">bdx:noprocess</ns6:Service>
2.3.2.1.1.1.27. <ns6:Action>TC1Leg1</ns6:Action>
2.3.2.1.1.1.28. <ns6:ConversationId>52f1c57d-bd35-4ab2-a0a5-da9a15101dba@domibus.eu</ns6:ConversationId>
2.3.2.1.1.1.29. </ns6:CollaborationInfo>
2.3.2.1.1.1.30. <ns6:MessageProperties>
2.3.2.1.1.1.31. <ns6:Property name="finalRecipient">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C4</ns6:Property>
2.3.2.1.1.1.32. <ns6:Property name="originalSender">urn:oasis:names:tc:ebcore:partyid-type:unregistered:C1</ns6:Property>
2.3.2.1.1.1.33. </ns6:MessageProperties>
2.3.2.1.1.1.34. <ns6:PayloadInfo>
2.3.2.1.1.1.35. <ns6:PartInfo href="cid:message">
2.3.2.1.1.1.36. <ns6:Schema/>
2.3.2.1.1.1.37. <ns6:PartProperties>
2.3.2.1.1.1.38. <ns6:Property name="MimeType">text/xml</ns6:Property>
2.3.2.1.1.1.39. </ns6:PartProperties>
2.3.2.1.1.1.40. </ns6:PartInfo>
2.3.2.1.1.1.41. </ns6:PayloadInfo>
2.3.2.1.1.1.42. </ns6:UserMessage>
2.3.2.1.1.1.43. </ns6:Messaging>
2.3.2.1.1.1.44. </soap:Header>
2.3.2.1.1.1.45. <soap:Body>
2.3.2.1.1.1.46. <ns5:retrieveMessageResponse xmlns:ns6="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
xmlns:ns5="http://org.ecodex.backend/1_1/" xmlns:xmime="http://www.w3.org/2005/05/xmlmime">
2.3.2.1.1.1.47. <payload payloadId="cid:message">
2.3.2.1.1.1.48. <value>PD94bWwgdmVyc2lvbjoIMSM4wliBlbmNvZGluzZ0iVVRGLTgiPz4KPghbGxvPndvcmxkPC9oZWxs4=</value>
2.3.2.1.1.1.49. </payload>
2.3.2.1.1.1.50. </ns5:retrieveMessageResponse>
2.3.2.1.1.1.51. </soap:Body>
</soap:Envelope>

```

## Security

In Multi-tenancy mode, the general property `domibus.auth.unsecureLoginAllowed` is ignored and authentication is always required. More about authentication options can be read in the [Administration Guide](#), section 6.1.2.1 Domibus Authentication.

### 2.3.3. 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 7 - Sequence Diagram – GetStatus

#### Actors

|    |                 |
|----|-----------------|
| C1 | Backend C1      |
| C2 | Access Point C2 |
| C3 | Access Point C3 |
| C4 | Backend C4      |

#### Preconditions

There is at least one message sent by Backend C1 or to be retrieved 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 statusRequest using the messageId.
2. The Access Point (Sending AP C2 or Receiving AP C3) retrieve the getStatusResponse.
3. Use case ends.

#### Exception flow

N/A

## Post conditions

### Successful conditions

The operation is a success if GetStatusResponse 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
- DOWNLOADED

### Failure Conditions

The message doesn't exist.

## Special requirements

N/A

## Security

In Multi-tenancy mode, the general property **domibus.auth.unsecureLoginAllowed** is ignored and authentication is always required. More about authentication options can be read in the [Administration Guide](#), section 6.1.2.1 Domibus Authentication.

### Example of the request

```
2.3.3.1.1.1.1. <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:_1="http://org.ecodex.backend/1_1/">
2.3.3.1.1.1.2. <soap:Header/>
2.3.3.1.1.1.3. <soap:Body>
2.3.3.1.1.1.4. <_1:statusRequest>
2.3.3.1.1.1.5. <messageID>$d05051c6-951c-4f40-90b5-459eca9d8302@domibus.eu</messageID>
2.3.3.1.1.1.6. </_1:statusRequest>
2.3.3.1.1.1.7. </soap:Body>
</soap:Envelope>
```

### Example of the response

```
2.3.3.1.1.1.8. <soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
2.3.3.1.1.1.9. <soap:Body>
2.3.3.1.1.1.10. <ns5:getStatusResponse xmlns:ns6="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/"
xmlns:ns5="http://org.ecodex.backend/1_1/"
xmlns:xmime="http://www.w3.org/2005/05/xmlmime">NOT_FOUND</ns5:getStatusResponse>
```

```
2.3.3.1.1.1.11. </soap:Body>  
</soap:Envelope>
```

### 2.3.4. UC04 – List Pending Messages

#### Brief description

List the status Messages pending to be received by the Backend C4.

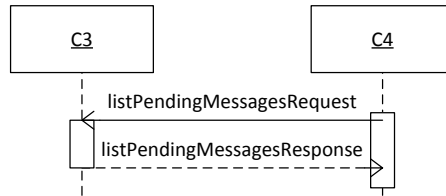


Figure 8 - Sequence Diagram C4 to C3 – ListPendingMessages

#### 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 retrieved or the list is empty in the case that there are no pending messages.

##### Failure Conditions



N/A

## Special requirements

N/A

## Security

In Multi-tenancy mode, the general property `domibus.auth.unsecureLoginAllowed` is ignored and authentication is always required. More about authentication options can be read in the [Administration Guide](#), section 6.1.2.1 Domibus Authentication.

### 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>
```

### 3. INTERFACE BEHAVIOURAL SPECIFICATION

#### 3.1. WS plugin configuration

The WS plugin configuration is done in the ws-plugin.properties file. Following properties are configurable in this file:

| Property name                      | Default value | Description                   |
|------------------------------------|---------------|-------------------------------|
| wsplugin.mtom.enabled              | false         | Enable the support for MTOM.  |
| wsplugin.schema.validation.enabled | true          | Enable the schema validation. |

