External Interface Specification

18.1.2019
Introduction ........................................................................................................................................... 6

1.1 Purpose ........................................................................................................................................... 6

1.2 Scope ............................................................................................................................................. 6

1.3 Target Audience ................................................................................................................................. 6

1.4 Document Structure .............................................................................................................................. 6

1.5 Document References .......................................................................................................................... 7

1.6 Glossary ............................................................................................................................................. 7

1.7 Symbols ............................................................................................................................................. 7

1.8 Open Issues ....................................................................................................................................... 7

Channel for MarketMessaging .................................................................................................................. 9

2.1 Interface overview ................................................................................................................................. 9

2.1.1 Asynchronous Processing ............................................................................................................... 9

2.1.2 Synchronous Processing ............................................................................................................... 11

2.1.3 Fetching Notifications .................................................................................................................... 11

2.2 Generic interface functionality ............................................................................................................. 12

2.2.1 Transport level ................................................................................................................................. 14

2.2.1.1 Transport level Error codes ....................................................................................................... 14

2.2.2 B2B service operation ....................................................................................................................... 14

2.2.2.1 B2B service operation Error codes .......................................................................................... 15

2.3 Exchanging Business Messages .......................................................................................................... 16

2.3.1 XML Business Messages in the Payload ......................................................................................... 17

2.4 System Users ...................................................................................................................................... 17
2.5 SOAP Fault details .................................................................................................................. 18
  2.5.1 Classifying Error codes in this document ....................................................................... 18
  2.5.2 SOAP Fault example ..................................................................................................... 19
  2.6 Timestamps ......................................................................................................................... 19

3 B2B Market Messaging services ............................................................................................ 20
  3.1 ProcessMessage operation ............................................................................................... 20
    3.1.1 Request ...................................................................................................................... 20
    3.1.2 Response ................................................................................................................... 20
    3.1.3 Error codes in SOAP Faults ...................................................................................... 21
  3.2 SendMessage operation ..................................................................................................... 21
    3.2.1 Request ...................................................................................................................... 21
    3.2.2 Response ................................................................................................................... 21
    3.2.3 Error codes in SOAP Faults ...................................................................................... 22
  3.3 PeekMessage operation ..................................................................................................... 22
    3.3.1 Request ...................................................................................................................... 22
    3.3.2 Response ................................................................................................................... 22
    3.3.3 Error codes in SOAP Faults ...................................................................................... 23
    3.3.4 Operational guidance ............................................................................................... 23
  3.4 Dequeue Message operation ............................................................................................. 23
    3.4.1 Request ...................................................................................................................... 23
    3.4.2 Response ................................................................................................................... 24
    3.4.3 Error codes in SOAP Faults ...................................................................................... 24

4 Summary ................................................................................................................................. 25
4.1 Summary of SOAP actions................................................................. 25
4.2 Summary of error codes................................................................. 25
5 Fingrid Datahub – WSDL details......................................................... 26
  5.1 Parameters ....................................................................................... 26
  5.2 Connection details ........................................................................... 26
  5.3 Recommendations ........................................................................... 26
  5.4 File descriptions ............................................................................... 26
  5.5 Fingrid Datahub Examples............................................................... 27
    5.5.1 Send Message Request ............................................................... 27
    5.5.2 Send message response ............................................................. 28
    5.5.3 Process message request ............................................................ 28
    5.5.4 Process message response .......................................................... 28
    5.5.5 Peek message request ................................................................. 29
    5.5.6 Peek message response .............................................................. 29
      5.5.6.1 Peek response with no available message in the message queue........... 29
      5.5.6.2 Peek response with available message in the message queue ............... 29
    5.5.7 Dequeue message request ......................................................... 30
    5.5.8 Dequeue message response ....................................................... 30
## Change log

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.01.2019</td>
<td>1.0</td>
<td>First official version</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Purpose
This document contains the external interface specifications for the B2B channel of the Fingrid Datahub. The CGI Central Market System (CMS) is used for implementing the B2B channel.

1.2 Scope
The B2B Market Messaging interfaces are the external interfaces used in a B2B context.

B2B Market Messaging services:

- `SendMessage` operation: interface for Market Participants to send Market Messages/Transactions to the Datahub to have them processed asynchronously;
- `ProcessMessage` operation: interface for Market Participants to process Market Messages/Transactions in the Datahub (synchronous);
- `PeekMessage` operation: interface for Market Participants to retrieve a Market Message/Transactions from the Datahub;
- `DequeueMessage` operation: interface for Market Participants to confirm the acceptance of a received Market Message/transactions from the Datahub.

1.3 Target Audience
The target audience for this document are architects, operators, development- and test teams of using the B2B-SOAP interface of the Fingrid Datahub.

1.4 Document Structure

Information

Chapter 1 contains the general information about this document, including its purpose, short description and references to other documents.

Product Standard:

Chapter 2 gives an overview of B2B channel for the market messaging.

Chapter 3 describes the details of the B2B MarketMessaging services.

Chapter 4 describes the B2B SOAP appendices.
Fingrid specific:

Chapter 5 describes Fingrid Datahub specific details.

