# Table of contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table of contents ...........................................................................</td>
<td>2</td>
</tr>
<tr>
<td>1 Document management .....................................................................</td>
<td>4</td>
</tr>
<tr>
<td>1.1 Document history .......................................................................</td>
<td>4</td>
</tr>
<tr>
<td>2 Introduction ..................................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Goal of the service .....................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.2 New in version 3.0 .....................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.3 Goal of the document ...................................................................</td>
<td>5</td>
</tr>
<tr>
<td>2.4 eHealth platform document references ......................................</td>
<td>6</td>
</tr>
<tr>
<td>2.5 External document references ..................................................</td>
<td>6</td>
</tr>
<tr>
<td>2.6 Service history ..........................................................................</td>
<td>6</td>
</tr>
<tr>
<td>3 Business and privacy requirements ...............................................</td>
<td>7</td>
</tr>
<tr>
<td>3.1 For issues in production ..........................................................</td>
<td>7</td>
</tr>
<tr>
<td>3.2 For issues in acceptance ..........................................................</td>
<td>7</td>
</tr>
<tr>
<td>3.3 For business issues ....................................................................</td>
<td>7</td>
</tr>
<tr>
<td>3.4 Certificates ................................................................................</td>
<td>7</td>
</tr>
<tr>
<td>4 Global overview .............................................................................</td>
<td>8</td>
</tr>
<tr>
<td>5 Step-by-step ..................................................................................</td>
<td>9</td>
</tr>
<tr>
<td>5.1 Technical requirements ................................................................</td>
<td>9</td>
</tr>
<tr>
<td>5.1.1 Use of the eHealth SSO solution ..........................................</td>
<td>9</td>
</tr>
<tr>
<td>5.1.2 End-to-End Encryption for known recipient .............................</td>
<td>9</td>
</tr>
<tr>
<td>5.2 Process overview .......................................................................</td>
<td>9</td>
</tr>
<tr>
<td>5.2.1 Web Service - Web Service Definition Language (WSDL) .............</td>
<td>9</td>
</tr>
<tr>
<td>5.3 eHealthBox Consultation WS ......................................................</td>
<td>11</td>
</tr>
<tr>
<td>5.3.1 Lifetime of a message ...........................................................</td>
<td>11</td>
</tr>
<tr>
<td>5.3.2 Out-of-Office system ..............................................................</td>
<td>11</td>
</tr>
<tr>
<td>5.3.3 getBoxInfo Method ..................................................................</td>
<td>11</td>
</tr>
<tr>
<td>5.3.4 getMessagesList Method ..........................................................</td>
<td>14</td>
</tr>
<tr>
<td>5.3.5 getMessagesList Request ........................................................</td>
<td>14</td>
</tr>
<tr>
<td>5.3.6 GetAllEhboxesMessagesList Method .........................................</td>
<td>19</td>
</tr>
<tr>
<td>5.3.7 getFullMessage Method ...........................................................</td>
<td>23</td>
</tr>
<tr>
<td>5.3.8 MoveMessage Method ................................................................</td>
<td>26</td>
</tr>
<tr>
<td>5.3.9 DeleteMessage Method .............................................................</td>
<td>29</td>
</tr>
<tr>
<td>5.3.10 getHistory Method ..................................................................</td>
<td>32</td>
</tr>
<tr>
<td>5.3.11 GetMessageAcknowledgmentsStatus Method ..................................</td>
<td>34</td>
</tr>
<tr>
<td>5.3.12 InsertOoO Method ..................................................................</td>
<td>37</td>
</tr>
<tr>
<td>5.3.13 DeleteOoO Method ..................................................................</td>
<td>40</td>
</tr>
<tr>
<td>5.3.14 GetOoOList Method ..................................................................</td>
<td>42</td>
</tr>
<tr>
<td>5.3.15 Used Types ..........................................................................</td>
<td>44</td>
</tr>
<tr>
<td>6 Risks and security ..........................................................................</td>
<td>58</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>6.1</td>
<td>SOAP with attachments</td>
</tr>
<tr>
<td>6.2</td>
<td>Security</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Business security</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Web Service Security Policy</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Security policies to apply</td>
</tr>
<tr>
<td>7</td>
<td>Test and release procedure</td>
</tr>
<tr>
<td>7.1</td>
<td>Procedure</td>
</tr>
<tr>
<td>7.1.1</td>
<td>Initiation</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Development and test procedure</td>
</tr>
<tr>
<td>7.1.3</td>
<td>Release procedure</td>
</tr>
<tr>
<td>7.1.4</td>
<td>Operational follow-up</td>
</tr>
<tr>
<td>7.2</td>
<td>Test cases</td>
</tr>
<tr>
<td>8</td>
<td>Error and failure messages</td>
</tr>
<tr>
<td>8.1</td>
<td>Error Element Status Codes</td>
</tr>
<tr>
<td>8.2</td>
<td>Consultation Response Status Codes</td>
</tr>
<tr>
<td>8.3</td>
<td>Soap Fault Error Codes</td>
</tr>
<tr>
<td>8.3.1</td>
<td>Schema Validation Errors</td>
</tr>
<tr>
<td>8.3.2</td>
<td>Technical Errors</td>
</tr>
</tbody>
</table>
1 Document management

1.1 Document history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Description of changes / remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
<td>06/05/2013</td>
<td>eHealth platform</td>
<td>eHealthBox v3.0 WS</td>
</tr>
<tr>
<td>3.1</td>
<td>05/02/2018</td>
<td>eHealth platform</td>
<td>New logo – modified links</td>
</tr>
<tr>
<td>3.2</td>
<td>30/04/2018</td>
<td>eHealth platform</td>
<td>Extract Annex</td>
</tr>
</tbody>
</table>
2 Introduction

2.1 Goal of the service

The eHealthBox Consultation WS allows an authenticated user to consult information about the content associated with his eHealthBox.

A user can get general information on his eHealthBox, a list of messages for a specific folder and the content of a specific message. He can also move a message to his inbox and handle his Out-of-Offices (OoO’s).

Fields indicated as ‘obsolete’ are old fields, still in use by some systems and kept for backward compatibility. They are out-of-date and must not be used by new partners for they do not provide any ‘extra’ feature.

The size of a message is currently limited to 10MB. The global size of an eHealthBox is also limited to 10mb (including inbox and bin folder). Note that an encrypted message weighs more due to the encryption overhead.

2.2 New in version 3.0

- A new “Out-of-Office” system was implemented. It will enable the sender to know if one of the recipients is absent and to send his message to a substitute or substitutes in order to be treated. E.g.: physicians on holiday may want to ensure continuity of healthcare services for their patients. To do so, they can automatically transfer their messages to another colleague responsible during their holidays thanks to the “Out-of-Office” system.
- Three new methods were added: InsertOoO used to add an OoO, DeleteOoO used to delete an inserted OoO, GetOoOList used to get the list of inserted OoO.
- eHealthBoxes are now limited in size on inbox and trash bin folder. A new method DeleteMessage was added in order to be able to delete message from the trash bin folder or directly from the inbox.
- SOAP with Attachments (SwA) standard is now being used in place of MTOM.
- News and Document types were merged: News is now a special case of a document.
- ContentSpecification contains a ContentType element now, specifying the type of message. 2 new message types (error, acknowledgment) were added to the 2 existing (document, news).

2.3 Goal of the document

This document provides functional and technical information about calling the eHealthBox Consultation WS, as provided by the eHealth platform.

In this service specification document, we will explain the structure and content aspects of the possible requests, as well as the replies of the eHealth WS. An example illustrates each of those messages. A list of possible errors can also be found in this document.

This information should allow (the IT department of) an organization to integrate and use the WS call.

Some technical and legal requirements must be met in order for the eHealth platform WS to be integrated in client applications; this document is meant to provide you with an overview of these requirements.

This document is neither a development nor a programming guide for internal applications; the partners of the eHealth platform always have a total freedom within those fields. Nevertheless, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, the partners must commit to comply with specifications, data format, and release processes described in this document.

In addition, our partners in the health sector must also comply with the business rules of validation and integration of data within their own applications in order to minimize errors and incidents.
2.4 eHealth platform document references

All the document references can be found on the portal of the eHealth platform¹. These versions or any following versions can be used for the eHealth platform service.

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Version</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Glossary.pdf</td>
<td>1.0</td>
<td>01/01/2010</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>2</td>
<td>Secure Token Service (STS)</td>
<td>1.0</td>
<td>31/08/2010</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>3</td>
<td>End-to-End Encryption: bekende bestemming/Destinataire connu</td>
<td>2.3</td>
<td>06/05/2011</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>4</td>
<td>eHealthBox Publication</td>
<td>3.3</td>
<td>28/02/2018</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>5</td>
<td>eBox_Quality</td>
<td>1.01</td>
<td>23/04/2018</td>
<td>eHealth platform</td>
</tr>
<tr>
<td>6</td>
<td>eHBox_SSO</td>
<td>1.01</td>
<td>23/04/2018</td>
<td>eHealth platform</td>
</tr>
</tbody>
</table>

2.5 External document references

All documents can be found through the internet. They are available to the public, but not supported by eHealth.

<table>
<thead>
<tr>
<th>ID</th>
<th>Title</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><a href="http://www.w3.org/TR/SOAP-attachments">http://www.w3.org/TR/SOAP-attachments</a></td>
<td>11/12/2000</td>
</tr>
<tr>
<td>2</td>
<td><a href="http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html">http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html</a></td>
<td>20/04/2006</td>
</tr>
<tr>
<td>3</td>
<td><a href="http://ws-i.org/profiles/basic/1.1/swaref.xsd">http://ws-i.org/profiles/basic/1.1/swaref.xsd</a></td>
<td>NA</td>
</tr>
<tr>
<td>4</td>
<td><a href="http://docs.oasis-open.org/ws-sx/ws-securypolicy/200702/">http://docs.oasis-open.org/ws-sx/ws-securypolicy/200702/</a></td>
<td>NA</td>
</tr>
</tbody>
</table>

2.6 Service history

This chapter contains the list of changes applied to the service with respect to the previous version.