#### 3.2. WSDL model for Domibus 4.x.y

##### 3.2.1. The WSDL schema

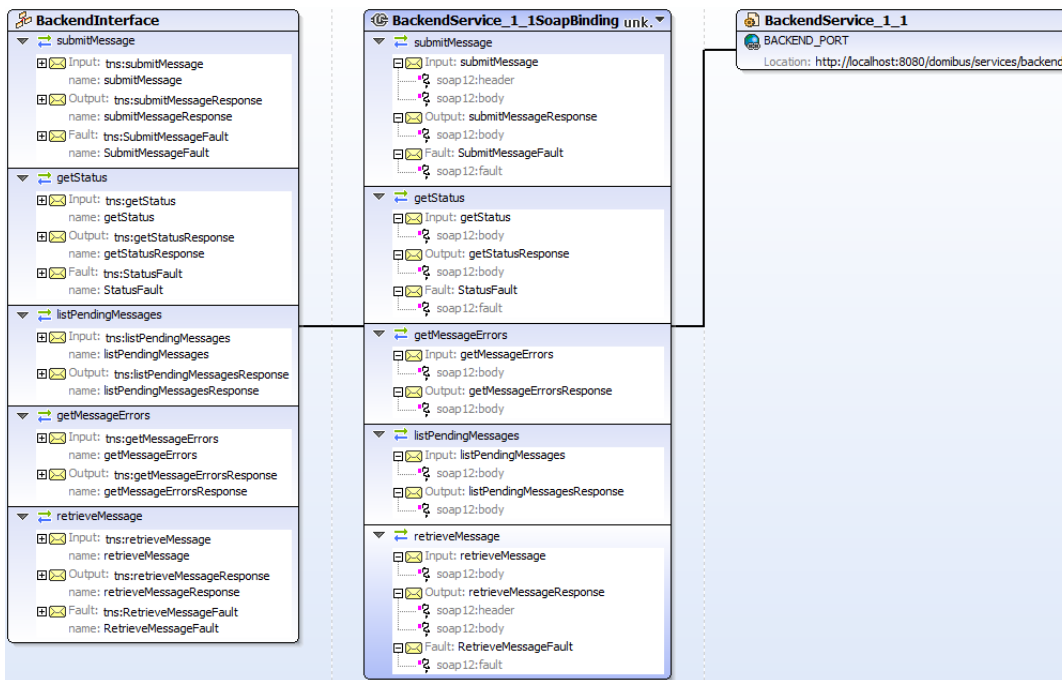


Figure 9 - WSDL model for Domibus 4.x.y

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**:

- submitMessage
- getStatus
- listPendingMessages
- getMessageErrors → It can be used if you get a SEND\_FAILURE or REJECTED status as response from the getStatus service 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.

- retrieveMessage

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

- <wsdl:fault name="SubmitMessageFault"/>
- <wsdl:fault name="RetrieveMessageFault"/>

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"/>

**3.2.2. The data model applicable to SubmitMessage from C1 to C2 (domibus-submission.xsd)**

In this section the data model is explained.

**3.2.2.1. Messaging/UserMessage/MessageInfo:**

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

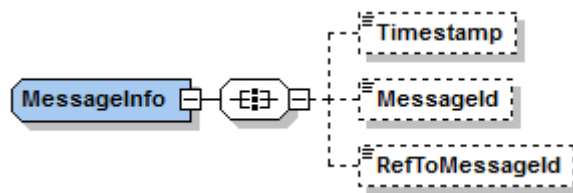


Figure 10 – MessageInfo type

- **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']   | N         | Max 1       | 1. A globally unique identifier<br>2. In the Message-Id and Content-Id MIME | 346ea37f-7583-40b0-9ffc- |

| Description    | Field (xpath)  | Mandatory | Occurrences | Constraints  | Valid example                                   |
|----------------|--|-----------|-------------|--|---|
|                | ma']/*[local-name()='complexType' and @name='MessageInfo']<br>/*[local-name()='all']/*[local-name()='element' and @name='MessageId']                           |           |             | 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.<br>3. Max length: value should not be more than 255 characters.  | 3f4cfa88bf8b@domibus.eu                         |
| RefToMessageId | /*[local-name()='schema']/*[local-name()='complexType' and @name='MessageInfo']<br>/*[local-name()='all']/*[local-name()='element' and @name='RefToMessageId'] | N         | Max 1       | 1. A globally unique identifier.<br>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.<br>3. Max length: value should not be more than 255 characters. | 346ea37f-7583-40b0-9ffc-3f4cfa88bf8b@domibus.eu |

### 3.2.2.2. 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.

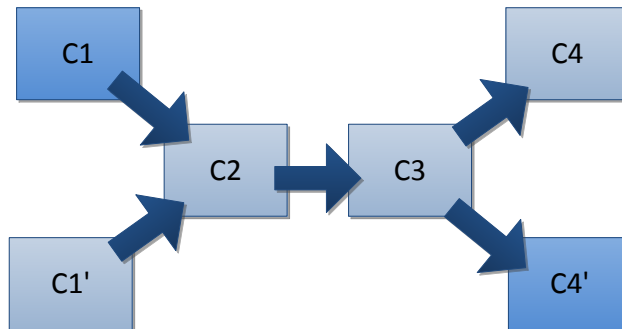


Figure 11 - The From - To PartyInfo

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.