1.5 Document References

<table>
<thead>
<tr>
<th>Reference</th>
<th>Document (ID)</th>
<th>Version</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAP</td>
<td><a href="https://www.w3.org/TR/soap12-part1/">https://www.w3.org/TR/soap12-part1/</a></td>
<td>1.2</td>
<td>27-04-2007</td>
</tr>
<tr>
<td>XSD</td>
<td><a href="https://www.w3.org/standards/techs/xmlschema#stds">https://www.w3.org/standards/techs/xmlschema#stds</a></td>
<td>1.1</td>
<td>05-04-2012</td>
</tr>
<tr>
<td>EVENTS</td>
<td>02-04-09-02 – Datahub Events</td>
<td>1.6.2.2</td>
<td>17.12.2018</td>
</tr>
<tr>
<td>SCHEMAS</td>
<td>Zip file: ExternalInterfaceSpecification</td>
<td>1.6.2.2</td>
<td>17-11-2018</td>
</tr>
</tbody>
</table>

1.6 Glossary

<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS</td>
<td>CGI Central Market System / Datahub</td>
</tr>
<tr>
<td>DEQUEUE</td>
<td>Signoff of a retrieved Message</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTTPS</td>
<td>HyperText Transfer Protocol Secure</td>
</tr>
<tr>
<td>PEEK</td>
<td>The retrieval of a Message from the Datahub</td>
</tr>
<tr>
<td>PKI</td>
<td>Public Key Infrastructure</td>
</tr>
<tr>
<td>UML</td>
<td>Unified Modeling Language</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>UUID</td>
<td>Universally unique identifier</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Service Description Language</td>
</tr>
<tr>
<td>XSD</td>
<td>XML Schema Definition Language</td>
</tr>
</tbody>
</table>

1.7 Symbols

The standard UML notation for flow and activity diagrams shall apply.

Text between [] refers to a document in (Document References) Eg [EVENTS]

Text between {} refers to implementation specific parameter in () eg {URL}

1.8 Open Issues

This section contains the identified open issues impacting the External Interface Specification.

<table>
<thead>
<tr>
<th>#</th>
<th>Description</th>
<th>Decision</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Message signing based on WS-Security.</td>
<td>Under discussion in architecture group.</td>
<td>CGI/Fingrid</td>
</tr>
<tr>
<td></td>
<td>Idempotency implementation</td>
<td>Will be part of future CMS product release.</td>
<td>CGI</td>
</tr>
<tr>
<td></td>
<td>Idempotency is replying with the original response message in case of sending a message request with the previous used messageid for same sender, instead of responding with the SOAPFault ‘not unique messageid’</td>
<td>Will impact the situations where MHB.MHD.006 is returned: impact on sections 3.1.3, 3.2.3, 4.2</td>
<td>CGI</td>
</tr>
<tr>
<td>Technical maximum message size for CMS is 100 MB. Business maximum has to be defined by Fingrid.</td>
<td>Under discussion in architecture group.</td>
<td>Fingrid/CGI</td>
<td></td>
</tr>
<tr>
<td>Big inbound messages will give also big outbound messages. Actors could have issues with processing large XML messages. Better to limit to eg 20 MB.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peek message with optional domain not yet possible. With PeekMessage the first available message, irrespective of the message package, will be returned.</td>
<td>Will be part of future CMS product release. Will impact the PeekMessageMessage. This message current does not contain an optional 'Message package' element: 3.3.1</td>
<td>CGI</td>
<td></td>
</tr>
<tr>
<td>HTTP compression</td>
<td>Will be part of future CMS product release.</td>
<td>CGI</td>
<td></td>
</tr>
</tbody>
</table>
2 Channel for MarketMessaging

This chapter gives an overview of the B2B channel for Market Messaging. The Message Hub provides access to the market functions of the Datahub via web services (B2B) and is the gateway for Market Participants to send and receive messages. These services are executed by the use of standardised messages.

This chapter explains common functionality and mechanisms that are the same for the individual service operations.

2.1 Interface overview

Connectivity through the B2B channel is via a single web service using XML representation of one or more transactions in a SOAP v1.2 envelope (Web Service Definition Language). This web service has four generic operations:

- SendMessage (for aSynchronous processing);
- ProcessMessage (for Synchronous processing);
- PeekMessage;
- DequeueMessage.

In the next paragraphs these are described in context of the different processing scenarios.

2.1.1 Asynchronous Processing

- SendMessage - the message and the transaction(s) inside the message are logged, validated, authenticated and authorised and the Market Participant receives a technical acknowledgement via a synchronous response. When the listed checks have passed the transactions are persisted, after which functional processing of the transaction(s) is done asynchronously from the initial Send action. Functional acknowledgements are available after functional processing through the PeekMessage operation.

- PeekMessage - is used by the Market Participant to retrieve the content of the first message available for it. The Market Participant can then process the message asynchronously and once successful can sign off the message using the DequeueMessage operation.

- DequeueMessage - the Market Participant informs the Datahub that the message has been processed by the Market Participant. If a message is not Dequeued any subsequent PeekMessage will return the same message.

The Send/Peek/Dequeue protocol ensures transactional integrity between the Market Participant and the Datahub.

- In terms of bulk receipt of transactions the Datahub allows multiple (*) transactions within a single message. The transactions can be of different types (***) and/or for different (****) data elements.
(*) The number of transactions is limited by:

- The schema defining the maxOccurs
- The B2B channel defining the max Message size.

(**) The schemas define the combinations that that are allowed. Depending on implementation different types are allowed or not.

A single synchronous receipt message is sent back to indicate whether the message was accepted or rejected (such as, due to failing XSD validation). Based on the process description the message is either

- Split up and each transaction is processed individually. Any subsequent functional messages (including error messages) are returned to the sending Market Participant individually; as are any follow up messages sent to other Market Participants.
- Processed as a whole. Any subsequent functional messages (including error messages) are returned to the sending Market Participant as group; as are any follow up messages sent to other Market Participants.
2.1.2 Synchronous Processing

ProcessMessage - to send transactions to the Datahub. The message and the transaction inside the message are logged, validated, authenticated, authorised, persisted and the transaction is functionally processed, after which the Market Participant receives the result in a synchronous response.

2.1.3 Fetching Notifications

The Peek and Dequeue protocol is also used in the context of receiving Notification Messages. (Messages for a Market Participant as result of their role in a Market Process, but not directly related to a ProcessMessage from the Market Participant itself).
2.1.4 Generic interface functionality

Each B2B service operation contains functionality that is the same across the operations. This functionality is described here, so this is not replicated for every specific interface.

The generic functionality consists of:

- Transport level
  - Request size checking
  - Transport level authentication
- B2B service operation
  - B2B service operation level authorisation
  - Audit logging of B2B service operation request and response
  - B2B service operation request syntax validation
The figure above shows the relation between Transport level, Business service operation level and Business service operation implementation, and the steps in each of the areas.

