

Message Control and Choreography (MCC) Profile-ebMS-V3

Release 11.00.00A

Specification Information	
Name	MCC – Profile-ebMS V3
Publication Date	1 June 2010
Version Identifier	Release 11.00.00A

Table of Content

1.	Document Management	4
1.1	Legal Disclaimer.....	4
1.2	Copyright.....	4
1.3	Trademarks.....	4
1.4	Acknowledgments	5
1.5	Related Documents	5
1.6	Document Version History	5
1.7	Document Purpose	5
2.	Single Business Document PIP Profiling for ebMS V3	6
2.1	PIP Definition Features.....	7
2.1.1	Parties involved.....	7
2.1.2	Business Document.....	7
2.1.3	Business State Alignment features	8
2.2	PIP execution outcome.....	10
2.3	Quality of Service features	10
2.3.1	Security options	10
2.3.2	Reliable Messaging.....	12
2.3.3	Timing Constraints.....	12
3.	PIP Parameterization and Execution Control.....	13
3.1	PIP Property Parameters	14
3.2	PIP execution modes and related parameters	16
3.2.1	Messaging Protocols.....	16
3.2.2	Message Exchange Patterns	16
3.3	PIP Instance Correlation and Identification.....	17

3.3.1	PIP Identification	17
3.3.2	Message Correlation.....	17
4.	Use Cases of PIP definition	18
4.1	Use Case 1 – Full features	18
4.2	Use Case 2 – Business Document Only.....	19
4.3	Sample Message Exchange	20
4.3.1	1-Action Message	20
4.3.2	Response Message.....	21

1. Document Management

1.1 Legal Disclaimer

RosettaNet, its members, officers, directors, employees, or agents shall not be liable for any injury, loss, damages, financial or otherwise, arising from, related to, or caused by the use of this document or the specifications herein, as well as associated guidelines and schemas. The use of said specifications shall constitute your express consent to the foregoing exculpation.

1.2 Copyright

©2010 RosettaNet. All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the inclusion of this copyright notice. Any derivative works must cite the copyright notice. Any public redistribution or sale of this publication or derivative works requires prior written permission of the publisher.

1.3 Trademarks

RosettaNet, Partner Interface Process, PIP and the RosettaNet logo are trademarks or registered trademarks of "RosettaNet," a non-profit organization. All other product names and company logos mentioned herein are the trademarks of their respective owners. In the best effort, all terms mentioned in this document that are known to be trademarks or registered trademarks have been appropriately recognized in the first occurrence of the term.

1.4 Acknowledgments

This document has been prepared by RosettaNet (<http://www.rosettanet.org/>) from requirements gathered during the Milestone Program and in conformance with the methodology. Listed below are the legal entities that contributed to the design and development of this PIP.

Axway	Cisco
DHL	IBM
KJC Solutions	Oracle
OASIS	Software AG
Tibco	University Bamberg
Vienna University of Technology	

1.5 Related Documents

- MCC Single Business Document PIP Template R11.00.00A
- MMS ebMS V3 Profile R12.00.00A

1.6 Document Version History

<u>Version</u>	<u>Date</u>	<u>Description</u>
Release 11.00.00A	1 June 2010	Released Version

1.7 Document Purpose

The purpose of the document is to explain the structure, the association between objects, the content of objects and the definition for single elements to a non-technical audience.

2. Single Business Document PIP Profiling for ebMS V3

The “Single business document PIP Template” [MCC-PIP-Template] defines a model for single business document PIPs. It is abstract in two different ways:

1. The realization of a PIP definition component may vary with the communication technology selected for implementing the PIP.
2. The realization of a PIP definition may vary depending on the execution context assumed.

This template can be used to define PIPs (PIP definitions) in a way that does not depend on specific messaging solutions and protocols.

In a next step, such PIP definitions need to be implemented as “concrete PIPs” or customized PIPs, which defines all execution details including the messaging solution to be used. This profiling document is concerned with defining how the different features and execution aspects of a PIP definition will map to the ebMS V3 messaging solution.