<table>
<thead>
<tr>
<th>Previous version</th>
<th>Previous release date</th>
<th>changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>29/04/2012</td>
<td>Major changes: SOAP with Attachment, OoO system, DeleteMessage, MoveMessage, structured notification messages, performances.</td>
</tr>
<tr>
<td>1.0</td>
<td>03/02/2011</td>
<td>DEPRECATED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Major changes: Encryption, Multi-box, Publication Web Application, Consultation Web Application reviewed, general reliability ...</td>
</tr>
</tbody>
</table>

¹ https://ehealth.fgov.be/ehealthplatform
3 Business and privacy requirements

3.1 For issues in production

eHealth platform contact center:

- Phone: 02/788 51 55
- Mail: support@ehealth.fgov.be
- Contact Form :
  - [https://www.ehealth.fgov.be/ehealthplatform/nl/contact](https://www.ehealth.fgov.be/ehealthplatform/nl/contact) (Dutch)
  - [https://www.ehealth.fgov.be/ehealthplatform/fr/contact](https://www.ehealth.fgov.be/ehealthplatform/fr/contact) (French)

3.2 For issues in acceptance

Integration-support@ehealth.fgov.be

3.3 For business issues

- regarding an existing project: the project manager in charge of the application or service
- regarding a new project and other business issues: info@ehealth.fgov.be

3.4 Certificates

- In order to access the secured eHealth platform environment you have to obtain an eHealth platform certificate, used to identify the initiator of the request. In case you do not have one, please consult:
  - Dutch version: [https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten](https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten)
- For technical issues regarding eHealth platform certificates
  - Acceptance: acceptance-certificates@ehealth.fgov.be
  - Production: support@ehealth.fgov.be
4 Global overview

This global overview aims to show the different steps needed to use the Consultation WS.

Step 1. The first required step for the user to use the Consultation WS is to contact the STS Service to acquire his proper token containing his data. (See 5.1.1)

Step 2. Secondly, thanks to his token, the user can use the different “Consultation WS” to retrieve his message(s), manage his eHealthBox or get information on it.

Step 3. Depending on the request of the user, the “Consultation WS” will provide the user with an answer concerning his eHealthBox or one of his eHealthBoxes.

Step 4. Finally, if the message was encrypted for the recipient, the client deciphers his message with his private key and the Crypto Library (please consult the “ETEE for known recipient” Cookbook on the portal of the eHealth platform).

In order to publish any message or reply, the user should use the eHealthBox publication process.
5 Step-by-step

5.1 Technical requirements

All the xml requests submitted to the WS must be encoded in the UTF-8 format.

5.1.1 Use of the eHealth SSO solution

This section specifies how the call to the Secure Token Service (STS) must be done in order to access the WS. You must precise several attributes in the request. The details on the identification attributes and the certification attributes can be found in the separate document eHealth eHBox_SSO.

To access the eHealth web service, the response token must contain “true” for the ‘boolean’ certification attribute.

If you obtain “false”, contact the eHealth contact center to verify that the requested test cases were correctly configured.

5.1.2 End-to-End Encryption for known recipient

If an encrypted message was received, it has to be deciphered first. See Chapter 4 – Global Overview.

In order to decipher the content of a message and the various fields, you have to use the local stored private key and the Crypto Libraries. Each field must be deciphered separately (one at a time).

For more information about the use of the encryption libraries: please consult the CookBook ETEE for known recipients (https://www.ehealth.fgov.be/ehealthplatform/etee-know recipient)

Encrypted message convention: If an encrypted message is received, ALL “Encryptable” fields contain (one and all) encrypted content. You can detect if the message has been encrypted with the element IsEncrypted in the responses from GetMessagesList, GetAllEhboxesMessagesList and GetFullMessage.

5.2 Process overview

5.2.1 Web Service - Web Service Definition Language (WSDL)

On website of the eHealth platform, you will find all the technical information.
The important sections of the WSDL of the Consultation WS are:

- The applicable Policies, which cover the security aspects.
- The different methods: getBoxInfo, getMessagesList, GetAllEhboxesMessagesList, getFullMessage, moveMessage, getMessageHistory, getMessageAcknowledgmentsStatus, InsertOoO, DeleteOoO, GetOoOList.
- The types that are used by the methods.
- The fault messages are also defined for each method.

5.3 eHealthBox Consultation WS

5.3.1 Lifetime of a message

- When a message reaches its expiration date, when it has been read AND already placed in the recycle bin; it is definitely removed from the application.
- When a message is older than 1 year (counted from publication date), it is definitely removed from the application, even if it has not been read.

5.3.2 Out-of-Office system

5.3.2.1 Introduction

This system enables the sender to know if one of the recipients is absent and to send his message to a substitute or substitutes, so it can be treated. For example, physicians on holiday may want to ensure continuity of healthcare services for their patients. To do so, they can automatically transfer their messages to another colleague responsible during their holidays thanks to the “Out-of-Office” system.

5.3.2.2 Out-of-Office restrictions

- A person cannot be defined as a substitute if he is already absent for this period (or part of the period).
- It is however possible to introduce an OoO\(^2\) while being a substitute for someone else.
- A person may be substitute for several other persons.
- The absent person and his substitutes are persons, not organizations.
- A person is identified by his ID and his quality.
- There may be maximum five substitutes per OoO.
- A maximum of TEN periods OoO may exist per eHealthBox.
- OoO periods may not overlap.
- A period may not end later than J + 1 Year.
- A period can last up to 1 Year.
- The end date is mandatory.

A WS request contains one period and a maximum of five substitutes may be specified for that period. The OoO activates automatically when the time is reached.

5.3.3 getBoxInfo Method

The getBoxInfo method allows an authenticated user to receive general information about his mailbox: the current used size of your mailbox, the maximum allowed size of your mailbox, and the number of messages, which could not be received because the mailbox was full. These messages are still waiting to be put in your

---

\(^2\) OoO : Out of Office
mailbox. You need to clean your mailbox until the current size is lower than the max size. The messages will then enter into your mailbox.

**5.3.3.1 getBoxInfo Request**

You can optionally request information of another of your mailboxes by specifying it via BoxId,

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use one of his other eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>

**5.3.3.2 getBoxInfo Response**

The response contains a success status code and general information on the eHealthBox as explained below. The BoxId element enables you to discover information on the current eHealthBox if you currently do not have any. Attention should be paid to NbrMessagesInStandBy larger than zero and if CurrentSize > MaxSize.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (Id) attributed to the exchange request/response by the eHealth platform is used to identify the session.</td>
</tr>
</tbody>
</table>
| Status     | The Status block contains a code and a message. If no error has occurred during transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. Otherwise: In case of a business error:  
  - The Code is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).  
  - The Message will be a description of the error. Each Message has a Lang (language) characteristic:  
    - “FR”: French  
    - “NL”: Dutch  
    - “EN”: English  
    - “DE”: German  
    - “NA”: Not applicable  
  In case of technical errors, you will receive a Soap Fault message (see Chapter 8). |
| BoxId      | The identification number (Id) and the Type of the eHealthBox are provided. If applicable, the Subtype is also returned.  
  - The Quality of the owner from the eHealthBox (see Error! Reference source not found.). |
| NbrMessagesInStandBy | The number of messages in standby because the eHealthBox is full. To consult these messages, the user has to delete some of the other. |
| CurrentSize | The current size of the eHealthBox expressed in bytes. |
| MaxSize    | The maximum size of the eHealthBox expressed in bytes. |