**Fingrid Datahub Oy**

Katuosoite  
Postiosoite  
Puhelin  
Faksi  
Y-tunnus 2745543-5, ALV rek.

etunimi.sukunimi@fingrid.fi  
www.fingrid.fi
The following sections describe these steps in more detail.

**2.1.5 Transport level**

The request is received. In case the size of the request is too large an HTTP response with status code 413 (Content too large) is generated and processing is stopped. See Chapter 4.2 - Summary of error codes for a full list of error responses. The maximum size is configurable, with a maximum of 100 MB.

The authentication of System Users is handled by the Message Hub. The System User is authenticated by a private certificate and can only be used for B2B communication through the web services. If the System User is not authenticated, an HTTP response with status code 401 (Access denied) is generated.

**2.1.5.1. Transport level Error codes**

The default HTTP codes apply in case of HTTPS transport level faults.

The following HTTP status codes are purposely returned based on the above description.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>Security</td>
<td>Access Denied – User authentication based on the provided certificate has failed. (check if user identity is created correctly)</td>
</tr>
<tr>
<td>403</td>
<td>Security</td>
<td>Access Denied – Client certificate is not provided or trusted. (check the certificate in the request)</td>
</tr>
<tr>
<td>404</td>
<td>System</td>
<td>Requested resource not found</td>
</tr>
<tr>
<td>413</td>
<td>System</td>
<td>Content length too large</td>
</tr>
<tr>
<td>500</td>
<td>System</td>
<td>In case of any unidentified errors.</td>
</tr>
</tbody>
</table>

**2.1.6 B2B service operation**

Based on the request the B2B SOAP operation is identified. In case this is an unknown B2B service operation, a SOAP Fault with code ‘MHB.MED.004’ is returned (See Chapter 4.2 - Summary of error codes for a full list of error responses).

The syntax of the SOAP Request is checked. When the message syntax is incorrect this results in a SOAP Fault with code ‘MHB.MHD.001’. Syntax checking includes validating the SOAP message using the XML Schema Definitions (XSD’s) and checking the proper grouping of transaction types within the SOAP message.

All the messages that have been authenticated are logged, without change, into the audit log and can be retrieved through the Party User Interface.

The authorisation of the System User for the specific B2B service operation is determined in two steps:

1. If the System User is not found or is not authorized for this B2B service operation then a SOAP Fault is returned with code ‘MHB.MHD.003’ (user not authorized for system function).
2. If the System User neither matches the Organisation specified as sender in the message (PhysicalSender) nor (one of) the delegated organisation(s) a SOAP Fault is returned with code ‘MHB.MHD.009’ (user not authorized for organisation).

Only if all generic checks and validations succeed the request is executed.

In case of an internal technical error the SOAP Fault with code ‘MHB.MHD.000’ is returned.

### 2.2.2.1 B2B service operation Error codes

The table below provides the error codes that can be returned as a SOAP Fault as part of the generic B2B message handling. Specific B2B message handling is described in chapter 3.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>SOAP Code</th>
<th>Meaning</th>
<th>Operational Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHB.MHD.000</td>
<td>System</td>
<td>Receiver</td>
<td>General Failure</td>
<td>Resend the message, if issues persists contact the Market Operator</td>
</tr>
<tr>
<td>MHB.MHD.001</td>
<td>Syntax</td>
<td>Sender</td>
<td>Message validation failed</td>
<td>Resend corrected message</td>
</tr>
<tr>
<td>MHB.MHD.002</td>
<td>System</td>
<td>Receiver</td>
<td>System configuration error</td>
<td>Resend the message, if issues persists contact the Market Operator</td>
</tr>
<tr>
<td>MHB.MHD.003</td>
<td>Security</td>
<td>Sender</td>
<td>User not authorized for system function (e.g. not found, no rights for the operation or message type, user blocked or inactive)</td>
<td>Check authorisation and contact Market Operator in case of questions. Resend message after correction of authorisation.</td>
</tr>
<tr>
<td>MHB.MHD.004</td>
<td>Security</td>
<td>Sender</td>
<td>Unknown request</td>
<td>Resend corrected message</td>
</tr>
<tr>
<td>MHB.MHD.005</td>
<td>System</td>
<td>Receiver</td>
<td>Back-end timeout</td>
<td>Resend the message, if issues persists contact the Market Operator. The system will prevent processing messages with the same transaction-id twice. If the system responses to the resent message with a MHB.MHD.006 (see chapter 3.1.3) then the system has already processed (or is still processing) the first message.</td>
</tr>
<tr>
<td>MHB.MHD.008</td>
<td>Security</td>
<td>Sender</td>
<td>Message content unsecure</td>
<td>Message content contains unsecure elements (e.g. SQL injection or cross-site scripting). Message must be adjusted before it can be accepted by the system.</td>
</tr>
<tr>
<td>MHB.MHD.009</td>
<td>Security</td>
<td>Sender</td>
<td>User not authorized for organisation (e.g. System User neither matches PhysicalSender nor (one of) the delegated Organisation(s))</td>
<td>Check message header, authorisation and delegation configuration and contact Market Operator in case of questions. Resend message after correction.</td>
</tr>
<tr>
<td>MHB.MHD.010</td>
<td>Syntax</td>
<td>Sender</td>
<td>Unknown TenantCode in URL</td>
<td>Correct the TenantCode in the URL and resend the message. See section 2.2.</td>
</tr>
<tr>
<td>MHB.MHD.011</td>
<td>Syntax</td>
<td>Sender</td>
<td>Unknown System Function</td>
<td>Check message and contact Market Operator in case of questions. Resend message after correction of message content.</td>
</tr>
</tbody>
</table>
2.2 Exchanging Business Messages

In the following B2B service operations the Business Messages are exchanged:

- SendMessage/ProcessMessage as part of the request;
- PeekMessage as part of the response.