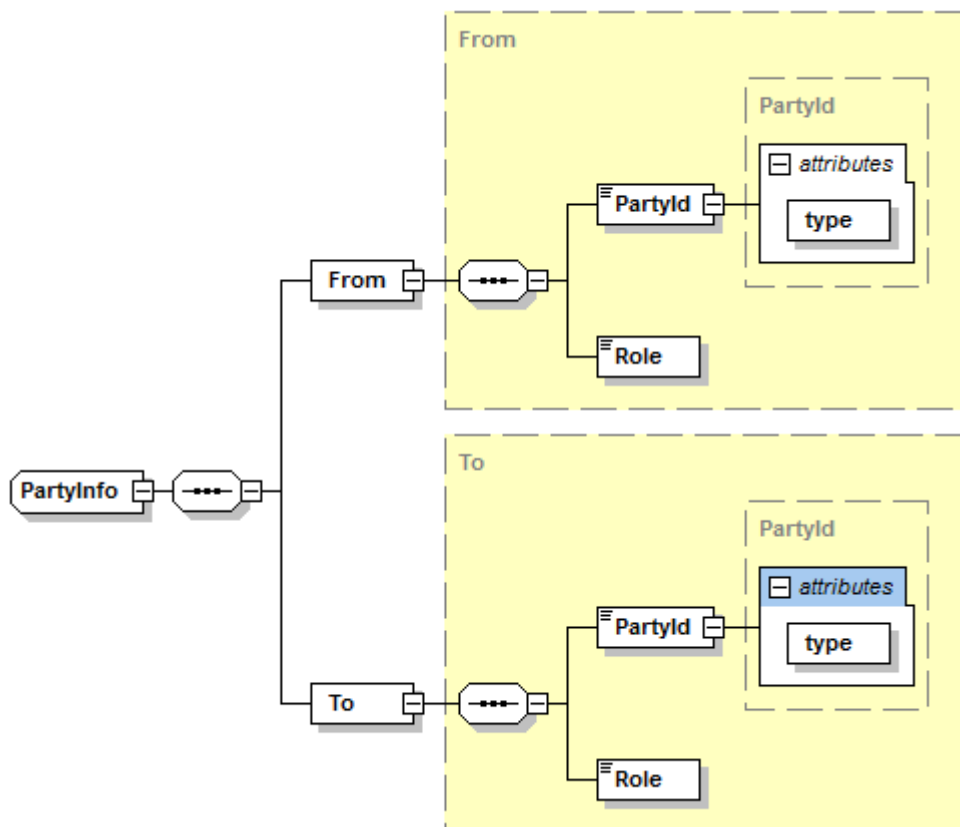


Figure 12 – PartyInfo type

| Description  | Field (xpath)   | Mandatory | Occurrences | Constraints  | Valid example |
|--------------|---|-----------|-------------|--|---------------|
| From/PartyId | /*[local-name()='schema']/*[local-name()='complexType' and @name='From']<br>/*[local-name()='all']<br>/*[local-name()='element' and @name='Pa | Y         | Max 1       | <ol style="list-style-type: none"> <li>1. The content of the PartyId element MUST be a URI if Type is not used.</li> <li>2. The PartyID should be the same that is used in the PMode configuration:</li> <li>3. Max length:255 characters</li> <li>4. Configuration in PMode: PMode.Initiator.Party</li> </ol> | C2            |

| Description       | Field (xpath)  | Mandatory | Occurrences | Constraints   | Valid example   |
|-------------------|--|-----------|-------------|---|---|
|                   | rtyId']  |           |             |   |   |
| From/Role         | /*[local-name()='schema']/*[local-name()='complexType' and @name='From']<br>/*[local-name()='all']<br>/*[local-name()='element' and @name='Role']  | Y         | Max 1       | 1. It is a non-empty string,<br>2. Max length:255 characters<br>3. Configuration in PMode: PMode.Initiator.Role   | <a href="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator">http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/initiator</a><br>or |
| To/PartyId        | /*[local-name()='schema']/*[local-name()='complexType' and @name='To']<br>/*[local-name()='all']<br>/*[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.<br>2. The PartyID should be the same that is used in the PMode configuration.<br>3. Max length:255 characters<br>4. Configuration in PMode: PMode.Responder.Party | C3  |
| To/Role           | /*[local-name()='schema']/*[local-name()='complexType' and @name='To']<br>/*[local-name()='all']<br>/*[local-name()='element' and @name='Role']    | Y         | Max 1       | 1. It is a non-empty string,<br>2. Max length:255 characters<br>3. Configuration in PMode: PMode.Responder.Role   | <a href="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder">http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/responder</a>       |
| From-To/PartyType | /*[local-name()='schema']/*[local-name()='complexType' and @name='PartyId']<br>/*[local-name()='attribute' and @name='type']                       | Y         | Max 1       | 1. It is a non-empty string,<br>2. Max length:255 characters  | <a href="urn:oasis:names:tc:ebcore:partyid-type:unregistered">urn:oasis:names:tc:ebcore:partyid-type:unregistered</a>   |

### 3.2.2.3. 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 eDelivery 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.

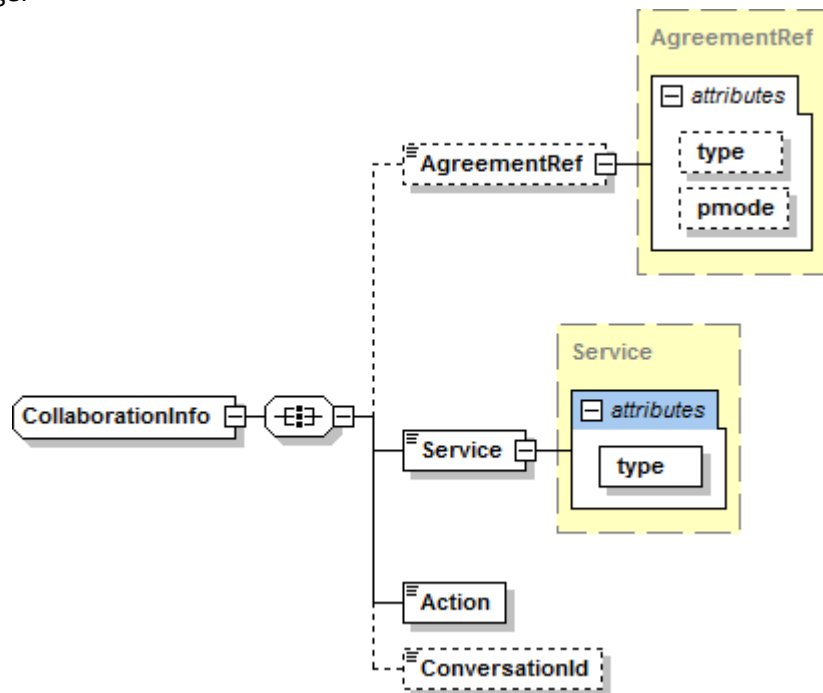


Figure 13 – CollaborationInfo type

| Description         | Field (xpath)  | Mandatory | Occurrences  | Constraints  | Valid example   |
|---------------------|--|-----------|--------------|--|---|
| <b>AgreementRef</b> | /*[local-name()='schema']/*[local-name()='complexType' and @name='CollaborationInfo']<br>/*[local-name()='all']<br>/*[local- | <b>N</b>  | <b>Max 1</b> | <ol style="list-style-type: none"> <li>1. It is a non-empty string.</li> <li>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.</li> <li>3. AgreementRef is a string value that</li> </ol> | <a href="https://joinup.ec.europa.eu/">https://joinup.ec.europa.eu/</a> |