### 5.3.3.3 Example

The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetBoxInfoRequest/>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/
  S:Body>
```
  <Status>
    <Code>100</Code>
    <Message Lang="EN">SUCCESS</Message>
  </Status>
  <BoxId>
    <Id>9999999964</Id>
    <Type>INSS</Type>
    <Quality>DOCTOR</Quality>
  </BoxId>
  <NbrMessagesInStandBy>0</NbrMessagesInStandBy>
  <CurrentSize>58</CurrentSize>
  <MaxSize>10485760</MaxSize>
</ns3:GetBoxInfoResponse>
</S:Body>
</S:Envelope>

5.3.4 getMessagesList Method

The getMessagesList method provides a list of messages for a specific folder of your eHealthBox listed in order by date (most recent first, index “1”). A consequence could be that if a new message arrives between two consecutive queries, a message will be shown two times (message “1” becomes “2”, “2” becomes “3”, etc.). E.g. if you requested the messages between “1” and “100” and then the messages between “101” and “200”, then the message “100” would be the same as message “101”.

The messages contents are not yet returned by this method but it returns a list with all the information needed to treat, filter, sorts the messages. The sender, recipient, title message, publication date, message size, custom metas are all displayed for example.

5.3.5 getMessagesList Request

You can optionally request information of another of your mailboxes by specifying it via BoxId, This method can only return 100 messages at a time; consequently, you must use it multiple times if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>Use another of my eBoxes</td>
</tr>
<tr>
<td>Source</td>
<td>The folder to consult</td>
</tr>
<tr>
<td>StartIndex</td>
<td>Index of the first message</td>
</tr>
<tr>
<td>EndIndex</td>
<td>Index of the last message</td>
</tr>
</tbody>
</table>
BoxId | If the client wants to use one of his other eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.

Source | You can mention the folder specific to your request via the Source parameter. The possible values are:
- “INBOX” for the inbox folder.
- “SENTBOX” for the sent box folder.
- “BININBOX” for messages moved from the inbox folder.
- “BINSENTBOX” for messages moved from the sent box folder.

StartIndex | Index of the first message (minimum 1).

EndIndex | Index of the last message (minimum 1). A maximum of 100 messages can be returned at once. EndIndex < StartIndex + 100

5.3.5.1 getMessagesList Response

The response contains a success status code and as many Message elements as there are messages in the considered eHealthBox. Each element contains all necessary information to treat the message without downloading each individual message.
**Field name** | **Description**  
--- | ---  
**Id** | The ticket number (Id), attributed to the exchange request/response by the eHealth platform and used to identify the eHealth platform session.  
**Status** | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be '100' and the Message 'SUCCESS'. Otherwise: In case of a business error:  
- The Code is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).
The Message will be a description of the error. Each Message has a Lang (language) characteristic:
- "FR": French
- "NL": Dutch
- "EN": English
- "DE": German
- "NA": Not applicable

In case of technical errors, you will receive a Soap Fault message (see Chapter 8).

### Source

You can specify the folder specific to your request via the Source parameter. The possible values are:
- "INBOX" for the inbox folder.
- "SENTBOX" for the sent box folder.
- "BININBOX" for messages moved from the inbox folder.
- "BINSENTBOX" for messages moved from the sent box folder.

### Message

0-to-more Message tag(s) describe(s) the eHealthBox Message(s). Each of them is defined by the following:
- The MessageId that represents a unique message identification generated by the system and returned during publication and when calling the getMessagesList. String of 13 digits.
- The Destination of the Message (see section 5.3.15.16)
- The Sender of the Message (see section Error! Reference source not found.)
- The MessageInfo that contains additional information about the Message such as publication date, size ... (see section 5.3.15.14).
- The ContentInfo of the Message such as title, mime type ... (see section 5.3.15.6)
- The ContentSpecification of the Message that contains information such as importance ... (see section 5.3.15.7).
- The CustomMeta of the Message that contains free Meta data specified by the user (see section 5.3.15.8)

### 5.3.5.2 Example

The following example does not contain the SAML assertion.

Request:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetMessagesListRequest>
      <Source>INBOX</Source>
      <StartIndex>1</StartIndex>
      <EndIndex>100</EndIndex>
    </urn:GetMessagesListRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
Response:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetMessagesListResponse
     xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
     xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <Source>INBOX</Source>
      <Message>
        <MessageId>9Y0002LKM100K</MessageId>
        <Destination>
          <Id>77012824158</Id>
          <Type>INSS</Type>
          <Quality>DOCTOR</Quality>
        </Destination>
        <Sender>
          <Id>71000139</Id>
          <Type>NIHII</Type>
          <Quality>HOSPITAL</Quality>
          <Name>Doe</Name>
          <FirstName>John</FirstName>
        </Sender>
        <MessageInfo>
          <PublicationDate>2011-06-28+02:00</PublicationDate>
          <ExpirationDate>2011-12-31+01:00</ExpirationDate>
          <Size>46</Size>
        </MessageInfo>
        <ContentInfo>
          <EncryptableINSSPatient>OTgwNTMwNDU3NDYyMQ0K</EncryptableINSSPatient>
          <ContentType>NEWS</ContentType>
          <Title>News in eHealthBox</Title>
          <MimeType>text/plain</MimeType>
          <HasFreeInformations>true</HasFreeInformations>
          <HasAnnex>false</HasAnnex>
        </ContentInfo>
        <ContentSpecification>
          <IsImportant>false</IsImportant>
          <IsEncrypted>false</IsEncrypted>
        </ContentSpecification>
      </Message>
    </ns4:GetMessagesListResponse>
  </S:Body>
</S:Envelope>
```
5.3.6  GetAllEhboxesMessagesList Method

The `GetAllEhboxesMessagesList` method provides a list of all the messages from all the eHealthBoxes of a user (personal and enterprise eHealthBoxes) for a specific folder. This method repeatedly calls upon the GetMessagesList method for all known eHealthBoxes of the currently connected user.

5.3.6.1  GetAllEhboxesMessagesList Request

This method can only return 100 messages at a time; consequently, you must call upon it multiple times if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>You can specify the folder specific to your request via the Source parameter.</td>
</tr>
<tr>
<td></td>
<td>The possible values are:</td>
</tr>
<tr>
<td></td>
<td>• “INBOX” for the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “SENTBOX” for the sent box folder.</td>
</tr>
<tr>
<td></td>
<td>• “BININBOX” for messages moved from the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “BINSENTBOX” for messages moved from the sent box folder.</td>
</tr>
<tr>
<td>StartIndex</td>
<td>Index of the first message (minimum 1).</td>
</tr>
<tr>
<td>EndIndex</td>
<td>Index of the last message (minimum 1). A maximum of 100 messages can be</td>
</tr>
<tr>
<td></td>
<td>returned at a time. EndIndex &lt; StartIndex + 100</td>
</tr>
</tbody>
</table>

5.3.6.2  GetAllEhboxesMessagesList Response

The response contains a success status code and as many `Message` elements as there are messages in all the eHealthBoxes of the currently connected user (personal and enterprise eHealthBoxes). Each element contains all necessary information to treat the message without downloading each individual message. You can identify which eHealthBox received the message via the `Destination` element.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Id</strong></td>
<td>The ticket number (<em>Id</em>), attributed to the exchange request/response by the eHealth platform and used to identify the eHealth platform session.</td>
</tr>
</tbody>
</table>
| **Status** | The *Status* block contains a code and a message. If no error has occurred during the transaction, the *Code* will be ‘100’ and the *Message* ‘SUCCESS’. Otherwise: In case of a business error:  
  - The *Code* is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).  
  - The *Message* will be a description of the error. Each Message has a *Lang* (language) characteristic: 
    - “FR”: French  
    - “NL”: Dutch |
<table>
<thead>
<tr>
<th>Source</th>
<th>You can specify the folder concerned by your request via the <em>Source</em> parameter. The possible values are:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• “INBOX” for the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “SENTBOX” for the sent box folder.</td>
</tr>
<tr>
<td></td>
<td>• “BININBOX” for messages moved from the inbox folder.</td>
</tr>
<tr>
<td></td>
<td>• “BINSENTBOX” for messages moved from the sent box folder.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Message</th>
<th>0-to-more Message tag(s) describe(s) the eHealthBox Message(s). Each of them is defined with the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• The MessageId that represents a unique message identification generated by the system and returned during the publication and when calling upon the getMessagesList.</td>
</tr>
<tr>
<td></td>
<td>• The <em>Destination</em> of the Message (see section 5.3.15.16).</td>
</tr>
<tr>
<td></td>
<td>• The <em>Sender</em> of the Message (see section Error! Reference source not found.).</td>
</tr>
<tr>
<td></td>
<td>• The <em>MessageInfo</em> that contains additional information about the Message such as publication date, size ... (see section 5.3.15.14).</td>
</tr>
<tr>
<td></td>
<td>• The <em>ContentInfo</em> of the Message such as title, mime type ... (see section 5.3.15.6).</td>
</tr>
<tr>
<td></td>
<td>• The <em>ContentSpecification</em> of the Message that contains information such as importance ... (see section 5.3.15.7).</td>
</tr>
<tr>
<td></td>
<td>• The <em>CustomMeta</em> of the Message that contains free Meta data specified by the user (see section 5.3.15.8).</td>
</tr>
</tbody>
</table>

### 5.3.6.3 Example

The following example does not contain the SAML assertion.

**Request:**

```xml
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetAllEhboxesMessagesListRequest>
      <Source>INBOX</Source>
      <StartIndex>1</StartIndex>
      <EndIndex>100</EndIndex>
    </urn:GetAllEhboxesMessagesListRequest>
  </soapenv:Body>
</soapenv:Envelope>
```
Response:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
    <S:Body>
        <ns4:GetAllEhboxesMessagesListResponse
            xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
            xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
            <Status>
                <Code>100</Code>
                <Message Lang="EN">SUCCESS</Message>
            </Status>
            <Source>INBOX</Source>
            <Message>
                <MessageId>9Y0002LVHU003</MessageId>
                <Destination>
                    <Id>99999999964</Id>
                    <Type>INSS</Type>
                    <Quality>DOCTOR</Quality>
                </Destination>
                <Sender>
                    <Id>99999999964</Id>
                    <Type>INSS</Type>
                    <Quality>DOCTOR</Quality>
                    <Name>Efstathios</Name>
                    <FirstName>Frens</FirstName>
                </Sender>
                <MessageInfo>
                    <PublicationDate>2011-10-12+02:00</PublicationDate>
                    <ExpirationDate>2011-10-17+02:00</ExpirationDate>
                    <Size>0</Size>
                </MessageInfo>
                <ContentInfo>
                    <EncryptableINSSPatient/>
                    <ContentType>NEWS</ContentType>
                    <Title>test title</Title>
                    <MimeType>application/octet-stream</MimeType>
                    <HasFreeInformations>false</HasFreeInformations>
                    <HasAnnex>false</HasAnnex>
                </ContentInfo>
                <ContentSpecification>
                    <IsImportant>false</IsImportant>
                    <IsEncrypted>false</IsEncrypted>
                </ContentSpecification>
            </Message>
        </ns4:GetAllEhboxesMessagesListResponse>
    </S:Body>
</S:Envelope>
```
5.3.7 **getFullMessage Method**

The *getFullMessage* method is used to get the corresponding complete message and its content to a provided *MessageId*.

### 5.3.7.1 **getFullMessage Request**

You can optionally request a message from another of your mailboxes by specifying it via *BoxId*.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BoxId</strong></td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
| **Source** | You can specify the folder specific to your request via the *Source* parameter. The possible values are:  
- “INBOX” for the inbox folder.  
- “SENTBOX” for the sent box folder. |
| **MessageId** | The *MessageId* is a unique message identification generated by the system and returned during the publication and when calling upon the *getMessagesList*. String of 13 digits. |

#### 5.3.7.2 **getFullMessage Response**

The response contains the same information as already returned by GetMessagesList plus the Message content in the element *Message*.
### Field name | Description
---|---
Id | The ticket number (Id) attributed to the exchange request/response by the eHealth platform is used to identify the eHealth platform session.

Status | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. Otherwise:
- The Code is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).
- The Message will be a description of the error. Each Message has a Lang (language) characteristic:
  - “FR”: French
  - “NL”: Dutch
  - “EN”: English
  - “DE”: German
  - “NA”: Not applicable

In case of technical errors, you will receive a Soap Fault message (see Chapter 8).

Sender | The Sender of the Message (see section Error! Reference source not found.)
Message | The Message itself (see section 5.3.15.13)
MessageInfo | Additional information about the Message (see section 5.3.15.14)

### 5.3.7.3 Example
The following example does not contain the SAML assertion.
Request:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetFullMessageRequest>
      <Source>INBOX</Source>
      <MessageId>9Y0002LKLP004</MessageId>
    </urn:GetFullMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Response:

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetFullMessageResponse
xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1
xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <Sender>
        <Id>71000139</Id>
        <Type>NIHII</Type>
        <Quality>HOSPITAL</Quality>
        <Name>Balduino</Name>
        <FirstName>Anil</FirstName>
      </Sender>
      <Message MessageId="9Y0002LKLP004">
        <PublicationId>InitialDoc</PublicationId>
        <DestinationContext>
          <Id>9999999964</Id>
          <Type>INSS</Type>
          <Quality>DOCTOR</Quality>
        </DestinationContext>
        <ContentContext>
          <Content>
            <Document>
              <Title>Document in eHealthBox</Title>
              <EncryptableBinaryContent/>
              <DownloadFileName>test.txt</DownloadFileName>
              <MimeType>text/plain</MimeType>
            </Document>
            <FreeInformations>
              <EncryptableFreeText/>
            </FreeInformations>
          </Content>
        </ContentContext>
      </Message>
    </ns4:GetFullMessageResponse>
  </S:Body>
</S:Envelope>
```
5.3.8 MoveMessage Method