The following URL is used for the posts:

https://<b2b gateway hostname>/soap/<TenantCode>?organisationUser=<organisation username>

The following attributes are provided on the input:

<table>
<thead>
<tr>
<th>A#</th>
<th>Name</th>
<th>Type</th>
<th>Obligation</th>
<th>Value</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Connection: Client Certificate</td>
<td>Certificate</td>
<td>Mandatory</td>
<td>The certificate provided by the client.</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>URL: tenantCode</td>
<td>String</td>
<td>Optional</td>
<td>See chapter 5.2</td>
<td>The Username is used to uniquely identify the OrganisationUser to be used to do the requests.</td>
</tr>
<tr>
<td>3</td>
<td>URL: organisationUser</td>
<td>String</td>
<td>Mandatory</td>
<td>See chapter 5.2</td>
<td>This allows linking multiple OrganisationUsers to one UserIdenity.</td>
</tr>
<tr>
<td>4</td>
<td>&lt;Request Body&gt;</td>
<td>String, UTF-8</td>
<td>Mandatory</td>
<td>The message that should be handled by the Datahub.</td>
<td>Request body as described below.</td>
</tr>
</tbody>
</table>

The above B2B service operation requests or responses will always use the MessageContainer structure to wrap/contain the Business Message.

The MessageContainer structure also contains additional information to identify and interpret the Business Message correctly. See Table below.

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Type</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageContainer</td>
<td>Mandatory</td>
<td>Complex Element</td>
<td>The DocumentReferenceNumber (i.e. UUID) is used to identify the data transfer of a Business Message for Peek and Dequeue of Messages (only available if part of a response message).</td>
</tr>
<tr>
<td>DocumentReferenceNumber</td>
<td>Conditional</td>
<td>xs:string, max=36</td>
<td>Contains the actual Business Message in XML message format. Payload should comply with the market message definition and should include the correct namespace.</td>
</tr>
<tr>
<td>Payload</td>
<td>Mandatory</td>
<td>Complex Element</td>
<td></td>
</tr>
</tbody>
</table>
2.2.1 XML Business Messages in the Payload

XML Business Messages are XML based and always have a single root element. This means that the entire Business Message is placed in the Payload element. If there are multiple transactions in the Business Message they are part of the same payload.

The namespace of the specific XML Business Message must be included in the message. This can be on the Business Message level (also known as locally defined), but also on a higher level (also known as globally defined); as long as the namespace is defined in the message.

This is based on and in line with [SOAP] and [XSD].

2.3 System Users

For the System User a User Identity is set up that is identified by a client certificate (PKI) for its authentication. This User Identity can be linked to one or more Organisation Users within the Datahub. The System Users can only be used for B2B communication through the B2B services.

According to the figure 1 below, the System User (User Identity) is related to an Organisation using the Organisation User entity and System User acts on behalf of that Organisation. The same User Identity can be linked to Organisation Users for different Organisations. An Organisation can as such have multiple System Users.

A System User can only execute transactions for the specific Organisations the System User is related to.

![Figure 1: Relation between User Identity, Organisation User, Organisation and User Roles and System Functions](image-url)
Authorization of a System User is performed at 2 levels based on the User Roles attached to the linked Organisation User:

- B2B Service operation: is the System User authorized to access this B2B service operation?
- Business Messages: is the System User authorized to send in this type of Business Message on behalf of the Organisation (or the delegating Organisation)?

### 2.4 SOAP Fault details

The SOAP fault message will have a limited set of information. The default SOAP fault is returned. The table below shows the elements in a SOAP fault message, where the table indents indicate child elements of a Complex Element. The table below also indicates the origin of the elements, either generic SOAP standard (SOAP) or specific to the CMS product.

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Origin</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault</td>
<td>Mandatory</td>
<td>SOAP</td>
<td>Complex Element. Indicates whether the fault is caused by the Sender or Receiver.</td>
</tr>
<tr>
<td>Code</td>
<td>Mandatory</td>
<td>SOAP</td>
<td>Complex Element. Indicates whether the fault is caused by the Sender or Receiver.</td>
</tr>
<tr>
<td>Value</td>
<td>Mandatory</td>
<td>SOAP</td>
<td>“Sender” or “Receiver” These values contain a prefix to the <a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a> namespace. Sender: The message was incorrectly formed or did not contain the appropriate information in order to succeed. It is generally an indication that the message is not to be resent without change Receiver: The message could not be processed for reasons attributable to the processing of the message rather than to the contents of the message itself. For example, processing could include communicating with an upstream SOAP node, which did not respond. The message could succeed if resent at a later point in time.</td>
</tr>
<tr>
<td>Subcode</td>
<td>Optional</td>
<td>SOAP</td>
<td>Not used</td>
</tr>
<tr>
<td>Reason</td>
<td>Mandatory</td>
<td>SOAP</td>
<td>Complex Element. Provides a human-readable explanation of the fault.</td>
</tr>
<tr>
<td>Text</td>
<td>Mandatory</td>
<td>SOAP</td>
<td>Contains the text of the human-readable explanation of the fault. Must contain an attribute specifying the language with local name “lang” in the namespace <a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a></td>
</tr>
<tr>
<td>Node</td>
<td>Optional</td>
<td>SOAP</td>
<td>Not used</td>
</tr>
<tr>
<td>Role</td>
<td>Optional</td>
<td>SOAP</td>
<td>Not used</td>
</tr>
<tr>
<td>Detail</td>
<td>Mandatory</td>
<td>SOAP</td>
<td>Complex Element. Elements below are specific to CMS product.</td>
</tr>
<tr>
<td>CMSFault</td>
<td>Mandatory</td>
<td>CMS</td>
<td>Complex Element. Root element for the specific CMS product Fault elements.</td>
</tr>
<tr>
<td>ErrorCode</td>
<td>Mandatory</td>
<td>CMS</td>
<td>Error Code (e.g. MHB.MHD.001) identifying specific error situation (See chapter 4.2).</td>
</tr>
<tr>
<td>ErrorIdentification</td>
<td>Optional</td>
<td>CMS</td>
<td>The number as logged in CMS. Based on the Error identifier the full details regarding the error can be obtained at the Market Operator.</td>
</tr>
<tr>
<td>ErrorDetails</td>
<td>Conditional</td>
<td>CMS</td>
<td>Details about the cause of the error (only if XML validation failed due to message syntax errors).</td>
</tr>
</tbody>
</table>