To summarize, this profile is relevant to the two lowest levels at which PIP material is defined:

- (1) **Customized PIP:** (or concrete PIP): At this level, all elements of a PIP are fully defined, and all parameters (such as QoS, timing) are given a specific value or specific range that is agreed between partners. The execution of such PIPs is determined in terms of QoS, alignment features and execution mode. The factors that condition a successful or a failed outcome are fully determined and known from partners.
- (2) **PIP instance:** This is an image of a particular execution of a PIP, i.e. a particular sequence of concrete messages where all components and PIP properties are given a value – e.g. a fully defined business document between two identified partners, a particular timing between these messages, etc.

2.1 PIP Definition Features

2.1.1 Parties involved

Party IDs and roles are represented in the message header. See Profiling requirements described in the MMS ebMS V3 Profile, section 4.2. "ebMS Header Profiling".

In particular, the following items describing parties are represented, and mapped to related RNIF document header elements:

- Party IDs (section 4.2.1)
- Party roles (section 4.2.2)
- Service invoked (section 4.2.3)
- Action invoked (section 4.2.4)
- Conversation Id (section 4.2.5)

2.1.2 Business Document

The Rosettanet payload may either be packaged in the SOAP Body or as an attachment. In both cases the referencing from the ebMS header must follow the MMS profile requirements:

Specification Header element:
Feature

eb:UserMessage/eb:PayloadInfo/eb:PartInfo

eb:UserMessage/eb:PayloadInfo/eb:PartInfo/eb:Schema

Specification ebMS 3, section 5.2.2.12, 5.2.2.13
Reference

Profiling Element eb:PartInfo: SHOULD include an eb:Schema element when the eb:PartInfo element is referring to the service content part (main XML document) of a PIP payload, or is referring to an XML RosettaNet signal. When present it MUST use an URN identifying the schema or DTD that applies to the part. The schema URN identifier MUST comply with **[RN-NameSpaces]**.

Example for a PIP service content with XML schema:

```
urn:rosettanet:specification:interchange:PIP3A4PurchaseO
```

```
OrderRequest:xsd:schema:1.0
```

When a legacy RNIF header (such as Service header) is included in the message, it must be added as a single attachment. The eb:Reference element SHOULD contain an eb:Schema element to identify it, which conforms to **[RN-NameSpaces]**.

Example for a Service header:

```
urn:rosettanet:specification:system:ServiceHeader:dtd:schema:2.0
```

NOTE: The use of an XLINK processor should not be required.

Alignment

Test

References

In case the SBDH needs be preserved as is – e.g. for reuse of legacy back-end integration -, then the entire SBDH header can be added as a distinct payload part (referenced by a separate eb:PartInfo element in the header). It is RECOMMENDED to add it in the SOAP Body instead of as an attachment.

2.1.3 Business State Alignment features

(1) Delivery Alignment

The Reliable Messaging (RM) feature of ebMS V3 (WS-ReliableMessaging used in compliance with WS-I Reliable and Secure Profile) provides state alignment about message **reception**. There is no explicit signaling confirming that a message has actually been delivered to the application. However, only reception failures are expected to be reported on the Sender side.

See in the MMS ebMS V3 Profile, section 6.4.1 for the use of RM for state alignment, in case Receipts are not used.

Unlike the RM acknowledgement, the ebMS Receipt signal provides a positive acknowledgement for delivery alignment, that has visibility at application level on the Sender side.

Relevant PMode parameters: **PMode[1].Reliability** and below.

(2) Validity Alignment

The ebMS Receipt signal is generally sent back by the MSH before payload validation occurs. In general it cannot be counted on to implement this semantics. However, the ebMS Receipt can be given a Non-Repudiation of Receipt semantics, in which case MMS ebMS V3 profile offers these options:

- **Simple non-repudiation:** In this variant, the signed eb:Receipt is sent back before document validation occurs. The eb:Receipt only means that the message has been well received and that the receiving endpoint is taking responsibility for further processing (including payload validation).
- **Validating non-repudiation:** In this variant, the signed eb:Receipt is sent back only after the document validation occurs. The eb:Receipt means that the message has been well received and that it is considered as valid for further business processing.

See the MMS ebMS V3 profile for more details.

- Relevant PMode parameters: **PMode[1].Security.SendReceipt** and below.

2.2 PIP execution outcome

As a general rule, a positive outcome (success) will be manifest on the Sender side with the ebMS protocol as (a) reception of expected Receipt(s) or acknowledgements, (b) no critical ebMS error being generated.

Some failures such as timeout failures for Receipts and for overall PIP execution, will however not be detected by the ebMS V3 layer and must be established at the layer above.