The MoveMessage method enables the user to move a message from a Source ("INBOX", "SENTBOX", "BININBOX", "BINSENTBOX") to a Destination ("INBOX", "SENTBOX", "BININBOX", "BINSENTBOX"). Only some combinations are allowed as explained below. You need to indicate if a message was received or sent by the concerned eHealthBox. You can do this by looking where the message is situated or compare the currently connected user and the Sender and/or Destination element.

Allowed combinations:

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>INBOX</td>
<td>BININBOX</td>
</tr>
<tr>
<td>SENT BOX</td>
<td>BINSENTBOX</td>
</tr>
<tr>
<td>BININBOX</td>
<td>INBOX</td>
</tr>
<tr>
<td>BINSENTBOX</td>
<td>SENTBOX</td>
</tr>
</tbody>
</table>

Limitation:
Max 100 messages/requests

5.3.8.1 MoveMessage Request

In Source specify where the message is currently situated and in Destination where the message must be moved to. In MessageId specify as many elements as there are messages to be moved. This method can only move 100 messages at a time; consequently, you must use it multiple times if necessary.
### Field name | Description
--- | ---
BoxId | If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.

Source | You can specify the folder specific to your request via the Source parameter. The possible values are:
- “INBOX” for the inbox folder.
- “SENTBOX” for the sent box folder.
- “BININBOX” for messages moved from the inbox folder.
- “BINSENTBOX” for messages moved from the sent box folder.

Destination | You can specify the folder specific to your request via the Destination parameter. The possible values are:
- “INBOX” for the inbox folder.
- “SENTBOX” for the sent box folder.
- “BININBOX” for messages moved from the inbox folder.
- “BINSENTBOX” for messages moved from the sent box folder.

MessageId | The MessageId’s corresponding to the message(s) to move.

### 5.3.8.2 MoveMessage Response

The response contains a success status code or a Business Error as defined in Chapter 8. The Business Error enables you to identify which messages where not successfully moved, even though all other have been moved successfully.
### Field name | Description
---|---
**Id** | The ticket number \((Id)\) attributed to the exchange request/response by the eHealth platform is used to identify the session.

**Status** | The *Status* block contains a code and a message. If no error has occurred during the transaction, the *Code* will be ‘100’ and the *Message* ‘SUCCESS’. Otherwise:
- In case of a business error:
  - The *Code* is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).
  - The *Message* will be a description of the error. Each Message has a *Lang* (language) characteristic:
    - “FR”: French
    - “NL”: Dutch
    - “EN”: English
    - “DE”: German
    - “NA”: Not applicable
  In case of technical errors, you will receive a Soap Fault message (see Chapter 8).

**MessageId** | List of MessageId, which could not be moved. The other messages however were successfully moved.

#### 5.3.8.3 Example
The following example does not contain the SAML assertion.

**Request:**

```xml
  <soapenv:Header/>
  <soapenv:Body>
```


<urn:MoveMessageRequest>
  <Source>INBOX</Source>
  <Destination>BIN</Destination>
  <MessageId>9Y0002LKM3006</MessageId>
  <MessageId>9Y0002LKM3007</MessageId>
</urn:MoveMessageRequest>

Response:

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:MoveMessageResponse
      xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
      xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>813</Code>
        <Message Lang="EN">Not all messages were moved successfully. Please verify for each message that the Source and the MessageID are correct. Also pay attention that a message in the recycle bin which was moved from the Inbox cannot be restored back to the Sentbox and vice versa.</Message>
      </Status>
      <MessageId>9Y0002LKM3006</MessageId>
    </ns4:MoveMessageResponse>
  </S:Body>
</S:Envelope>

5.3.9 DeleteMessage Method

The DeleteMessage method enables the user to delete physically and definitely one or more messages from the inbox, sent box or bin. This can be used to clean up the eHealthBox when reaching the size limit. Be cautious when using this method. Best is to show a warning message to the user before deleting the messages.

Limitation:
Max 100 messages/requests

5.3.9.1 DeleteMessage Request

In MessageId you specify where the messages to delete from the bin or directly from the inbox or sent box. This method can only delete 100 messages at a time; consequently, you must use it multiple times if necessary.
Field name | Description
--- | ---
BoxId | If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.
Source | You can specify the folder specific to your request via the Source parameter. The possible values are:
   - “INBOX” for the inbox folder.
   - “SENTBOX” for the sent box folder.
   - “BININBOX” for messages moved from the inbox folder.
   - “BINSENTBOX” for messages moved from the sent box folder.
MessageId | The MessageId’s corresponding to the message(s) to delete.

### 5.3.9.2 DeleteMessage Response

The response contains a success status code or a business error as defined in Chapter 8. The business error enables you to identify which messages where not successfully deleted, even though all other have been deleted successfully.
### Field name | Description
--- | ---
Id | The ticket number (Id) attributed to the exchange request/response by the eHealth platform is used to identify the eHealth session.
Status | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. Otherwise:

- In case of a business error:
  - The Code is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).
  - The Message will be a description of the error. Each Message has a Lang (language) characteristic:
    - “FR”: French
    - “NL”: Dutch
    - “EN”: English
    - “DE”: German
    - “NA”: Not applicable
- In case of technical errors, you will receive a Soap Fault message (see Chapter 8).
MessageId | List of MessageId which could not be deleted. All other messages were successfully deleted.

#### 5.3.9.3 Example

The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:DeleteMessageRequest>
      <MessageId>INBOX</MessageId>
      <MessageId>9Y0002LKM3006</MessageId>
      <MessageId>9Y0002LKM3007</MessageId>
    </urn:DeleteMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:DeleteMessageResponse
xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>815</Code>
    </ns4:DeleteMessageResponse>
  </S:Body>
</S:Envelope>
```
<Message Lang="EN">One or more messages couldn’t be deleted. All other messages were successfully deleted. Please verify for each message that the MessageId is correct, and it is in the Inbox or in the recycle bin.</Message>
</Status>
<MessageId>9Y0002LKM3006</MessageId>
</ns4:DeleteMessageResponse>
</S:Body>
</S:Envelope>

5.3.10 getHistory Method

When a new message is sent and updates an old news item (by using the same PublicationId), the old news item is archived and replaced by the newer one. This method enables you to request a list of the old versions of that news item by using the MessageId attributed to the newer news item.

The getHistory method is used to get the older message versions of a “news” type message. The method returns a list of MessageId’s corresponding to the previous version of a “news item”, which enables the user to enter a getFullMessage on those MessageId’s.

The getHistory method cannot retrieve a history of a document.

5.3.10.1 getHistory Request

You can request the list of MessageId of a news item from your Inbox or from your Sent Box. You can optionally request information of another of your mailboxes by specifying it via BoxId. In MessageId, you can place the MessageId of the newer news item, or the MessageId of an old version of the same news item.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
| Source       | You can specify the folder specific to your request via the Source parameter. The possible values are:  
- “INBOX” for the inbox folder.  
- “SENTBOX” for the sent box folder. |
| MessageId    | The MessageId of the message to consult. The MessageId is a unique message identification generated by the system and returned during the publication and when calling upon getMessagesList. String of 13 digits. |

5.3.10.2 getHistory Response

The response gives you a group of MessageId’s which concern the same news item. You can then enter a GetFullMessage in order to retrieve the old news item if necessary.
<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (id) attributed to the exchange request/response by the eHealth platform is used to identify the eHealth session.</td>
</tr>
</tbody>
</table>
| Status      | The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. In case of a business error:  
  - The Code is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).  
  - The Message will be a description of the error. Each Message has a Lang (language) characteristic:  
    - "FR": French  
    - "NL": Dutch  
    - "EN": English  
    - "DE": German  
    - "NA": Not applicable  
In case of technical errors, you will receive a Soap Fault message (see Chapter 8). |
| MessageId   | List of MessageId’s from the older message versions.                                                                                         |

**5.3.10.3 Example**

The following example does not contain the SAML assertion.

**Request:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
 xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
```

---

![eHealthBox Consultation Web Service Cookbookv.3.2 - 30/04/2018](image)
Response:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns4:GetHistoryResponse
    xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
    xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <MessageId>9Y0002LKL004</MessageId>
    </ns4:GetHistoryResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

### 5.3.11 GetMessageAcknowledgmentsStatus Method

The `GetMessageAcknowledgmentsStatus` method is used to find out which message the user has sent, which recipients have received, viewed or read the message and at what time.