### 2.4.1 Classifying Error codes in this document

In this document the error codes are classified to provide more insight. The classifications are:

- Security error; related to authentication, authorization issues
2.4.2 SOAP Fault example

Below is an example SOAP Fault:

```
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <SOAP-ENV:Fault>
      <SOAP-ENV:Code>
      </SOAP-ENV:Code>
      <SOAP-ENV:Reason>
      </SOAP-ENV:Reason>
      <SOAP-ENV:Detail>
        <urn:CMSFault>
          <urn:ErrorDetails>javax.xml.stream.XMLStreamException: org.xml.sax.SAXParseException; lineNumber: 7; columnNumber: 49; cvc-datatype-valid.1.2.1: '?' is not a valid value for 'dateTime'.</urn:ErrorDetails>
        </urn:CMSFault>
      </SOAP-ENV:Detail>
    </SOAP-ENV:Fault>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

2.5 Timestamps

Datahub logs the timestamp for each incoming operation request and each outgoing operation response. These timestamps are logged in the system and used for reporting. Any timestamps in the message and transactions as provided by the Market Participant are stored for reference but not used for operational management or reporting. All timestamps in B2B communication are presented in UTC and in XML format YYYY-MM-DDTHH:MM:SS+00:00 or YYYY-MM-DDTHH:MM:SS. All timestamps that are stored in the system are stored in UTC.
3 B2B MarketMessaging services

This chapter describes the external interfaces used in a B2B context.

3.1 ProcessMessage operation

A System User uses ‘ProcessMessage’ to send XML Business Messages concerning a market process to the Datahub that needs synchronous processing. The operation implementation directly executes the Market processes and returns the processing result (i.e. synchronously).

3.1.1 Request

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessMessageRequest</td>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageContainer</td>
<td>Mandatory</td>
<td>Complex</td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td>Mandatory</td>
<td>Complex</td>
<td>Contains the actual Business Message in XML message format. Payload should comply with the market message definition and should include the right namespace (see section 2.3).</td>
</tr>
</tbody>
</table>

The following rules are enforced:

- The Payload is syntactically correct (validated against the XSD)
- The System User is authorized to send in the message (Document Type) for the Sender mentioned in the Business Message
- The Message and Transaction ID's used in the message are unique (i.e. not used before).

Failing these rules will result in a corresponding SOAP Fault.

Process validations errors will result in a Response message and not in a SOAP fault.

3.1.2 Response

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProcessMessageResponse</td>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageContainer</td>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DocumentReferenceNumber</td>
<td>Mandatory</td>
<td>xs:string max=36</td>
<td>The DocumentReferenceNumber (i.e. UUID) as generated by the Datahub is used to identify the data transfer of a Business Message for Peek and Dequeue of messages.</td>
</tr>
<tr>
<td>Payload</td>
<td>Mandatory</td>
<td>Complex</td>
<td>Contains the actual Business Message in XML message format. Payload should comply with the market message definition and should include the right namespace (see section 2.3).</td>
</tr>
</tbody>
</table>
3.1.3 Error codes in SOAP Faults

The error codes in the table below are specific to the ProcessMessage B2B service operation, besides the generic error codes mentioned in paragraph 2.2.2.1.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Type</th>
<th>SOAP Code</th>
<th>Meaning</th>
<th>Operational Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHB.MHD.006</td>
<td>Syntax</td>
<td>Sender</td>
<td>The provided Ids are not unique or have been used before</td>
<td>Two possible issues, for both resending is not applicable:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Issue in the client system which results in sending messages with the same id's</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Retry after a MHB.MHD.005 (see operational guidance on that error)</td>
</tr>
</tbody>
</table>

3.2 SendMessage operation

A System User uses ‘SendMessage’ to send XML Business Messages concerning a market process to the Datahub. The operation implementation does not directly execute the Market processes but only receives, checks the syntax, authorisation, uniqueness of ID’s and forwards the Business Message internally. Therefore the response is only a technical acknowledgement to indicate that the request will be processed. The actual processing of the Business Messages is done internally (i.e. asynchronously).

3.2.1 Request

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendMessageRequest</td>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MessageContainer</td>
<td>Mandatory</td>
<td>Complex Element</td>
<td>Contains the actual Business Message in XML message format. Payload should comply with the market message definition and should include the right namespace (see section 2.3).</td>
</tr>
<tr>
<td>Payload</td>
<td>Mandatory</td>
<td>Complex Element</td>
<td></td>
</tr>
</tbody>
</table>

The following rules are enforced:

- The Payload is syntactically correct (validated against the XSD)
- The System User is authorized to send in the message (Document Type) for the Sender mentioned in the Business Message
- The Message and Transaction ID’s used in the message are unique (i.e. not used before).

Failing these rules will result in a corresponding SOAP Fault.

3.2.2 Response
3.2.3 Error codes in SOAP Faults

The error codes in the table below are specific to the SendMessage B2B service operation, besides the generic error codes mentioned in paragraph 2.2.2.1.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Type</th>
<th>SOAP Code</th>
<th>Meaning</th>
<th>Operational Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHB.MHD.006</td>
<td>Syntax</td>
<td>Sender</td>
<td>The provided ids are not unique or have been used before</td>
<td>Two possible issues, for both resending is not applicable: &lt;br&gt; 1. Issue in the client system which results in sending messages with the same id’s &lt;br&gt; 2. Retry after a MHB.MHD.005 (see operational guidance on that error)</td>
</tr>
</tbody>
</table>
A Peek Message is acknowledged by a Dequeue Message containing the DocumentReferenceNumber sent in the header of the Peek Message. As a consequence, a new PeekMessage before dequeue will result in resending the last retrieved message. This allows for a Market Participant to receive the same message for processing in case processing failed on their end. All retrieved messages are registered in the audit log.

### 3.3.3 Error codes in SOAP Faults

There are no additional error codes for this B2B service operation, besides the error codes mentioned in 2.2.2.1.

### 3.3.4 Operational guidance

In case the PeekMessage response does not contain a payload, the Market Participant system should implement a wait cycle before issuing a new PeekMessage request. See section 5.3 for wait time. This also applies where the system returns an internal server error.

### 3.4 Dequeue Message operation

A System User uses ‘DequeueMessage’ to indicate to the Datahub it has handled a message it retrieved from the Datahub using ‘PeekMessage’ and has been able to extract the content in order to use the DocumentReferenceNumber of the business message to sign off. This message will be removed from the logical queue for this Market Participant. Note that when this is done the next message becomes available on the logical queue for retrieving.

It is the responsibility of the Market Participant to regularly peek, process and dequeue messages. The Datahub will continue processing and prepare additional messages independent of the Market Participant retrieving messages. Messages are delivered in the order that the Datahub has created the messages.