2.3 Quality of Service features

2.3.1 Security options

Authentication:

- in ebMS V3, supported by WS-Security. Profiling requirements described in the MMS ebMS V3 Profile, section 7.1 "General Security Policies" apply.

See PMode parameters:

- **PMode[1].Security.X509.Sign**
- **PMode[1].Security.X509.Signature**

Confidentiality:

- in ebMS V3, supported by WS-Security. Profiling requirements described in the MMS ebMS V3 Profile, section 7.1 "General Security Policies" apply.

See PMode parameters:

- **PMode[1].Security.X509.Encryption**

Integrity:

- in ebMS V3, supported by WS-Security. Profiling requirements described in the MMS ebMS V3 Profile, section 7.1 "General Security Policies" apply.

Non Repudiation/Non Repudiation Of Receipt:

- in ebMS V3: Non Repudiation or Receipt is supported by the ebMS V3 protocol with Receipt messages containing digests, as described in Core ebMS V3. Profiling requirements described in the MMS ebMS V3 Profile, section 6.3 "Receipt Semantics" and section 7.2 "Handling of Receipts" apply. Two flavors of NRR are supported by the MMS profile: Simple and Validating NRR.
- NRR SHOULD be accomplished by generating Receipts that comply with the AS4 Profile (Section 4.1.8 in the AS4 Profile specification [...]).

See PMode parameters:

- **PMode[1].Security.X509** (for signature of the action message)
- **PMode[1].Security.SendReceipt**

Authorization:

- in ebMS V3, supported by WS-Security. Also allows for fine-grain access control (e.g. to specific Service/Action or specific MPC), based on the "message authorization" feature (section 7.10 in ebMS V3 standard). Profiling requirements described in the MMS ebMS V3 Profile, section 7.1 "General Security Policies" apply. Authorization of pulling is described in 7.1.2.

See PMode parameters:

- **PMode[1].Security.PModeAuthorize**
- **PMode.Responder.Authorization**

2.3.2 Reliable Messaging

Guaranteed delivery (At-least-Once delivery): Supported by the reliable messaging feature (WS-ReliableMessaging) in ebMS V3. Profiling requirements described in the MMS ebMS V3 Profile, section 7.1 “General Reliability Policies” apply: duplicate elimination is expected to be used when guaranteed delivery is used.

- **PMode[1].Reliability.AtLeastOnce.Contract** (true/false, for “guaranteed delivery”)
- **PMode[1].Reliability.AtLeastOnce.Contract.AckOnDelivery** (false= Ack on reception by gateway, true= Ack on delivery to the application layer. Usually constrained by the implementation.)
- **PMode[1].Reliability.AtLeastOnce.Contract.AcksTo**
- **PMode[1].Reliability.AtLeastOnce.Contract.AckResponse**
- **PMode[1].Reliability.AtLeastOnce.ReplyPattern**

Duplicate elimination (At-Most-Once delivery): Supported by the reliable messaging feature (WS-ReliableMessaging) in ebMS V3.

- **PMode[1].Reliability.AtMostOnce.Contract**

2.3.3 Timing Constraints

Time to acknowledge validity (or invalidity): There is no capability in the ebMS V3 protocol for detecting timeouts. Such timeouts must be detected at the layer above the messaging layer.

Time to Perform: There is no capability in the ebMS V3 protocol for detecting timeouts. Such timeouts must be detected at the layer above the messaging layer.

3. PIP Parameterization and Execution Control

Parameterization of the messaging behavior for PIP definitions as well as PIP instance customization, is represented by PMode abstract parameters. These have been classified in two categories in the PIP Template document:

1. **PIP property parameters**
2. **PIP execution parameters**

3.1 PIP Property Parameters

The following parameters are configurable on a PIP definition and a PIP implementation instance basis:

Specification item	Configurable	Implication	Explanation
Send Request Document	No		(part of the PIP definition) A request document always has to be sent. The MSH does not itself create/controls the document and its sending. (Identified with GlobalBusinessActionCode)
Overall Time-To-Perform	No	Not ebMS configurable.	Time for performing the messaging Technology-specific PIP protocol. Controlled at a higher level than MSH.
Receipt Acknowledgement	Yes	Controlled by ebMS V3 PMode: PMode[1].Security.SendReceipt	Use of ebMS Receipt signal.
Non-Repudiation of-origin	Yes		Controlled by Digital signature of action message. Requires persistence of the message.
Non-Repudiation of-Receipt	Yes	Controlled by ebMS V3 PMode: See MMS ebMS V3 Profile, section 6.6.2 See PMode[1].Security.SendReceipt	Based on ebMS Receipt signal.
TimeTo Acknowledge Receipt	No	Sending a ReceiptAcknowledgement is controlled by PMode[1].Security.SendReceipt Timing is not ebMS configurable.	Time for sending a ValidityAcknowledgement measured from the receipt of the action message.
Reliability	Yes	ebMS V3 PMode: see PMode[1].Reliability	

Specification item	Configurable	Implication	Explanation
Confidentiality	Yes	ebMS V3 PMode: see PMode[1].Security	
Integrity	Yes	ebMS V3 PMode: see PMode[1].Security	
Authentication	Yes	ebMS V3 PMode: see PMode[1].Security	
Authorization	Yes	ebMS V3 PMode: see PMode[1].Security and PMode.Responder.Authorization	Possibility to control what message header content is allowed for which sender. (which PMode can be used by which user, which MPC channel can be pulled by which user)
IntelligibleCheck Required	No	Sending an Acceptance Acknowledgement reflecting business rule and semantic checks. Sending a ReceiptAcknowledgement reflecting Message and possibly Document syntax checks.	Integration partners have to define the additional validation steps that have to be performed in case this flag is used.
RetryCount	No	Reliable Messaging feature is not a substitute for this level of retry. Message-layer retries are handled by RM feature.	Describes how often a business document/signal can be submitted by the PIP process engine to the messaging layer. (would belong to a new PIP instance identifier) This is not a feature at the level of Reliable Messaging.

Examples for defining PIPs will be given in the use cases section.

3.2 PIP execution modes and related parameters

3.2.1 Messaging Protocols

Specifying the protocol in use, see PMode parameter: **PMode[1].Protocol.Address**

Specifying the SOAP version in use, see PMode parameter: **PMode[1].Protocol.SOAPVersion:**

3.2.2 Message Exchange Patterns

Synchronous execution

For ebMS V3: In MMS ebMS V3 Profile:

See Section 6.4.1: One-action PIP without Non-Repudiation of Receipt

See Section 6.4.2: One-action PIP, with Simple Non-Repudiation of Receipt

See Section 6.4.3: One-action PIP, with Validating Non-Repudiation of Receipt

Also: "Pure client" cases in 6.5.1, 6.5.2 and 6.5.3.

Asynchronous execution with callback

See Section 6.4.4: One-action PIP, with Callback Receipt for Non-Repudiation

Asynchronous execution with pulling

See Section 6.5.4: One-action PIP, with Pulled Receipt for Validating Non-Repudiation

3.3 PIP Instance Correlation and Identification

3.3.1 PIP Identification

Generation of Globally Unique Ids (GUIDs) for PIP instances

See section 4.2.5 MMS ebMS V3 Profile:

The ConversationId header element represents the PIP instance ID value,
It MUST map to Standard Business Doc Header (SBDH) element when applicable:
RequestingDocumentInformation / BusinessProcessInstanceIdentifier

It MUST map to (in RNIF Service header) element:
ServiceHeader/ProcessControl/pipInstanceId/InstanceIdentifier

Inclusion of PIP instance GUIDs within RosettaNet message definitions

In ebMS V3: See section 4.2.5 MMS ebMS V3 Profile

It is RECOMMENDED that ConversationId header element represents the PIP instance ID value, i.e. has same value for all messages related to the same PIP instance.

In other words, messages from the same PIP instance MUST have same ConversationID, and it is recommended that this ConversationID be unique to this PIP instance (not shared with other PIP instances).

3.3.2 Message Correlation

ebMS V3: See section 4.2.6 MMS ebMS V3 Profile:

Every message involved in a PIP instance MUST refer to another previous message of this instance with RefToMessageId header element (except for the initial message of the instance, which MUST NOT have a RefToMessageId element.)

4. Use Cases of PIP definition

This section gives some sample configurations of PIPs according to the configurability matrix above. The MCC messaging technology profiles are expected to describe the implementation of these use cases.