#### 5.3.11.1 GetMessageAcknowledgmentsStatus Request

This method can only return 100 acknowledgements at a time; consequently, you must call upon it multiple times if necessary.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
<tr>
<td><strong>Field name</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>MessageId</td>
<td>The <code>MessageId</code>'s of the message to consult. The <code>MessageId</code> is a unique message identification generated by the system and returned during the publication and when calling upon the <code>getMessagesList</code>. String of 13 digits.</td>
</tr>
<tr>
<td>StartIndex</td>
<td>Index of the first acknowledgment (minimum 1)</td>
</tr>
<tr>
<td>EndIndex</td>
<td>Index of the last acknowledgment (minimum 1). A maximum of 100 acknowledgments can be returned at once. <code>EndIndex &lt; StartIndex + 100</code></td>
</tr>
</tbody>
</table>

### 5.3.11.2 `GetMessageAcknowledgmentsStatus` Response

The response gives you information about your sent message: who received and read your message and at what time.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The ticket number (id) attributed to the exchange request/response by the eHealth platform and used to identify the eHealth session.</td>
</tr>
</tbody>
</table>
### Status

The **Status** block contains a code and a message. If no error has occurred during the transaction, the **Code** will be ‘100’ and the **Message** ‘SUCCESS’. Otherwise:

- In case of a business error:
  - The **Code** is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).
  - The **Message** will be a description of the error. Each Message has a **Lang** (language) characteristic:
    - “FR”: French
    - “NL”: Dutch
    - “EN”: English
    - “DE”: German
    - “NA”: Not applicable
- In case of technical errors, you will receive a Soap Fault message (see Chapter 8).

### AcknowledgmentsStatus

Contains a **Row** for each different **Recipient** of the message. Each **Row** contains the identification of the **Recipient** (**EhboxIdentifier**, see section 5.3.15.16), the time the message was published, the time the message was received (=viewed) by that **Recipient**, and the time the message was read by that **Recipient**.

#### 5.3.11.3 Example

The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetMessageAcknowledgmentsStatusRequest>
      <MessageId>9Y0002LKH020J</MessageId>
      <StartIndex>1</StartIndex>
      <EndIndex>100</EndIndex>
    </urn:GetMessageAcknowledgmentsStatusRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetMessageAcknowledgmentsStatusResponse
xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <AcknowledgmentsStatus>
        <Row>
```
5.3.12 InsertOoO Method

The InsertOoO method is used to insert an OoO for the considered eHealthBox. In order to insert an OoO for another eHealthBox, BoxId can be used. There are a number of rules and limitations to respect as described in section 5.3.2.2. If these rules are not respected a business error is returned as detailed in section 8.2.

5.3.12.1 InsertOoO Request

Only one OoO can be inserted by request, but multiple substitutes may be specified.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>
**StartDate**
The *StartDate* of the OoO. This date is inclusive. Ex. 2013-07-03+02:00

**EndDate**
The *EndDate* of the OoO. This date is inclusive. Ex. 2013-07-17+02:00

**Substitute**
Optional. You may want to specify someone, which is replacing you during your absence. The structure is as usual, detailed in 5.3.15.3

### 5.3.12.2 InsertOoO Response

The response is “Success” without further details if everything went right. Otherwise the wrong or absent substitutes will be returned in the response as detailed below.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>The Id element contains the OoOId attributed to this OoO by the system, which can later be used in DeleteOoO for example.</td>
</tr>
<tr>
<td>Status</td>
<td>The Status block contains a code and a message. If no error has occurred during the transaction, the <em>Code</em> will be ‘100’ and the <em>Message</em> ‘SUCCESS’. Otherwise:</td>
</tr>
<tr>
<td></td>
<td>In case of a business error:</td>
</tr>
<tr>
<td></td>
<td>- The <em>Code</em> is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).</td>
</tr>
<tr>
<td></td>
<td>- The <em>Message</em> will be a description of the error. Each Message has a Lang (language) characteristic:</td>
</tr>
</tbody>
</table>
In case of technical errors, you will receive a Soap Fault message (see Chapter 8).

| Substitute | Contains a Row for each different Substitute which is incorrect, unknown, or absent (absence period of this substitute is then specified). |

### 5.3.12.3 Example

The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:InsertOoORequest>
      <BoxId>
        <Id>982374923273</Id>
        <Type>NHII</Type>
        <Quality>DOCTOR</Quality>
      </BoxId>
      <StartDate>2013-07-03+02:00</StartDate>
      <EndDate>2013-07-17+02:00</EndDate>
      <Substitute>
        <Id>82351425106</Id>
        <Type>INSS</Type>
        <Quality>DOCTOR</Quality>
      </Substitute>
    </urn:InsertOoORequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  S:Body>
  <ns4:InsertOoOResponse
    xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
    xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
    <Status>
      <Code>824</Code>
      <Message Lang="EN">One or more substitutes cannot be chosen because they are absent.</Message>
    </Status>
    <Substitute>
```
5.3.13 DeleteOoO Method

The *DeleteOoO* method is used to delete an OoO for the considered eHealthBox. In order to delete an OoO for another eHealthBox, BoxId can be used. The OoO Id must be specified. It can be retrieved from InsertOoO, in the Id element or from GetOoOList.

### 5.3.13.1 DeleteOoO Request

Only one OoO can be inserted by request, but multiple substitutes may be specified.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
<tr>
<td>OoOId</td>
<td>The OoOId from the OoOs that the user wants to delete. The OoOId’s can be retrieved from InsertOoO, in the Id element or from GetOoOList.</td>
</tr>
</tbody>
</table>

### 5.3.13.2 DeleteOoO Response

The response is “Success” without further details if everything went right. Otherwise, the OoOId which could not be found, will be returned in the response as detailed below. All other OoO that were found were deleted.
### Field name | Description
---|---
Id | The ticket number (Id) attributed to the exchange request/response by the eHealth platform, is used to identify the eHealth session.

### Status
The Status block contains a code and a message. If no error has occurred during the transaction, the Code will be ‘100’ and the Message ‘SUCCESS’. Otherwise:

- In case of a business error:
  - The Code is an error code that unequivocally identifies the problem (see Chapter 7 for the possible values).
  - The Message will be a description of the error. Each Message has a Lang (language) characteristic:
    - “FR”: French
    - “NL”: Dutch
    - “EN”: English
    - “DE”: German
    - “NA”: Not applicable

- In case of technical errors, you will receive a Soap Fault message (see Chapter 8).

### OoOId
Contains the OoOId’s which could not be found and were not deleted. All others were deleted.

### 5.3.13.3 Example
The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/
xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:DeleteOoORequest>
      <BoxId>
        <Id>9999999964</Id>
        <Type>INSS</Type>
        <Quality>DOCTOR</Quality>
      </BoxId>
      <OoOId>1005</OoOId>
    </urn:DeleteOoORequest>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:DeleteOoOResponse
xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
```
<Code>840</Code>
<Message Lang="EN">One or more OoOId are invalid.</Message>
</Status>
</OoOId>
</ns4:DeleteOoOResponse>
</S:Body>
</S:Envelope>

5.3.14 GetOoOList Method

The GetOoOList method is used to list all inserted OoO for the considered eHealthBox. The list contains all OoO: past, active and future OoO. The OoOId returned can further be used to delete an OoO.

5.3.14.1 GetOoOList Request

Only one OoO can be inserted by request, but multiple substitutes may be specified.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoxId</td>
<td>If the client wants to use another of his eHealthBoxes, he can specify it here (see section 5.3.15.3). This avoids the client having to re-authenticate himself each time.</td>
</tr>
</tbody>
</table>

5.3.14.2 GetOoOList Response

The response contains a success status code and as many OoO elements as there are OoO in the considered eHealthBox. Each element contains all OoO data when it was inserted.
Field name | Description
---|---
OoOId | The OoOId assigned by the system to the OoO.
StartDate | The *StartDate* of the OoO. This date is inclusive. Ex. 2013-07-03+02:00
EndDate | The *EndDate* of the OoO. This date is inclusive. Ex. 2013-07-17+02:00
Substitue | Optional. The person who is replacing the absent person. The structure is as usual, detailed in 5.3.15.3

### 5.3.14.3 Example

The following example does not contain the SAML assertion.

**Request:**

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
   xmlns:urn="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
  <soapenv:Header/>
  <soapenv:Body>
    <urn:GetOoOListRequest/>
  </soapenv:Body>
</soapenv:Envelope>
```

**Response:**

```xml
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetOoOListResponse
       xmlns:ns2="urn:be:fgov:ehealth:errors:service:v1"
       xmlns:ns4="urn:be:fgov:ehealth:ehbox:consultation:protocol:v3">
      <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
      </Status>
      <OoO>
        <OoOId>1000</OoOId>
        <StartDate>2013-07-03+02:00</StartDate>
        <EndDate>2013-07-17+02:00</EndDate>
        <Substitue>
          <Id>82351425106</Id>
          <Type>INSS</Type>
          <Quality>DOCTOR</Quality>
        </Substitue>
      </OoO>
    </ns4:GetOoOListResponse>
  </S:Body>
</S:Envelope>
```
### 5.3.15 Used Types

#### 5.3.15.1 Acknowledgment

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageId</td>
<td>The MessageId of the original message.</td>
</tr>
<tr>
<td>Recipient</td>
<td>The recipient receiving or reading the original message. This element is only present for an acknowledgment of type ‘RECEIVED’ or ‘READ’.</td>
</tr>
</tbody>
</table>
| AckType    | 3 acknowledgment types exist: ‘PUBLISHED’, ‘RECEIVED’ and ‘READ’:
  - An ack. ‘published’ => when the system is treating the original message
  - An ack. ‘Received’ => when the recipient has seen the original message (request getMessageList)
  - An ack. ‘Read’ => when the recipient has opened the original message (request getFullMessage) |
| DateTime   | Date & Time when the acknowledgment was issued. E.g. <DateTime>2001-12-17T09:30:47Z</DateTime> |
5.3.15.2 Annex

An Annex has an EncryptableTitle, a human readable description of its content (minimum 1, maximum 400 characters in binary form). If IsEncrypted is true (see Section 5.3.15.7), the title must be encrypted (see section 5.1.2).