#### 3.4.1 Request

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DequeueMessageRequest</td>
<td>Mandatory</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DocumentReferenceNumber</td>
<td>Mandatory</td>
<td>xs:string max=36</td>
<td>UUID</td>
</tr>
</tbody>
</table>

### Constraints

Fingrid Datahub Oy

Katuosoite
Läkärikesätie 21
00620 Helsinki

Puhelin: 030 395 5000
Faksi: 030 395 5196

Y-tunnus 2745543-5, ALV rek. etunimi.sukunimi@fingrid.fi
www.fingrid.fi
### 3.4.2 Response

<table>
<thead>
<tr>
<th>Element</th>
<th>Obligation</th>
<th>Type</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>DequeueMessageResponse</td>
<td>Mandatory</td>
<td></td>
<td>Empty element.</td>
</tr>
</tbody>
</table>

### 3.4.3 Error codes in SOAP Faults

The following error codes are specific to this B2B service operation, besides the error codes mentioned in 2.2.2.1.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Type</th>
<th>SOAP Code</th>
<th>Meaning</th>
<th>Operational Guidance</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHB.MHD.007</td>
<td>System</td>
<td>Sender</td>
<td>Unknown or invalid message reference (e.g. cannot dequeue the current message in the MessageQueue if message reference provided does not match the message reference that has been peeked before (i.e. current message))</td>
<td>Sender error. There are two possible causes: • The wrong DocumentReferenceNumber is taken from the Retrieved message. • The message has already been Dequeued before (parallel peek and dequeue are not possible).</td>
</tr>
</tbody>
</table>

---

**DocumentReferenceNumber**

Must be the DocumentReferenceNumber in the Business Message from the previously peeked message (see PeekMessage).
4 Summary

4.1 Summary of SOAP actions
The following SOAP Actions have to be used in the requests

<table>
<thead>
<tr>
<th>Service operation</th>
<th>SOAP Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SendMessage</td>
<td>sendMessage</td>
</tr>
<tr>
<td>ProcessMessage</td>
<td>processMessage</td>
</tr>
<tr>
<td>PeekMessage</td>
<td>peekMessage</td>
</tr>
<tr>
<td>DequeueMessage</td>
<td>dequeueMessage</td>
</tr>
</tbody>
</table>

4.2 Summary of error codes
This section provides a summary of all error codes that may occur.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>401</td>
<td>Security</td>
<td>Access Denied – User authentication based on the provided certificate has failed. (check if userIdentity is created correctly)</td>
</tr>
<tr>
<td>403</td>
<td>Security</td>
<td>Access Denied – Client certificate is not provided or trusted. (check the certificate in the request)</td>
</tr>
<tr>
<td>404</td>
<td>System</td>
<td>Requested resource not found</td>
</tr>
<tr>
<td>413</td>
<td>System</td>
<td>Content length too large</td>
</tr>
<tr>
<td>500</td>
<td>System</td>
<td>In case of any unidentified errors.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error code</th>
<th>Type</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>MHB.MHD.000</td>
<td>System</td>
<td>General Failure</td>
</tr>
<tr>
<td>MHB.MHD.001</td>
<td>Syntax</td>
<td>Message validation failed</td>
</tr>
<tr>
<td>MHB.MHD.002</td>
<td>System</td>
<td>System configuration error</td>
</tr>
<tr>
<td>MHB.MHD.003</td>
<td>Security</td>
<td>User not authorized for system function (e.g. not found, no rights for the operation or message type, user blocked or inactive)</td>
</tr>
<tr>
<td>MHB.MHD.004</td>
<td>Security</td>
<td>Unknown request</td>
</tr>
<tr>
<td>MHB.MHD.005</td>
<td>System</td>
<td>Back-end timeout</td>
</tr>
<tr>
<td>MHB.MHD.006</td>
<td>Syntax</td>
<td>The provided Ids are not unique or have been used before (MessageID and TransactionID)</td>
</tr>
<tr>
<td>MHB.MHD.007</td>
<td>System</td>
<td>Unknown or invalid message reference (e.g. cannot dequeue the current message in the MessageQueue if message reference provided does not match the message reference that has been peeked before (i.e. current message))</td>
</tr>
<tr>
<td>MHB.MHD.008</td>
<td>Security</td>
<td>Message content unsecure</td>
</tr>
<tr>
<td>MHB.MHD.009</td>
<td>Security</td>
<td>User not authorized for organisation (e.g. System User neither matches PhysicalSender nor (one of) the delegated Organisation(s))</td>
</tr>
<tr>
<td>MHB.MHD.010</td>
<td>Syntax</td>
<td>URL not correct</td>
</tr>
<tr>
<td>MHB.MHD.011</td>
<td>Syntax</td>
<td>Unknown System Function</td>
</tr>
</tbody>
</table>
5  Fingrid Datahub – WSDL details

This section contains Fingrid specific recommendations, file descriptions and examples.

5.1  Parameters

Parameters for Fingrid

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>{TenantCode}</td>
<td>FGR</td>
</tr>
<tr>
<td>{URL}</td>
<td>Will be communicated separately.</td>
</tr>
<tr>
<td>{OrganisationUser}</td>
<td>Specific per Organisation. Defines the registered b2b user in the Datahub created by Fingrid having specific UserRoles for B2B communication.</td>
</tr>
<tr>
<td>Max message size</td>
<td>TBD (see open issue #3)</td>
</tr>
</tbody>
</table>

5.2  Connection details

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Protocol</td>
<td>HTTPS (TLS 1.2)</td>
</tr>
<tr>
<td>Port</td>
<td>443</td>
</tr>
<tr>
<td>SOAP Version</td>
<td>1.2</td>
</tr>
<tr>
<td>Style / use</td>
<td>Document / Literal</td>
</tr>
</tbody>
</table>

5.3  Recommendations


A system not operating according to these recommendations is considered to be using the interface incorrectly.

| #   | Recommendations                                      | Reasoning                                                              |
|-----|------------------------------------------------------|                                                                      |
| 1.  | UTF-8 shall be used for the XML messages             | Preventing encoding issues                                           |
|     | Content-Type: text/xml; charset=UTF-8                |                                                                      |
| 2.  | Prevent pretty print.                               | Pretty printing gives overhead of quite some whitespace.             |
| 3.  | Use of UUID in Message and Transaction ID’s         | Ensuring uniqueness of identifiers                                   |
| 4.  | Prevent sending messages for same accounting point /| Datahub does support processing messages in parallel (Send/Process)  |
|     | agreement/ customer at the same time.               | but can’t not guarantee a sequence in processing when messages are send in parallel |
| 5.  | Don’t use Peek/Dequeue in parallel                  | Datahub does not support parallel Peek messages for a domain. Parallel request will result in getting same message multiple time |
| 6.  | Polling: When Peek message does not return a message sleep for 30 sec seconds before retry. | Prevent too much requests in time window.                            |

5.4  File descriptions

The following table describes the folder structure of the schemas delivery:

<table>
<thead>
<tr>
<th>Folder</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>common</td>
<td>general data elements</td>
</tr>
<tr>
<td>invoicing</td>
<td>message structure for the domain invoicing</td>
</tr>
<tr>
<td>masterdata</td>
<td>message structure for the domain masterdata</td>
</tr>
<tr>
<td>metering</td>
<td>message structure for the domain metering</td>
</tr>
<tr>
<td>query</td>
<td>message structure for the domain query</td>
</tr>
<tr>
<td>wsdl</td>
<td>WSDL for use in communication with Datahub</td>
</tr>
<tr>
<td>test</td>
<td>WSDL for test preparation purposes only. Supports easier setup of test messages in tools like SOAP-UI.</td>
</tr>
</tbody>
</table>
5.5 Fingrid Datahub Examples

The examples are for showing correct structure of the messages. The used values for elements do apply the schema validation rules but are not based on real data.

The examples are based on the [EVENTS] document.

5.5.1 Send Message Request