4.1 Use Case 1 – Full features

```

<DataExchange
  name="bt-PIP3A20"
  nameID="bt-PIP3A20"
  isGuaranteedDeliveryRequired="true">
  <RequestingRole name="Purchase Order Confirmation Sender" nameID="bt-
PIP3A20-role-sender" />
  <RespondingRole name="Purchase Order Confirmation Receiver"
nameID="bt-PIP3A20-role-receiver" />
  <RequestingBusinessActivity
    name="Send Purchase Order Confirmation"
    nameID="bt-PIP3A20-ba-req"
    isIntelligibleCheckRequired="true"
    isNonRepudiationRequired="true"
    isNonRepudiationReceiptRequired="true"
    retryCount="3"
    timeToAcknowledgeReceipt="PT3M"
  >
  <DocumentEnvelope
    name="doc-PIP3A20-PurchaseOrderConfirmation"
    businessDocumentRef="doc-PIP3A20-PurchaseOrderConfirmation"
    nameID="doc-PIP3A20-PurchaseOrderConfirmation-de"
    isAuthenticated="transient"
    isConfidential="transient"
    isTamperDetectable="transient"
  />
  <ReceiptAcknowledgement
    name="ra"
    nameID="bt-PIP3A20-ack-ra"
    signalDefinitionRef="ra2" />
  <ReceiptAcknowledgementException
    name="rae"
    nameID="bt-PIP3A20-ack-rae"
    signalDefinitionRef="rae2" />
  </RequestingBusinessActivity>
  <RespondingBusinessActivity name="xsd-pacifier" nameID="bt-PIP3A20-ba-
resp" />
</DataExchange>

```

4.2 Use Case 2 – Business Document Only

```
<DataExchange
  name="bt-PIP3A20"
  nameID="bt-PIP3A20"
  isGuaranteedDeliveryRequired="true">
  <RequestingRole name="Purchase Order Confirmation Sender" nameID="bt-
PIP3A20-role-sender" />
  <RespondingRole name="Purchase Order Confirmation Receiver"
nameID="bt-PIP3A20-role-receiver" />
  <!-- No TTAR, nor isIntelligibleCheckRequired -->
  <RequestingBusinessActivity
    name="Send Purchase Order Confirmation"
    nameID="bt-PIP3A20-ba-req"
    isNonRepudiationRequired="true"
    isNonRepudiationReceiptRequired="true"
    retryCount="1"
  >
  <DocumentEnvelope
    name="doc-PIP3A20-PurchaseOrderConfirmation"
    businessDocumentRef="doc-PIP3A20-PurchaseOrderConfirmation"
    nameID="doc-PIP3A20-PurchaseOrderConfirmation-de"
    isAuthenticated="transient"
    isConfidential="transient"
    isTamperDetectable="transient"
  />
  <!-- No ReceiptAcknowledgement/Exception definitions here -->
</RequestingBusinessActivity>
  <RespondingBusinessActivity name="xsd-pacifier" nameID="bt-PIP3A20-ba-
resp" />
</DataExchange>
```

4.3 Sample Message Exchange

4.3.1 1-Action Message

```

<S11:Envelope xmlns:S11="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eb="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/">
<S11:Header>
<eb:Messaging S11:mustUnderstand="1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/ns/core/200704/http://docs.oasis-open.org/ebxml-
msg/ebms/v3.0/core/ebms-header-3_0-200704.xsd"> <eb:UserMessage>
<eb:MessageInfo>
  <eb:Timestamp>2006-07-25T12:19:05</eb:Timestamp>
  <eb:MessageId>12345@requester.example.com</eb:MessageId>
</eb:MessageInfo>
<eb:PartyInfo>
  <eb:From>
    <eb:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:D-
U-N-SNumber:0060">123456789</eb:PartyId>
    <eb:Role>bt-PIP3A20-role-sender</eb:Role>
  </eb:From>
  <eb:To>
    <eb:PartyId tp:type="urn:oasis:names:tc:ebxml-cppa:partyid-type:D-
U-N-SNumber:0060">112233445</eb:PartyId>
    <eb:Role>bt-PIP3A20-role-receiver</eb:Role>
  </eb:To>
</eb:PartyInfo>
<eb:CollaborationInfo>
  <eb:AgreementRef>http://registry.example.com/cpa/123456
  </eb:AgreementRef>
  <eb:Service>urn:rosettanet:specification:interchange:bt-
PIP3A20.xml:ebbp:v11_00</eb:Service>
  <eb:Action>doc-PIP3A20-PurchaseOrderConfirmation-de</eb:Action>
  <eb:ConversationId>4321</eb:ConversationId>
</eb:CollaborationInfo>
<eb:PayloadInfo>
  <eb:PartInfo href="cid:part@example.com">
    <eb:Schema location="http://registry.example.org/po.xsd"
version="2.0"/>
    <eb:PartProperties>
      <eb:Property name="Description">Purchase Order
Confirmation</eb:Property>
      <eb:Property name="MimeType">application/xml</eb:Property>
    </eb:PartProperties>
  </eb:PartInfo>
</eb:PayloadInfo>
</eb:UserMessage>
</eb:Messaging>
</S11:Header>
<S11:Body>
...
</S11:Body>
</S11:Envelope>