EncryptableBinaryContent: Contains a SOAP reference to the attachment E.g. `<EncryptableBinaryContent>cid:4906633928</EncryptableBinaryContent>`. Type is xsd:anyURI. If IsEncrypted is true (see Section 5.3.15.7), the attachment must be encrypted (see section 5.1.2).

EncryptableTextContent: Text content, in binary form. If IsEncrypted is true (see Section 5.3.15.7), the text content must be encrypted (see section 5.1.2).

DownloadFileName: E.g. “principal.pdf” (string minimum 1, maximum 255).

MimeType: Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 255).

5.3.15.3 BoxId

A BoxId contains all the information on the eHealthBox the client wants to use for the request. BoxId cannot be used when authenticated as an organization.

You will find all mandatory information about the allowed combinations Id-Type-Quality (See eHealthBox – Supported qualities v1.1).
### Field name | Descriptions
---|---
ID | Your eHealthBox's identification number. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.
Type | Your eHealthBox’s ID type (“INSS”, “NIHII”, “FAMPH” or “CBE”). String.
Subtype (obsolete) | If the recipient is an organization, the Subtype allows (if necessary) further specification (such as "HOSPITAL" SubType for a Hospital Quality, or "GROUP" SubType for a Group Quality). String.
Quality | Your eHealthBox’s Quality. String (See eHBox_Quality)

### 5.3.15.4 Content

A Content contains the message content (a document or a news item) and optionally zero-or-more free information, a Patient INSS and zero-or-more annexes.

### Field name | Descriptions
---|---
Document | A news item is now a special case of a Document. See section 5.3.15.10
FreelInformations | See section 5.3.15.12
EncryptableINSSPatient | This optional field allows specifying an INSS number of a patient concerned by the message content. If IsEncrypted is true (see Section 5.3.15.7), the INSS number must be encrypted (see section 5.1.2).

Annex | See section 5.3.15.2

Acknowledgment | See section 5.3.15.1

Error | See section 5.3.15.11

### 5.3.15.5 ContentContext

A **ContentContext** contains the message content and message details, as well as zero-or-more (50 maximum) free **CustomMetas**. The user for internal usage can freely specify these CustomMetas. You can define a Key and a value for each **CustomMeta** (see 5.3.15.8).

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>See section 5.3.15.4</td>
</tr>
<tr>
<td>ContentSpecification</td>
<td>See section 5.3.15.7</td>
</tr>
<tr>
<td>CustomMeta</td>
<td>See section 5.3.15.8</td>
</tr>
</tbody>
</table>

### 5.3.15.6 ContentInfo

**ContentInfo**

`ehboxc:ContentInfoType`
### Field name | Descriptions
--- | ---
EncryptableINSSPatient | This optional field allows specifying an INSS number of a patient concerned by the message content. If `IsEncrypted` is true (see Section 5.3.15.7), the content must be encrypted before being converted to `xs:base64Binary` (see section 5.1.2).
Title | The `Title` of the message, a human readable description of its purpose (string minimum 1, maximum 400).
MimeType | Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 255).
HasFreeInformation | A flag (true or false) that indicates if the message has any *Free Information*.
HasAnnex | A flag (true or false) that indicates if the message has an *Annex*.

#### 5.3.15.7 ContentSpecification

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationName</td>
<td>The Application sending the message (optional, string minimum 1, maximum 25).</td>
</tr>
<tr>
<td>ContentType</td>
<td>The <code>ContentType</code> of the message (“DOCUMENT”, “NEWS”, “ACKNOWLEDGMENT”, “ERROR”).</td>
</tr>
<tr>
<td>IsImportant</td>
<td>Boolean (true or false) that indicates if the message is to be considered as important.</td>
</tr>
<tr>
<td>IsEncrypted</td>
<td>Boolean (true or false) that indicates if the content has been encrypted.</td>
</tr>
</tbody>
</table>

#### 5.3.15.8 CustomMeta

*CustomMeta* was introduced in order to enable the client to transport any Meta information relative to the message he wants. You can specify a maximum of 50 different pairs (key, value). The fields are limited each to 250 characters. Those *CustomMetas* will be transported from the sender to the recipient. You can for example add a CustomMeta for internal usage as “CategoryId, 17”, or “MessageContent, Blood analysis”.

---

*eHealthBox Consultation Web Service Cookbook v.3.2 - 30/04/2018*
### 5.3.15.9 DestinationContext

A *DestinationContext* contains all the information on the recipient. You will find all mandatory information about the allowed combinations Id-Type- Quality (See eHealthBox – Supported qualities v1.1).
<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The recipient’s identification number. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.</td>
</tr>
<tr>
<td>Type</td>
<td>The recipient’s ID type (“INSS”, “NIHII”, “FAMPH” or “CBE”). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the recipient is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or “GROUP” SubType for a Group Quality). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A Quality defines the recipient’s eHealthBox. String (see Error! Reference source not found.)</td>
</tr>
<tr>
<td>User</td>
<td>An optional User (FirstName and LastName) can be added in the destination context. In case of a publication to an organization, this field is used to specify a member of this organization (e.g. a doctor working in a hospital), (string minimum 1, maximum 100). String.</td>
</tr>
<tr>
<td>Mandate (obsolete)</td>
<td>Optional authority information will be added if the recipient has been granted an authority. The constituent’s identification number (Id) and Type are requested. If the constituent is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or &quot;GROUP&quot; SubType for a Group Quality). The recipient’s name may be specified.</td>
</tr>
</tbody>
</table>

### 5.3.15.10 Document

Please note that a message will contain either a news item or a document, not both.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>A Document has a Title, a human readable description of its intent (string minimum 1, maximum 400).</td>
</tr>
</tbody>
</table>
EncryptableBinaryContent
Contains a SOAP reference to the attachment E.g. <EncryptableBinaryContent>cid:490663928</EncryptableBinaryContent>. Type is xsd:anyURI.
If IsEncrypted is true (see Section 5.3.15.7), the attachment must be encrypted (see section 5.1.2).

EncryptableTextContent
Text content, in binary form.
If IsEncrypted is true (see Section 5.3.15.7), the text content must be encrypted (see section 5.1.2).

DownloadFileName
E.g. “principal.pdf” (string minimum 1, maximum 255).

MimeType
Represents the mime type of the content. E.g. “application/pdf”, “text/plain”, “application/octet-stream” (string minimum 1, maximum 255).

5.3.15.11 Error

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PublicationId</td>
<td>The PublicationId of the sent message (in error).</td>
</tr>
<tr>
<td>Code</td>
<td>The Error Code as defined in table 8.1. The number of possible error codes can evolve.</td>
</tr>
<tr>
<td>Message</td>
<td>A human readable English message.</td>
</tr>
<tr>
<td>Destination</td>
<td>The recipient causing an error e.g. because he could not be found. Check if the identifier and identifier type is correct.</td>
</tr>
<tr>
<td>Failure</td>
<td>All-purpose field containing technical Ids as PublicationId or MessageId. Not currently used, but for future needs.</td>
</tr>
</tbody>
</table>
5.3.15.12 FreeInformations

The sender is free to add more information via the FreeInformations field, which will be transparently provided to the recipient(s).

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>EncryptableFreeText</td>
<td>Text content, in binary form. If IsEncrypted is true (see section 5.3.15.7), the text must be encrypted (see section 5.1.2).</td>
</tr>
<tr>
<td>Table</td>
<td>Title (the title of the table) and 1 or more Row(s) (each Row has a EncryptableLeftCell and a EncryptableRightCell as string). If IsEncrypted is true (see section 5.3.15.7), the content must be encrypted (see section 5.1.2).</td>
</tr>
<tr>
<td>EncryptableOldFreeInformation (obsolete)</td>
<td>Text content, in binary form and a Render attribute (as a string). If IsEncrypted is true (see section 5.3.15.7), the content must be encrypted (see section 5.1.2).</td>
</tr>
</tbody>
</table>
### 5.3.15.13 Message

- **MessageId**: The `MessageId` is a unique message identification generated by the system and returned during the publication and when calling upon the `getMessagesList`. String of 13 digits.

- **PublicationId**: The Id that the sender has used to publish the message. String, minimum 1, maximum 13.

- **DestinationContext**: The `DestinationContext` is a complex type that contains information about the recipients. See details in section 5.3.15.9. A `Message` can have numerous `DestinationContext` (numerous recipients).

- **ContentContext**: The `ContentContext` is a complex type that contains the message content. See details in section 5.3.15.5.

- **Meta**: Currently, no meta information is defined. Additional system meta information can be defined by the eHealth platform and used in convention with the client (for future needs). The type of meta information must be defined in the eHealthBox system before it can be used (see section 5.3.15.15).