| Description               | Field (xpath)   | Mandatory | Occurrences  | Constraints   | Valid example                  |
|---------------------------|---|-----------|--------------|---|--------------------------------|
|                           | name()='element' and @name='AgreementRef']  |           |              | <p>identifies the agreement that governs the exchange. The P-Mode under which the MSH operates for this message should be aligned with this agreement.</p> <p>4. Max length:255 characters</p>  |                                |
| <b>agreementRef@type</b>  | /*[local-name()='schema']/*[local-name()='complexType' and @name='AgreementRef']/*[local-name()='simpleContent']/*[local-name()='extension']/*[local-name()='attribute' and @name='type'] | <b>N</b>  | <b>Max 1</b> | <ol style="list-style-type: none"> <li>1. It is a non-empty string.</li> <li>2. Max length:255 characters</li> <li>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.</li> <li>4. If the type attribute is not present, the content of the AgreementRef element MUST be a URI.</li> </ol>  | <i>MyServiceTypes</i>          |
| <b>agreementRef@pmode</b> | /*[local-name()='schema']/*[local-name()='complexType' and @name='AgreementRef']/*[local-name()='simpleContent']/*[local-name()='extension']/*[local-name()='attribute' and @name='type'] | <b>N</b>  | <b>Max 1</b> | <ol style="list-style-type: none"> <li>1. It is a non-empty string.</li> <li>2. Max length:255 characters</li> <li>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.).</li> </ol> | <i>PurchaseOrderFrom ACME</i>  |
| <b>Service</b>            | /*[local-name()='schema']/*[local-name()='complexType' and @name='CollaborationInfo']/*[local-name()='all']/*[local-name()='element' and @name='Service']                                 | <b>Y</b>  | <b>Max 1</b> | <ol style="list-style-type: none"> <li>4. It is a non-empty string,</li> <li>5. Max length:255 characters</li> <li>6. Configuration in PMode: PMode[1].BusinessInfo.Service</li> </ol>  | <i>SupplierOrderProcessing</i> |
| <b>Service@Type</b>       | /*[local-name()='schema']/*[local-name()='complexType']   | <b>Y</b>  |              | <ol style="list-style-type: none"> <li>1. Indicates how the parties sending and receiving the message will interpret the value of the element.</li> <li>2. It is a non-empty string,</li> </ol>   |                                |



| Description           | Field (xpath)   | Mandatory | Occurrences  | Constraints  | Valid example                               |
|-----------------------|---|-----------|--------------|--|---|
|                       | complexType' and @name='Service'] /*[local-name()='simpleContent'] /*[local-name()='extension'] /*[local-name()='attribute']  |           |              | 3. Max length:255 characters   |   |
| <b>Action</b>         | /*[local-name()='schema'] /*[local-name()='complexType' and @name='CollaborationInfo'] /*[local-name()='all'] /*[local-name()='element' and @name='Action']         | <b>Y</b>  | <b>Max 1</b> | <ol style="list-style-type: none"> <li>1. It is a non-empty string.</li> <li>2. Action SHALL be unique within the Service in which it is defined.</li> <li>3. Max length:255 characters</li> <li>4. Configuration in PMode: PMode[1].BusinessInfo.Action</li> </ol>                | <i>NewOrder</i>                             |
| <b>ConversationId</b> | /*[local-name()='schema'] /*[local-name()='complexType' and @name='CollaborationInfo'] /*[local-name()='all'] /*[local-name()='element' and @name='ConversationId'] | <b>N</b>  | <b>Max 1</b> | <ol style="list-style-type: none"> <li>1. It is a non-empty string. Represents an immutable universally unique identifier (UUID). A UUID represents a 128-bit value.</li> <li>2. Created randomly in the Receiving Access Point C2</li> <li>3. Max length:36 characters</li> </ol> | <i>06689621-428e-48a4-86e6-4a86539363f5</i> |