```

4.3.2 Response Message

In this exchange NRR is required, so the ebMS Receipt contains a NonRepudiationInformation element, which contains a sequence of MessagePartNRInformation items for each message part for which evidence of non repudiation of receipt is being provided. In the normal default usage, these message parts are those that have been signed in the original message. Each message part is described with information defined by an XML Digital Signature Reference information item. The following example illustrates the ebMS V3 Signal Message header.

```
<eb3:Messaging Soap12:mustUnderstand="true" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="ValueOfMessagingHeader">
<eb3:SignalMessage>
  <eb3:MessageInfo>
    <eb3:Timestamp>2009-11-06T08:00:09Z</eb3:Timestamp>
    <eb3:MessageId>orderreceipt1234@seller.com</eb3:MessageId>
    <eb3:RefToMessageId>12345@requester.example.com </eb3:RefToMessageId>
  </eb3:MessageInfo>
  <eb3:Receipt>
    <ebbp:NonRepudiationInformation>
      <ebbp:MessagePartNRInformation>
        <dsig:Reference URI="#5cb44655-5720-4cf4-a772-19cd480b0ad4">
          <dsig:Transforms>
            <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          </dsig:Transforms>
          <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
/>
          <dsig:DigestValue>o9QDCwWSiGVQACEsJH5nqkVE2s0=</dsig:DigestValue>
        </dsig:Reference>
      </ebbp:MessagePartNRInformation>
      <ebbp:MessagePartNRInformation>
        <dsig:Reference URI="cid:a1d7fdf5-d67e-403a-ad92-3b9deff25d43@buyer.com">
          <dsig:Transforms>
            <dsig:Transform Algorithm="http://docs.oasis-open.org/wss/oasis-wss-SwAProfile-
1.1#Attachment-Content-Signature-Transform" />
          </dsig:Transforms>
          <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          <dsig:DigestValue>iWNSv2W6SxbOYZliPzZDCXAxrwl=</dsig:DigestValue>
        </dsig:Reference>
      </ebbp:MessagePartNRInformation>
    </ebbp:NonRepudiationInformation>
  </eb3:Receipt>
</eb3:SignalMessage>
</eb3:Messaging>
```

For a signed receipt, a Web Services Security header signing over (at least) the signal header is required. An example WS-Security header is as follows :

```
<wsse:Security s:mustUnderstand="1" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:s="http://www.w3.org/2003/05/soap-envelope">
  <wsu:Timestamp wsu:Id="_1" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsu:Created>2009-11-06T08:00:10Z</wsu:Created>
<wsu:Expires>2009-11-06T08:50:00Z</wsu:Expires>
</wsu:Timestamp>
<wsse:BinarySecurityToken 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#X509v3" wsu:Id="_2"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd">MIIFADCCBGmgAwIBAgIEOmitted</wsse:BinarySecurityToken>
<ds:Signature Id="_3" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
/>
<ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
<ds:Reference URI="#ValueOfMessagingHeader">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
<InclusiveNamespaces PrefixList="xsd" xmlns="http://www.w3.org/2001/10/xml-exc-
c14n#" />
</ds:Transform>
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
<ds:DigestValue>ZXnOmitted=</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>rxap4of8JCpUkOmitted=</ds:SignatureValue>
<ds:KeyInfo>
<wsse:SecurityTokenReference xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
  <wsse:Reference URI="#_2" ValueType="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3" />
</wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
```