<table>
<thead>
<tr>
<th>Field name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageId</td>
<td>The <code>MessageId</code> is a unique message identification generated by the system and returned during the publication and when calling upon the <code>getMessagesList</code>. String of 13 digits.</td>
</tr>
<tr>
<td>PublicationId</td>
<td>The Id that the sender has used to publish the message. String, minimum 1, maximum 13.</td>
</tr>
<tr>
<td>DestinationContext</td>
<td>The <code>DestinationContext</code> is a complex type that contains information about the recipients. See details in section 5.3.15.9. A <code>Message</code> can have numerous <code>DestinationContext</code> (numerous recipients).</td>
</tr>
<tr>
<td>ContentContext</td>
<td>The <code>ContentContext</code> is a complex type that contains the message content. See details in section 5.3.15.5.</td>
</tr>
<tr>
<td>Meta</td>
<td>Currently, no meta information is defined. Additional system meta information can be defined by the eHealth platform and used in convention with the client (for future needs). The type of meta information must be defined in the eHealthBox system before it can be used (see section 5.3.15.15).</td>
</tr>
</tbody>
</table>
### 5.3.15.14 MessageInfo

A *MessageInfo* contains short details about the message.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PublicationDate</td>
<td>The <em>Date</em> on which the message has been published (format: yyyy-mm-dd+hh:mm).</td>
</tr>
<tr>
<td>ExpirationDate</td>
<td>The <em>Expiration Date</em> of the message (format: yyyy-mm-dd+hh:mm). Please find more information about the Expiration Date in Lifetime of a message.</td>
</tr>
<tr>
<td>Size</td>
<td>The <em>Size</em> of the message in bytes.</td>
</tr>
</tbody>
</table>

### 5.3.15.15 Meta

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>The type of the meta information (string minimum 1, maximum 250)</td>
</tr>
<tr>
<td>Value</td>
<td>A list of <em>Values</em> for this <em>Type</em> (string minimum 1, maximum 250)</td>
</tr>
</tbody>
</table>

### 5.3.15.16 Recipient

A *Recipient* contains all the information relative to your eHealthBox that received the message. This is very useful when you call upon the *GetAllEhboxesMessagesList* to identify which message was received by which...

---

CopyMailTo

One or more email address(es) that will receive a notification when the message has been published (optional, string minimum 1, maximum 80). If you would like to notify more than one recipient, you can add each e-mail address in a separate *CopyMailTo* element.

By default, a notification will be sent to the hospital’s security manager (registered in the user management of the social security) or in case of a publication to an individual person (doctor, citizen...): the person will receive a notification if he has updated his email address on the web application eHealth Update Info (https://www.ehealth.fgov.be/fr/esante/professionnels-de-la-sante/uppad).

---

---

---

---
eHealthBox. After all, two different eHealthBoxes could receive the same message. The message would then appear two times in `GetAllEhboxesMessagesList`.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The recipient’s identification number. This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.</td>
</tr>
<tr>
<td>Type</td>
<td>The recipient’s ID type (&quot;INSS&quot;, &quot;NIHII&quot;, &quot;FAMPH&quot; or &quot;CBE&quot;). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the recipient is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or &quot;GROUP&quot; SubType for a Group Quality). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A Quality defines the recipient’s eHealthBox. String (see Error! Reference source not found.).</td>
</tr>
<tr>
<td>User</td>
<td>An optional User (FirstName and LastName) can be added in the destination context. In case of a publication to an organization, this field is used to specify a member of this organization (e.g. a doctor working in a hospital), (string minimum 1, maximum 100).</td>
</tr>
</tbody>
</table>
5.3.15.17 Sender

A Sender element contains all the information relative to the Sender of the message.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>This is a digital number representing an INSS, NIHII, FAMPH, or CBE. This number is in a String format.</td>
</tr>
<tr>
<td>Type</td>
<td>The Sender ID type (&quot;INSS&quot;, &quot;NIHII&quot;, &quot;FAMPH&quot; or &quot;CBE&quot;). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the sender is an organization, the Subtype allows (if necessary) further specification (such as &quot;HOSPITAL&quot; SubType for a Hospital Quality, or &quot;GROUP&quot; SubType for a Group Quality). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A Quality defines the Sender eHealthBox. String (see Error! Reference source not found.).</td>
</tr>
<tr>
<td>Name</td>
<td>Name of the Sender. String.</td>
</tr>
<tr>
<td>FirstName</td>
<td>FirstName of the Sender (optional). String.</td>
</tr>
<tr>
<td>PersonInOrganisation</td>
<td>If the message was sent by an organization like a hospital, the organization name will be stored in &quot;Name&quot; above and the INSS of the person really sending the message will be stored in &quot;PersonInOrganisation&quot; for information. String.</td>
</tr>
</tbody>
</table>
5.3.15.18  Substitute

A *Substitute* element contains all the information relative to the *Substitute* designated for a person.

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>This is a digital number representing an INSS, NIHII, FAMPH, or CBE. String.</td>
</tr>
<tr>
<td>Type</td>
<td>The Sender ID type (&quot;INSS&quot;, &quot;NIHII&quot;, &quot;FAMPH&quot; or &quot;CBE&quot;). String.</td>
</tr>
<tr>
<td>Subtype (obsolete)</td>
<td>If the sender is an organization, the <em>Subtype</em> allows (if necessary) further specification (such as &quot;HOSPITAL&quot; <em>SubType</em> for a Hospital <em>Quality</em>, or &quot;GROUP&quot; <em>SubType</em> for a Group <em>Quality</em>). String.</td>
</tr>
<tr>
<td>Quality</td>
<td>A <em>Quality</em> defines the Sender <em>eHealthBox</em>. String (see <a href="#">Error! Reference source not found.</a>).</td>
</tr>
<tr>
<td>AbsentFrom</td>
<td>The <em>AbsentFrom</em> of the OoO. This date is inclusive. Ex. 2013-07-03+02:00</td>
</tr>
<tr>
<td>AbsentTo</td>
<td>The <em>AbsentTo</em> of the OoO. This date is inclusive. Ex. 2013-07-17+02:00</td>
</tr>
</tbody>
</table>
6  Risks and security

6.1  SOAP with attachments

The standard is used to transfer binary data via SOAP web services. SOAP with attachments is an older alternative to MTOM. SwA is used in combination with WS-Security.

The SwA specification describes how to send binary data separated from the SOAP body using multipart MIME messages.

SOAP Messages with attachments is a link to the W3C specification of the standard:
http://www.w3.org/TR/soap-attachments

Attachments Profile Version 1.0 is a link to the WS-I profile with clarifications and amendments to the SwA specification:
http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html

To keep a reference to the attachment from within the message payload, WS-I defined the simpleType swaRef:
http://ws-i.org/profiles/basic/1.1/swaRef.xsd

The attachment data is no part of the message payload. Therefore, when you sign the SOAP body for secure transfer, the attachment data itself is not signed.

6.2  Security

6.2.1  Business security

In case the development adds an additional use case based on an existing integration, the eHealth platform (info@eHealth.fgov.be) must be informed at least one month in advance with a detailed estimate of the expected load. This will ensure an effective capacity management.

In case of technical issues on the WS, the partner may obtain support from the contact center (see section Error! Reference source not found. that is responsible for this service.

In case the eHealth platform finds a bug or vulnerability in its software, we advise the partner to update his application with the newest version of the software within 10 business days.

In case the partner finds a bug or vulnerability in the software or WS that the eHealth platform has delivered, he is obliged to contact and inform the eHealth platform immediately. He is prohibited to publish this bug or vulnerability in any case.

6.2.2  Web Service Security Policy

Web service security used in this manner is in accordance with the common standards. Your call will provide:

- that the request is authenticated with the SAML security profile policy.

See the internet link http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/ for the specifications.

See also Error! Reference source not found. and Error! Reference source not found. for a more detailed description of the SSO Access in the case of the eHealth platform.

- SSL one way.

- an X.509 certificate. This certificate will contain the identifiers of the caller: INSS or NIHII number or CBE enterprise number. For more information on how to obtain a certificate please consult the chapter about the eHealth Certificates in the section “Services de base” (French) – “Basisdiensten” (Dutch) on the portal of the eHealth platform (https://www.ehealth.fgov.be/ehealthplatform)

- the time-to-live of the message: one minute.

- the signature of the timestamp, body and binary security token. This will allow the eHealth platform to verify the integrity of the message and the identity of the message author.
In order to use the WS, an authorization from the eHealth platform is required.

6.2.3 Security policies to apply

We expect that you use SSL one way for the transport layer. As a WS security policy, we expect:

- a timestamp (the date of the request), with a Time to live of one minute. (If the message does not arrive during this minute, it shall not be treated).
- the signature with the certificate of
  - the timestamp (the one mentioned above)
  - the body (the message itself)
  - and the binary security token: an eHealth certificate or a SAML token issued by STS

This will allow the eHealth platform to verify the integrity of the message and the identity of the message author.

For more information about this security policy: please consult

https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten (Dutch version)

https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth (French version)
7 Test and release procedure

7.1 Procedure

This chapter explains the procedures for testing and releasing an application in acceptance or production.

7.1.1 Initiation

If you intend to use the eHealth service, please contact info@ehealth.fgov.be. The Project department will provide you with the necessary information and mandatory documents.

7.1.2 Development and test procedure

You have to develop a client (service consumer) in order to connect to our WS. Most of the required information about the integration is published on the eHealth portal site. (https://ehealth.fgov.be/ehealthplatform)

In some cases, the eHealth platform provides you with a mock-up service or test cases in order for you to test your client (service consumer) before releasing it in the acceptance environment.

7.1.2.1 Create test cases

Rules to access the Publication services are the same in test and in production.

Access rules:
- to use the Consultation WS, the user must be part of one of the following profiles: hospital, nurse, group, institution, doctor, and laboratory ...
- authentication with a certificate

The development team of the eHealth platform configures all test cases.

7.1.2.2 Request a certificate

Prior to requesting the certificate, you need to have installed the latest version of Java 1.6 and the Belgium eID middleware. You also need a smart-card reader and a Belgian eID.

Before requesting an eHealth certificate: please consult the chapter about the eHealth Certificates.

https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten (Dutch version)

https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth (French version)

Depending on the user, you will need NIHII, INSS or CBE identification numbers in order to request the certificate.

7.1.2.3 Obtain SAML token

The usage of the STS (STS) and the structure of the exchanged xml-messages are described in the eHealth STS cookbook.