### 3.2.2.4. MessageProperties

This element is REQUIRED in the 4-corner model. It 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>
```

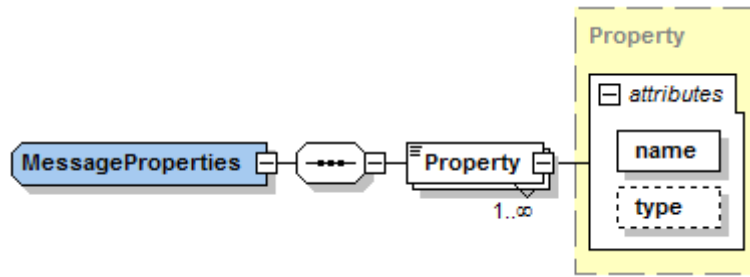


Figure 14 – MessageProperties type

| Description   | Field (xpath)   | Mandatory | Occurrences | Constraints  | Valid example         |
|---------------|---|-----------|-------------|--|-----------------------|
| Property name | <code>//*[local-name()='schema']//*[local-name()='complexType' and @name='Property']//*[local-name()='attribute' and @name='name']</code> | Y         | Max 1       | <ol style="list-style-type: none"> <li>It is a non-empty string.</li> <li>Max length:255 characters</li> <li>Configuration in PMode: PMode[1].BusinessInfo.Properties</li> </ol> | <i>originalSender</i> |
| Property type | <code>//*[local-name()='schema']//*[local-name()='complexType' and @name='Property']//*[local-name()='attribute' and @name='type']</code> | N         | Max 1       | <ol style="list-style-type: none"> <li>It is a non-empty string</li> </ol>   | <i>String</i>         |

### 3.2.2.5. 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.

- **PartProperties:** has a REQUIRED @name attribute, value which must be agreed per implementation.

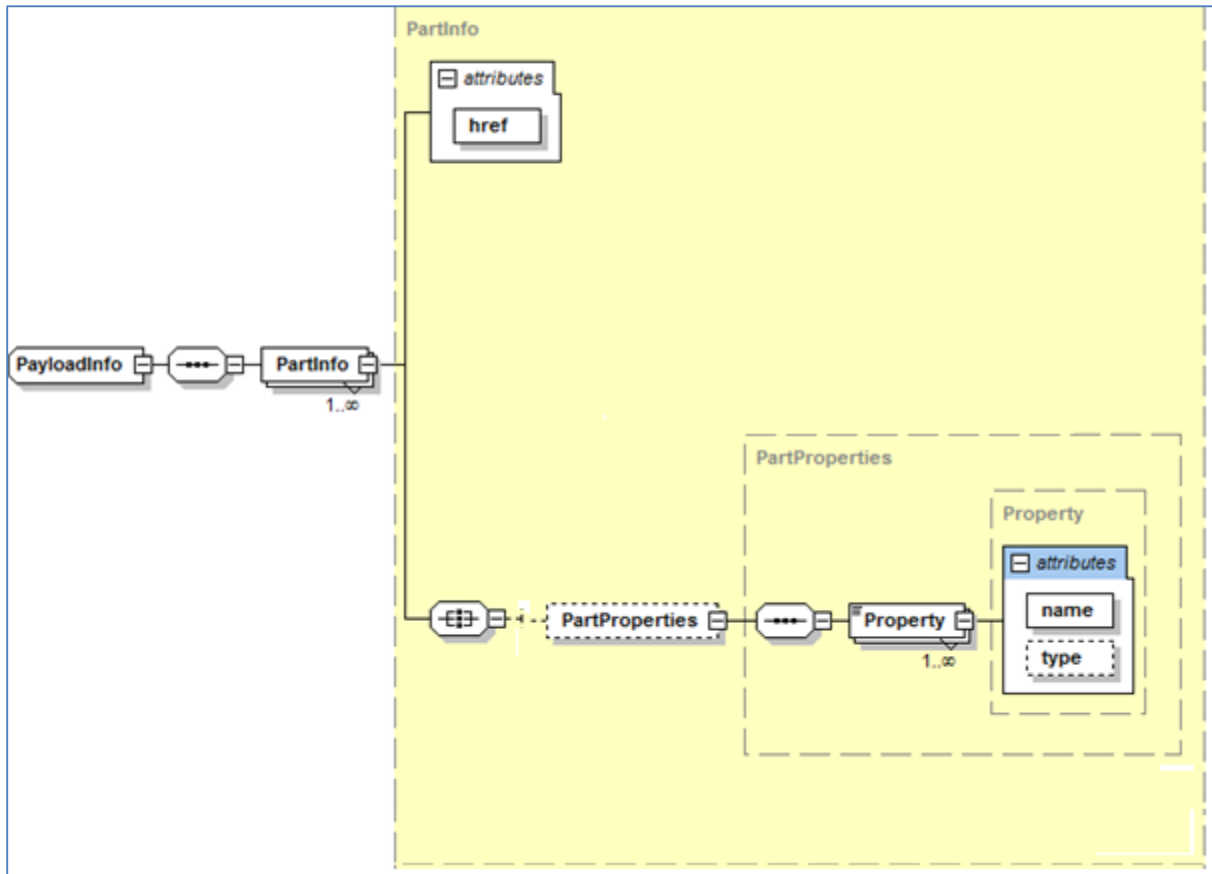


Figure 15 – PayloadInfo type

| Description                              | Field (xpath)  | Mandatory | Occurrences | Constraints  | Valid example          |
|--|--|-----------|-------------|--|------------------------|
| href                                     | <code>/*[local-name()='schema']/*[local-name()='complexType' and @name='PartInfo']/*[local-name()='attribute' and @name='href']</code> | N         | Max 1       | 1. Max length:255 characters   | <i>cid:message</i>     |
| PartProperties/<br>Property/<br>MimeType | <code>/*[local-name()='schema']/*[local-name()='complexType' and @name='PartProperties']/*[local-name()='seq</code>                    | N         | Max 1       | 1. Max length:255 characters<br>2. If the PMode compression is enabled this field is mandatory | <i>application/xml</i> |

| Description                         | Field (xpath)  | Mandatory | Occurrences  | Constraints                  | Valid example          |
|-------------------------------------|--|-----------|--------------|------------------------------|------------------------|
|                                     | uence']/*[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'] | <b>N</b>  | <b>Max 1</b> | 1. Max length:255 characters | <i>Message Payload</i> |

## 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 value greater than the known FragmentCount.   |
| EBMS_0049  | BadFragmentCount          | failure              | Content        | A value is set for FragmentCount, but a previously received fragment had a greater value.  |
| EBMS_0050  | FragmentSizeExceeded      | warning              | Unpackaging    | The size of the data part in a fragment message is greater than <b>Pmode[].Splitting.FragmentSize</b>  |
| EBMS_0051  | ReceiveIntervalExceeded   | failure              | Unpackaging    | More time than <b>Pmode[].Splitting.JoinInterval</b> 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 <b>Pmode[].Splitting.RoutingProperties</b>  |
| 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 against 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="14400"/>
    </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">
        <identifier partyId="domibus-red" partyIdType="partyTypeUrn"/>
      </party>
      <party name="blue_gw"
        endpoint="http://<blue_hostname>:8080/domibus/services/msh">
        <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="eDeliveryPropertySet">
        <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"> <!-- maxSize is currently ignored -->
        <attachment name="businessContentPayload"/>
        <attachment name="businessContentAttachment"/>
      </payloadProfile>
    </payloadProfiles>
  </db:configuration>