```xml
  <soap:Header/>
  <soap:Body>
    <urn:SendMessageRequest>
      <urn:MessageContainer>
        <urn:Payload>
          <urn7:InvoicingDataInfoMessage>
            <urn7:InvoicingDataInfo>
              <urn7:Header>
                <urn2:Identification>345345345345345</urn2:Identification>
                <urn2:DocumentType>F13</urn2:DocumentType>
                <urn2:Creation>2018-01-01T10:10:00</urn2:Creation>
                <urn2:PhysicalSenderEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000001</urn2:Identification>
                </urn2:PhysicalSenderEnergyParty>
                <urn2:JuridicalSenderEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000001</urn2:Identification>
                </urn2:JuridicalSenderEnergyParty>
                <urn2:JuridicalRecipientEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000002</urn2:Identification>
                </urn2:JuridicalRecipientEnergyParty>
                <urn2:PhysicalRecipientEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000002</urn2:Identification>
                </urn2:PhysicalRecipientEnergyParty>
                <urn2:SenderRoutingInformation/>
              </urn7:Header>
              <urn7:ProcessEnergyContext>
                <urn3:EnergyBusinessProcessRole>DQ</urn3:EnergyBusinessProcessRole>
                <urn3:EnergyIndustryClassification>23</urn3:EnergyIndustryClassification>
              </urn7:ProcessEnergyContext>
              <urn7:Transaction>
                <urn8:MeteringPoint schemeAgencyIdentifier="9">123456789012345678</urn8:MeteringPoint>
                <urn8:Contract>43456</urn8:Contract>
                <urn8:PartyIdentification schemeAgencyIdentifier="9">456546456</urn8:PartyIdentification>
                <urn8:InvoicingPeriod>
                  <urn8:Start>2018-01-01T10:10:00</urn8:Start>
                  <urn8:End>2018-01-02T10:10:00</urn8:End>
                </urn8:InvoicingPeriod>
                <urn8:InvoicingRow>
                  <urn8:ProductIdentification>345345</urn8:ProductIdentification>
                  <urn8:ProductComponentIdentification>23452345</urn8:ProductComponentIdentification>
                  <urn8:Price>00000009999999.9999999999</urn8:Price>
                  <urn8:PriceUnit>EUR</urn8:PriceUnit>
                  <urn8:PriceUnitCode>EUR</urn8:PriceUnitCode>
                  <urn8:Currency>EUR</urn8:Currency>
                  <urn8:TaxIncluded>1</urn8:TaxIncluded>
                  <urn8:Volume>1.887</urn8:Volume>
                  <urn8:VolumeUnit>KWH</urn8:VolumeUnit>
                  <urn8:VolumeUnitCode>EUR</urn8:VolumeUnitCode>
                  <urn8:Amount>12.99</urn8:Amount>
                  <urn8:Description>This is a test</urn8:Description>
                  <urn8:Tax>0</urn8:Tax>
                  <urn8:Start>2018-01-01T10:10:00</urn8:Start>
                  <urn8:End>2018-01-02T10:10:00</urn8:End>
                </urn8:InvoicingRow>
              </urn7:Transaction>
            </urn7:InvoicingDataInfo>
          </urn7:InvoicingDataInfoMessage>
        </urn:Payload>
      </urn:MessageContainer>
    </urn:SendMessageRequest>
  </soap:Body>
</soap:Envelope>
```
5.5.2 Send message response

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01">
  <soap:Header/>
  <soap:Body>
    <urn:SendMessageResponse>
      <urn:DocumentReferenceNumber>34838cba-b51d-4215-b98f-0af0a34903e6</urn:DocumentReferenceNumber>
    </urn:SendMessageResponse>
  </soap:Body>
</soap:Envelope>
```