In the case of eHealthBox Publication WS, see Error! Reference source not found..

7.1.3 Release procedure

When the development tests are successful, you can request to access the eHealth acceptance environment.

From the moment you start the integration and acceptance tests, the eHealth platform suggests testing during at least one month.

After the acceptance tests have been successfully completed, the partner sends his test results and performance results with a sample of the “eHealth request” and “eHealth answer” by email to his point of contact at the eHealth platform.

Then the eHealth platform and the partner agree on a release date. The eHealth platform prepares the connection to the production environment and provides the partner with the necessary information. During the release day, the partner provides the eHealth platform with feedback on the test and on the performance tests.
7.1.4 Operational follow-up

Once in production, the partner using the eHealth platform service for one of its applications will always perform test first in the acceptance environment before releasing any adaptations of his application in production. In addition, he will inform the eHealth platform on the progress and test period.

7.2 Test cases

This section describes a systematic process to test the Consultation WS.

The eHealth platform recommends performing tests for all of the following cases:

1. Consult your eHealthbox information with the method “getBoxInfo()”.
2. Based on your tests cases defined previously for the Publication service, get the list of messages contained in your eHealthbox with the method “getMessagesList()”. Execute this request on your inbox, sent box and on your bin.
3. Pick one of the “MessageIds” returned from your messages list and use it to get the full message with the method “getFullMessage()”. Execute this request on a message from your inbox and sent box.
4. Use again your “MessageID” to move your received message from your inbox to your recycle bin with the method “moveMessage()”.
5. Always consult the previous history of your message through a “MessageID” with the method “getMessageHistory()”.
6. Use a “MessageID” from your sent box to consult the state of that message with “getMessageAcknowledgmentsStatus()”.
7. Consult all your messages from all your eHealthboxes with “GetAllEhboxesMessagesList()”. Execute this request on your inbox, sent box and on your bin.
8 Error and failure messages

8.1 Error Element Status Codes

Mail delivery system error codes originating from the application:
These error codes first indicate a problem with the message and or its recipients. This table can evolve.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Component</th>
<th>Description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>700</td>
<td>Error</td>
<td>Unknown technical error</td>
<td>Call the Contact Center</td>
</tr>
<tr>
<td>701</td>
<td>Error</td>
<td>Business validation error</td>
<td>Check all fields, especially the recipients. The publication id is found under PublicationId.</td>
</tr>
<tr>
<td>702</td>
<td>Error</td>
<td>Duplicate publication id</td>
<td>Chose a new publication id, and send again. The duplicate publication id can be found under PublicationId.</td>
</tr>
<tr>
<td>703</td>
<td>Error</td>
<td>One or more recipients are invalid</td>
<td>Invalid recipients can be found under Destination.</td>
</tr>
</tbody>
</table>

8.2 Consultation Response Status Codes

Error codes originating from the eHealth platform:
These error codes first indicate a problem in the sent arguments. This table can evolve.

<table>
<thead>
<tr>
<th>Error code</th>
<th>Component</th>
<th>Description</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>*</td>
<td>SUCCESS</td>
<td></td>
</tr>
</tbody>
</table>
| 806        | * / MessageId   | The specified MessageId is invalid; please verify that the Source and the MessageId are correct and that you can access it. | Is the MessageId correct? Is the message present in “Source”?
| 807        | * / Start & EndIndex | EndIndex must be larger or equal to StartIndex; please correct StartIndex and EndIndex. | Is “EndIndex > StartIndex”?
| 808        | * / Start & EndIndex | A maximum of 100 messages can be returned by request; please correct StartIndex and EndIndex. | Is “EndIndex - StartIndex + 1 <= 100”?
<p>| 809        | GetMessage- AcknowledgmentsStatus/ MessageId | The specified MessageId is invalid; please verify that the MessageId is correct and that you are the sender. | Are you the sender of the message? Is the MessageId correct? |
| 810        | * / BoxId       | The specified BoxId is invalid; please verify the data and that you can access it. | Can you normally access that eHealthBox? |</p>
<table>
<thead>
<tr>
<th>Code</th>
<th>Scenario</th>
<th>Description</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>812</td>
<td>MoveMessage/ Source &amp; Destination</td>
<td>You cannot move a message from your Inbox to your Sent box (even via recycle bin) and vice versa.</td>
<td>Is the message present in “Source”?</td>
</tr>
<tr>
<td>813</td>
<td>MoveMessage/ Source &amp; Destination</td>
<td>Not all messages were moved successfully. Please verify for each message that the Source and the MessageID are correct. Also pay attention that a message in the recycle bin which was moved from the Inbox cannot be restored back to the Sent box and vice versa.</td>
<td>Some messages where not found in the folder specified in “Source”. Some messages cannot be moved to “Destination”</td>
</tr>
<tr>
<td>815</td>
<td>DeleteMessage</td>
<td>Not all messages were deleted successfully. Please verify for each message that the Source and MessageID are correct.</td>
<td>Not deleted messages are listed in MessageId in the response.</td>
</tr>
<tr>
<td>820</td>
<td>InsertOoO</td>
<td>The period 01/07/2012 to 12/07/2012 is invalid because it overlaps another period.</td>
<td>Correct start or end date.</td>
</tr>
<tr>
<td>821</td>
<td>InsertOoO</td>
<td>The end of the period cannot be further than a year in the future.</td>
<td>Correct end date.</td>
</tr>
<tr>
<td>822</td>
<td>InsertOoO</td>
<td>The start date cannot be after the end date.</td>
<td>Correct start date.</td>
</tr>
<tr>
<td>823</td>
<td>InsertOoO</td>
<td>The start date cannot be in the past.</td>
<td>Correct start date.</td>
</tr>
<tr>
<td>824</td>
<td>InsertOoO</td>
<td>One or more substitutes cannot be chosen because they are absent.</td>
<td>Change one or more substitutes.</td>
</tr>
<tr>
<td>825</td>
<td>InsertOoO</td>
<td>The number of substitutes may not exceed 5.</td>
<td>Remove a substitute.</td>
</tr>
<tr>
<td>826</td>
<td>InsertOoO</td>
<td>The number of out of office for one eHealthBox may not exceed 10.</td>
<td>Delete and out of office.</td>
</tr>
<tr>
<td>827</td>
<td>InsertOoO</td>
<td>One or more substitutes are unknown or not correct, please correct them.</td>
<td>Correct one or more substitutes.</td>
</tr>
<tr>
<td>828</td>
<td>InsertOoO</td>
<td>The user is unknown or not correct, please correct him.</td>
<td>Re-authenticate.</td>
</tr>
<tr>
<td>829</td>
<td>InsertOoO</td>
<td>A valid substitute is a person, not an organization.</td>
<td>Change one or more substitutes.</td>
</tr>
<tr>
<td>830</td>
<td>InsertOoO</td>
<td>A person cannot be substitute for himself.</td>
<td>Change one or more substitutes.</td>
</tr>
<tr>
<td>840</td>
<td>DeleteOoO</td>
<td>One or more OoOld are invalid.</td>
<td>Verify the OoOld’s.</td>
</tr>
</tbody>
</table>
### 8.3 Soap Fault Error Codes

They contain the following attributes:

<table>
<thead>
<tr>
<th>Field name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>Unique number identifying this message. If present, the ticket that was created for the client's request, leading to this error. When placed here this Id can be used to trace back the message from the request.</td>
</tr>
<tr>
<td>Origin</td>
<td>The component/party causing the error: consumer or provider, client or server.</td>
</tr>
<tr>
<td>Code</td>
<td>The error code</td>
</tr>
<tr>
<td>Message</td>
<td>A human readable message</td>
</tr>
<tr>
<td>Retry</td>
<td>An optional Boolean that indicates if it is worth resending the same request.</td>
</tr>
<tr>
<td>Contact</td>
<td>An optional field specifying a contact description.</td>
</tr>
<tr>
<td>Environment</td>
<td>The eHealth platform environment in which the error occurs: integration, acceptance or production.</td>
</tr>
</tbody>
</table>
8.3.1 Schema Validation Errors

When invoking the WS, you must provide a valid XML.
Before executing any action, the eHealthBox system verifies if the XML is valid by running a validation check towards the SendMessageRequest XSD.

If the validation fails, a SOAP Fault is returned with the following code and message:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA-03006</td>
<td>XSD compliance failure</td>
</tr>
</tbody>
</table>

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:id="id-6">
    <soapenv:Fault>
      <faultcode>soapenv:Client</faultcode>
      <faultstring>SOA-03006</faultstring>
      <detail>
        <soa:SystemError xmlns:soa="urn:be:fgov:ehealth:errors:soa:v1" id="5bbd8a2a-bb21-4cf8-99bc-8d52c18e2801">
          <Origin>Consumer</Origin>
          <Code>SOA-03006</Code>
          <Message xml:lang="en">XSD compliance failure.</Message>
          <soa:Environment>Production</soa:Environment>
        </soa:SystemError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

### 8.3.2 Technical Errors

Technical errors are errors inherent to the internal working of the eHealth platform WS. These errors can also occur if the token used to call the WS is not valid.

They contain the standard SOAP Fault attributes.

The table provides the different codes and messages returned in a SOAP fault message:

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOA-00001</td>
<td>An internal error has occurred. Please contact the Contact Center.</td>
</tr>
</tbody>
</table>

This list can evolve.

Example:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
    <env:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>SOA-00001</faultstring>
      <detail>
```
<soa:SystemError Id="ec582704-d623-4b05-ab7f-98d5c9706dd1"
xmlns:soa="urn:be:fgov:ehealth:errors:soa:v1">
  <Origin>Server</Origin>
  <Code>SOA-00001</Code>
  <Message xml:lang="en">An internal error has occurred. Please contact service desk.</Message>
  <soa:Environment>Production</soa:Environment>
</soa:SystemError>
</detail>
</env:Fault>
</env:Body>
</soapenv:Envelope>