```



```

<securities>
  <security name="eDeliveryAS4Policy"
    policy="eDeliveryAS4Policy.xml"
    signatureMethod="RSA_SHA256" />
</securities>
<errorHandlings>
  <errorHandling name="demoErrorHandling"
    errorAsResponse="true"
    businessErrorNotifyProducer="true"
    businessErrorNotifyConsumer="true"
    deliveryFailureNotifyProducer="true"/>
</errorHandlings>
<agreements>
  <agreement name="agreement1" value="A1" type="T1"/>
</agreements>
<services>
  <service name="testService1" value="bdx:noprocess" type="tc1"/>
  <service name="testService" value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/service"/>
</services>
<actions>
  <action name="tc1Action" value="TC1Leg1"/>
  <action name="testAction" value="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/test"/>
</actions>
<as4>
  <receptionAwareness name="receptionAwareness" retry="12;4;CONSTANT" duplicateDetection="true"/>
  <reliability name="AS4Reliability" nonRepudiation="true" replyPattern="response"/>
</as4>
<legConfigurations>
  <legConfiguration name="pushTestcase1tc1Action"
    service="testService1"
    action="tc1Action"
    defaultMpc="defaultMpc"
    reliability="AS4Reliability"
    security="eDeliveryAS4Policy"
    receptionAwareness="receptionAwareness"
    propertySet="eDeliveryPropertySet"
    payloadProfile="MessageProfile"
    errorHandling="demoErrorHandling"
    compressPayloads="true"/>
  <legConfiguration name="testServiceCase"
    service="testService"
    action="testAction"
    defaultMpc="defaultMpc"
    reliability="AS4Reliability"
    security="eDeliveryAS4Policy"
    receptionAwareness="receptionAwareness"
    propertySet="eDeliveryPropertySet"
    payloadProfile="MessageProfile"
    errorHandling="demoErrorHandling"
    compressPayloads="true"/>
</legConfigurations>
<process name="tc1Process"
  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"/>
    <leg name="testServiceCase"/>
  </legs>
</process>

```

```
</process>  
</businessProcesses>  
</db:configuration>
```

### 4.1.3. The PMode validation file (xsd)

This file validates the domibus configuration files:

```

<?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:element>
        <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:all>
    </xs:complexType>
  </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:element>
        <xs:unique name="uniqueRoleName">
          <xs:selector xpath="role"/>
          <xs:field xpath="@name"/>
        </xs:unique>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

```

    </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:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```

        <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:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

```

        <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. Web service WSDL

```

<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:tns="http://org.ecodex.backend/1_1/" 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>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation="xml.xsd"/>
    </schema>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://www.w3.org/2005/05/xmlmime" schemaLocation="xmlmime.xsd"/>
    </schema>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://www.w3.org/2003/05/soap-envelope" schemaLocation="envelope.xsd"/>
    </schema>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://org.ecodex.backend/1_1/" schemaLocation="domibus-backend.xsd"/>
    </schema>
    <schema xmlns="http://www.w3.org/2001/XMLSchema">
      <import namespace="http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/ns/core/200704/" schemaLocation="domibus-
header.xsd"/>
    </schema>
  </wsdl:types>
  <wsdl:message name="getMessageErrors">
    <wsdl:part name="getErrorsRequest" element="tns:getErrorsRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="retrieveMessage">
    <wsdl:part name="retrieveMessageRequest" element="tns:retrieveMessageRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="RetrieveMessageFault">
    <wsdl:part name="RetrieveMessageFault" element="tns:FaultDetail">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="retrieveMessageResponse">
    <wsdl:part name="retrieveMessageResponse" element="tns:retrieveMessageResponse">
    </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="getStatusResponse">
    <wsdl:part name="getStatusResponse" element="tns:getStatusResponse">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="listPendingMessages">
    <wsdl:part name="listPendingMessagesRequest" element="tns:listPendingMessagesRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="getStatus">
    <wsdl:part name="statusRequest" element="tns:statusRequest">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="StatusFault">
    <wsdl:part name="StatusFault" element="tns:FaultDetail">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="submitMessage">
    <wsdl:part name="submitRequest" element="tns:submitRequest">
    </wsdl:part>
    <wsdl:part name="ebMSHeaderInfo" element="ns1:Messaging">
    </wsdl:part>
  </wsdl:message>
  <wsdl:message name="submitMessageResponse">

```



```

    <wsdl:part name="submitResponse" element="tns:submitResponse">
  </wsdl:part>
</wsdl:message>
<wsdl:message name="SubmitMessageFault">
  <wsdl:part name="SubmitMessageFault" 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="submitMessage">
    <wsdl:input name="submitMessage" message="tns:submitMessage">
</wsdl:input>
    <wsdl:output name="submitMessageResponse" message="tns:submitMessageResponse">
</wsdl:output>
    <wsdl:fault name="SubmitMessageFault" message="tns:SubmitMessageFault">
</wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="getStatus">
    <wsdl:input name="getStatus" message="tns:getStatus">
</wsdl:input>
    <wsdl:output name="getStatusResponse" message="tns:getStatusResponse">
</wsdl:output>
    <wsdl:fault name="StatusFault" message="tns:StatusFault">
</wsdl:fault>
  </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="retrieveMessage">
    <wsdl:input name="retrieveMessage" message="tns:retrieveMessage">
</wsdl:input>
    <wsdl:output name="retrieveMessageResponse" message="tns:retrieveMessageResponse">
</wsdl:output>
    <wsdl:fault name="RetrieveMessageFault" message="tns:RetrieveMessageFault">
</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="submitMessage">
    <soap12:operation soapAction="" style="document"/>
    <wsdl:input name="submitMessage">
      <soap12:header message="tns:submitMessage" part="ebMSHeaderInfo" use="literal"/>
      <soap12:body parts="submitRequest" use="literal"/>
    </wsdl:input>
    <wsdl:output name="submitMessageResponse">
      <soap12:body use="literal"/>
    </wsdl:output>
    <wsdl:fault name="SubmitMessageFault">
      <soap12:fault name="SubmitMessageFault" use="literal"/>
    </wsdl:fault>
  </wsdl:operation>
  <wsdl:operation name="getStatus">
    <soap12:operation soapAction="" style="document"/>
    <wsdl:input name="getStatus">
      <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output name="getStatusResponse">
      <soap12:body use="literal"/>
    </wsdl:output>
  </wsdl:operation>

```