5.5.3 Process message request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01"
xmlns:urn2="urn:fi:Datahub:mif:common:HDR_Header:elements:v1"
xmlns:urn29="urn:fi:Datahub:mif:query:F17_PartyInfoRequest:v1"
  <soap:Header/>
  <soap:Body>
    <urn:ProcessMessageRequest>
      <urn:MessageContainer>
        <urn:Payload>
          <urn29:PartyInfoRequestMessage>
            <urn29:PartyInfoRequest>
              <urn29:Header>
                <urn2:Identification>345345345345345</urn2:Identification>
                <urn2:DocumentType>F17</urn2:DocumentType>
                <urn2:Creation>2018-01-01T10:10:00</urn2:Creation>
                <urn2:PhysicalSenderEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000001</urn2:Identification>
                </urn2:PhysicalSenderEnergyParty>
                <urn2:JuridicalSenderEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000001</urn2:Identification>
                </urn2:JuridicalSenderEnergyParty>
                <urn2:JuridicalRecipientEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000002</urn2:Identification>
                </urn2:JuridicalRecipientEnergyParty>
                <urn2:PhysicalRecipientEnergyParty>
                  <urn2:Identification schemeAgencyIdentifier="9">8700000000002</urn2:Identification>
                </urn2:PhysicalRecipientEnergyParty>
                <urn2:SenderRoutingInformation>rtyrtyr</urn2:SenderRoutingInformation>
              </urn2:Header>
              <urn29:ProcessEnergyContext>
                <urn3:EnergyBusinessProcessRole>DSQ</urn3:EnergyBusinessProcessRole>
                <urn3:EnergyIndustryClassification>23</urn3:EnergyIndustryClassification>
              </urn29:ProcessEnergyContext>
            </urn29:Header>
            <urn29:Transaction>
              <urn30:PartyIdentification schemeAgencyIdentifier="9">8700000000001</urn30:PartyIdentification>
            </urn29:Transaction>
          </urn29:PartyInfoRequestMessage>
        </urn:Payload>
      </urn:MessageContainer>
    </urn:ProcessMessageRequest>
  </soap:Body>
</soap:Envelope>
```

5.5.4 Process message response

Example of a response message having an Acknowledgement as answer:
5.5.5 Peek message request

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01"
xmlns:urn2="urn:fi:Datahub:mif:common:HDR_Header:elements:v1"
xmlns:urn4="urn:fi:Datahub:mif:common:ACK_Acknowledgement:elements:v1">
  <soap:Header/>
  <soap:Body>
    <urn:ProcessMessageResponse>
      <urn:MessageContainer>
        <urn:DocumentReferenceNumber>12a10e73-dd65-4787-9ad7-a40a533f225</urn:DocumentReferenceNumber>
        <urn:Payload>
          <urn1:AcknowledgementMessage>
            <urn1:Acknowledgement>
              ...
            </urn1:Acknowledgement>
          </urn1:AcknowledgementMessage>
        </urn:Payload>
      </urn:MessageContainer>
    </urn:ProcessMessageResponse>
  </soap:Body>
</soap:Envelope>
```

5.5.6 Peek message response

5.5.6.1 Peek response with no available message in the message queue

```
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01"
xmlns:urn2="urn:fi:Datahub:mif:common:HDR_Header:elements:v1">
  <env:Header/>
  <env:Body>
    <urn:PeekMessageResponse/>
  </env:Body>
</env:Envelope>
```

5.5.6.2 Peek response with available message in the message queue

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01"
xmlns:urn2="urn:fi:Datahub:mif:common:HDR_Header:elements:v1"
xmlns:urn4="urn:fi:Datahub:mif:masterdata:F01_MasterDataCustomerEvent:elements:v1">
  <soap:Header/>
  <soap:Body>
    <urn:PeekMessageResponse>
      <urn:MessageContainer>
        <urn:DocumentReferenceNumber>?</urn:DocumentReferenceNumber>
        <urn:Payload>
          <urn1:MasterDataCustomerEventMessage>
            <urn1:MasterDataCustomerEvent/>
          </urn1:MasterDataCustomerEventMessage>
        </urn:Payload>
      </urn:MessageContainer>
    </urn:PeekMessageResponse>
  </soap:Body>
</soap:Envelope>
```
5.5.7 Dequeue message request

```xml
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01">
  <soap:Header/>
  <soap:Body>
    <urn:DequeueMessageRequest>
      <urn:DocumentReferenceNumber>8f3d3929-2dcd-475a-8824-5a566f128bd7</urn:DocumentReferenceNumber>
    </urn:DequeueMessageRequest>
  </soap:Body>
</soap:Envelope>
```

5.5.8 Dequeue message response

```xml
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" xmlns:urn="urn:cms:b2b:v01">
  <soap:Header/>
  <soap:Body>
    <urn:DequeueMessageResponse/>
  </soap:Body>
</soap:Envelope>
```