```
<wsdl:fault name="StatusFault">
  <soap12:fault name="StatusFault" use="literal"/>
</wsdl:fault>
</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="retrieveMessage">
  <soap12:operation soapAction="" style="document"/>
  <wsdl:input name="retrieveMessage">
    <soap12:body use="literal"/>
  </wsdl:input>
  <wsdl:output name="retrieveMessageResponse">
    <soap12:header message="tns:retrieveMessageResponse" part="ebMSHeaderInfo" use="literal"/>
    <soap12:body parts="retrieveMessageResponse" use="literal"/>
  </wsdl:output>
  <wsdl:fault name="RetrieveMessageFault">
    <soap12:fault name="RetrieveMessageFault" 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>
```

## 4.1.5. Web service Schema's

### 4.1.5.1. *xmlmime.xsd*

```

<?xml version="1.0"?>
<!--
W3C XML Schema defined in the Describing Media Content of Binary Data in XML
specification
http://www.w3.org/TR/xml-media-types

Copyright © 2005 World Wide Web Consortium,

(Massachusetts Institute of Technology, European Research Consortium for
Informatics and Mathematics, Keio University). All Rights Reserved. This
work is distributed under the W3C® Software License [1] in the hope that
it will be useful, but WITHOUT ANY WARRANTY; without even the implied
warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[1] http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231

$Id: xmlmime.xsd,v 1.1 2005/04/25 17:08:35 hugo Exp $
-->
<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>

```

### 4.1.5.2. *envelope.xsd*

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:env="http://www.w3.org/2003/05/soap-envelope"
targetNamespace="http://www.w3.org/2003/05/soap-envelope" elementFormDefault="qualified" version="1.0">
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:element name="Body" type="env:Body"/>
  <xs:element name="Envelope" type="env:Envelope"/>
  <xs:element name="Fault" type="env:Fault"/>
  <xs:element name="Header" type="env:Header"/>
  <xs:element name="NotUnderstood" type="env:NotUnderstoodType"/>
  <xs:element name="Upgrade" type="env:UpgradeType"/>
  <xs:complexType name="Fault">
    <xs:sequence>
      <xs:element name="Code" type="env:faultcode"/>
      <xs:element name="Reason" type="env:faultreason"/>
      <xs:element name="Node" type="xs:anyURI" minOccurs="0"/>
      <xs:element name="Role" type="xs:anyURI" minOccurs="0"/>
      <xs:element name="Detail" type="env:detail" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="faultcode">

```

```

        <xs:sequence>
            <xs:element name="Value" type="xs:QName"/>
            <xs:element name="Subcode" type="env:subcode" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="subcode">
        <xs:sequence>
            <xs:element name="Value" type="xs:QName"/>
            <xs:element name="Subcode" type="env:subcode" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="faultreason">
        <xs:sequence>
            <xs:element name="Text" type="env:reasontext" maxOccurs="unbounded"/>
        </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 namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="Envelope">
        <xs:sequence>
            <xs:element name="Header" type="env:Header" minOccurs="0"/>
            <xs:element name="Body" type="env:Body"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="Header">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
        <xs:anyAttribute namespace="##other" processContents="skip"/>
    </xs:complexType>
    <xs:complexType name="Body">
        <xs:sequence>
            <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </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 name="SupportedEnvelope" type="env:SupportedEnvType" maxOccurs="unbounded"/>
        </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>

```

#### 4.1.5.3. domibus-backend.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="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/">
<xsd:import namespace="http://www.w3.org/2005/05/xmlmime"/>
<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:element name="FaultDetail">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="code" type="xsd:string"/>
      <xsd:element name="message" nillable="true" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveMessageRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageID" type="tns:max255-non-empty-string" nillable="true"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="retrieveMessageResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" name="bodyload" type="tns:LargePayloadType"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="payload" type="tns:LargePayloadType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="listPendingMessagesRequest" type="xsd:anyType" nillable="false"/>
<xsd:element name="listPendingMessagesResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="messageID" nillable="true" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="messageErrorsRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageID" type="tns:max255-non-empty-string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="messageStatusRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageID" type="tns:max255-non-empty-string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="submitRequest">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element minOccurs="0" name="bodyload" type="tns:LargePayloadType"/>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="payload" nillable="true" type="tns:LargePayloadType"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="submitResponse">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element maxOccurs="unbounded" minOccurs="0" name="messageID" nillable="true" type="xsd:string"/>

```

```

    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

<xsd:complexType name="PayloadType">
  <xsd:simpleContent>
    <xsd:extension base="ns1:base64Binary">
      <xsd:attribute name="payloadId" type="xsd:token" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:complexType name="LargePayloadType">
  <xsd:sequence>
    <xsd:element name="value" type="xsd:base64Binary" ns1:expectedContentTypes="application/octet-stream"></xsd:element>
  </xsd:sequence>
  <xsd:attribute name="payloadId" type="xsd:token" use="required"/>
  <xsd:attribute name="contentType" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="errorResultImpl">
  <xsd:sequence>
    <xsd:element minOccurs="0" name="errorCode" type="tns:errorCode"/>
    <xsd:element minOccurs="0" name="errorDetail" type="xsd:string"/>
    <xsd:element minOccurs="0" name="messageInErrorId" type="xsd:string"/>
    <xsd:element minOccurs="0" name="mshRole" type="tns:mshRole"/>
    <xsd:element minOccurs="0" name="notified" type="xsd:dateTime"/>
    <xsd:element minOccurs="0" name="timestamp" type="xsd:dateTime"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="PayloadURLType">
  <xsd:simpleContent>
    <xsd:extension base="xsd:string">
      <xsd:attribute name="payloadId" type="xsd:token" use="required"/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>

<xsd:simpleType name="messageStatus">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="READY_TO_SEND"/>
    <xsd:enumeration value="READY_TO_PULL"/>
    <xsd:enumeration value="BEING_PULLED"/>
    <xsd:enumeration value="SEND_ENQUEUED"/>
    <xsd:enumeration value="SEND_IN_PROGRESS"/>
    <xsd:enumeration value="WAITING_FOR_RECEIPT"/>
    <xsd:enumeration value="ACKNOWLEDGED"/>
    <xsd:enumeration value="ACKNOWLEDGED_WITH_WARNING"/>
    <xsd:enumeration value="SEND_ATTEMPT_FAILED"/>
    <xsd:enumeration value="SEND_FAILURE"/>
    <xsd:enumeration value="NOT_FOUND"/>
    <xsd:enumeration value="WAITING_FOR_RETRY"/>
    <xsd:enumeration value="RECEIVED"/>
    <xsd:enumeration value="RECEIVED_WITH_WARNINGS"/>
    <xsd:enumeration value="DELETED"/>
    <xsd:enumeration value="DOWNLOADED"/>
  </xsd:restriction>
</xsd:simpleType>

<xsd:simpleType name="errorCode">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="EBMS_0001"/>
    <xsd:enumeration value="EBMS_0002"/>
    <xsd:enumeration value="EBMS_0003"/>
    <xsd:enumeration value="EBMS_0004"/>
    <xsd:enumeration value="EBMS_0005"/>
    <xsd:enumeration value="EBMS_0006"/>
    <xsd:enumeration value="EBMS_0007"/>
    <xsd:enumeration value="EBMS_0008"/>
    <xsd:enumeration value="EBMS_0009"/>
    <xsd:enumeration value="EBMS_0010"/>
  </xsd:restriction>

```

```

<xsd:enumeration value="EBMS_0011"/>
<xsd:enumeration value="EBMS_0101"/>
<xsd:enumeration value="EBMS_0102"/>
<xsd:enumeration value="EBMS_0103"/>
<xsd:enumeration value="EBMS_0201"/>
<xsd:enumeration value="EBMS_0202"/>
<xsd:enumeration value="EBMS_0301"/>
<xsd:enumeration value="EBMS_0302"/>
<xsd:enumeration value="EBMS_0303"/>
<xsd:enumeration value="EBMS_0020"/>
<xsd:enumeration value="EBMS_0021"/>
<xsd:enumeration value="EBMS_0022"/>
<xsd:enumeration value="EBMS_0023"/>
<xsd:enumeration value="EBMS_0030"/>
<xsd:enumeration value="EBMS_0031"/>
<xsd:enumeration value="EBMS_0040"/>
<xsd:enumeration value="EBMS_0041"/>
<xsd:enumeration value="EBMS_0042"/>
<xsd:enumeration value="EBMS_0043"/>
<xsd:enumeration value="EBMS_0044"/>
<xsd:enumeration value="EBMS_0045"/>
<xsd:enumeration value="EBMS_0046"/>
<xsd:enumeration value="EBMS_0047"/>
<xsd:enumeration value="EBMS_0048"/>
<xsd:enumeration value="EBMS_0049"/>
<xsd:enumeration value="EBMS_0050"/>
<xsd:enumeration value="EBMS_0051"/>
<xsd:enumeration value="EBMS_0052"/>
<xsd:enumeration value="EBMS_0053"/>
<xsd:enumeration value="EBMS_0054"/>
<xsd:enumeration value="EBMS_0055"/>
<xsd:enumeration value="EBMS_0060"/>
<xsd:enumeration value="EBMS_0065"/>
</xsd:restriction>
</xsd:simpleType>
<xsd:simpleType name="mshRole">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="SENDING"/>
    <xsd:enumeration value="RECEIVING"/>
  </xsd:restriction>
</xsd:simpleType>
<xsd:complexType final="#all" name="errorResultImplArray">
  <xsd:sequence>
    <xsd:element maxOccurs="unbounded" minOccurs="0" name="item" nillable="true" type="tns:errorResultImpl"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:element name="getStatusRequest" nillable="true">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageID" type="tns:max255-non-empty-string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="statusRequest" nillable="true">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageID" type="tns:max255-non-empty-string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getStatusResponse" nillable="true" type="tns:messageStatus"/>
<xsd:element name="getErrorsRequest" nillable="true">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:element name="messageID" type="tns:max255-non-empty-string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>
<xsd:element name="getMessageErrorsResponse" nillable="true" type="tns:errorResultImplArray"/>
</xsd:schema>

```

#### 4.1.5.4. domibus-header.xsd

```

<?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" minOccurs="0"/>
    </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" minOccurs="0"/>
    </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="optional"/>
  <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>

```

## 5. ANNEXE 1 – DOCUMENT PARTS



(eDelivery)(AP)(ICD)(WS plugin)(1.10).vsd

## 6. LIST OF FIGURES

|  |    |
|--|----|
| Figure 1 - Backend C1 Use cases diagram .....                    | 9  |
| Figure 2 - backend C4 Use cases diagram .....                    | 10 |
| Figure 3 – State machine of C2.....                              | 11 |
| Figure 4 - State machine of C3 .....                             | 11 |
| Figure 5 - Sequence Diagram C1 to C2 – SubmitMessage.....        | 12 |
| Figure 6 - Sequence Diagram C4 to C3 – RetrieveMessage .....     | 23 |
| Figure 7 - Sequence Diagram – GetStatus.....                     | 29 |
| Figure 8 - Sequence Diagram C4 to C3 – ListPendingMessages ..... | 32 |
| Figure 9 - WSDL model for Domibus 4.x.y.....                     | 34 |
| Figure 10 – MessageInfo type .....                               | 35 |
| Figure 11 - The From - To PartyInfo .....                        | 36 |
| Figure 12 – PartyInfo type .....                                 | 37 |
| Figure 13 – CollaborationInfo type.....                          | 39 |
| Figure 14 – MessageProperties type.....                          | 42 |
| Figure 15 – PayloadInfo type.....                                | 43 |
| <br>   |    |
| List of TablesTable 1 - Actor list.....                          | 9  |
| Table 2 - C2 Use cases .....                                     | 9  |
| Table 3 - C3 Use cases .....                                     | 10 |

## 7. CONTACT INFORMATION

### CEF Support Team

**By email:** CEF-EDELIVERY-SUPPORT@ec.europa.eu

**By phone:** +32 2 299 09 09

- **Standard Service:** 8am to 6pm (Normal EC working Days)
- **Standby Service\*:** 6pm to 8am (Commission and Public Holidays, Weekends)

**\* Only for critical and urgent incidents and only by phone**

---