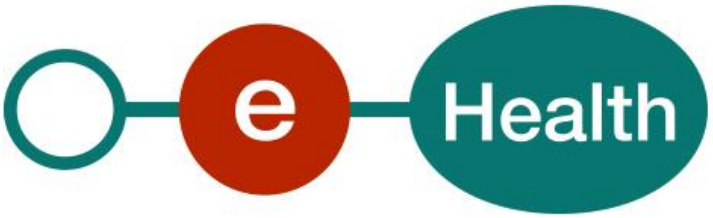


eHealthBox v2.0 Publication Web Service
Cookbook
Version 2.4

This document is provided to you free of charge by the

eHealth platform

Willebroekkaai 38 – 1000 Brussel
38, Quai de Willebroek – 1000 Bruxelles

All are free to circulate this document with reference to the URL source.

Table of contents

To the attention of: “IT expert” willing to integrate this web service.

1. Document management

1.1 Document history

Version	Date	Author	Description of changes / remarks
2.2	06/05/2013	eHealth platform	Annex v1.0 20171026.docx
2.3	19/09/2017	eHealth platform	Update list of qualities Annex v1.0 20171026.docx
2.4	15/01/2018	eHealth platform	Externalize SSO and quality specification

2. Introduction

2.1 Goal of the service

The eHealthBox Publication WS allows an authenticated user to publish an (encrypted) eHealthBox message (Document or News) for different addressees.

The publication request is received by the eHealthBox central systems and processed ***asynchronously***.

This means that a successful response does not guarantee that the message will be correctly published at the end of the process. A publication failure can occur later on due to the behaviour of external systems.

A successful response message only guarantees that the message will be processed.

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 They are out-of-date and should not be used by new partners for they do not provide any 'extra' feature.

The size of a message and of an eHealthBox is currently limited to 10MB, **on inbox and trash bin folder**. Note that an encrypted message weighs more due to the encryption overhead.

2.2 Goal of the document

This document provides functional and technical information about calling the eHealthBox Publication 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 platform WS. An example illustrates each of those messages. You can find a list of possible errors further 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 web services 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; eHealth partners 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, eHealth 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.

2.3 New in version 2.0

- Send an encrypted message by placing the encrypted content in the '*Encryptable*' elements and by setting *IsEncrypted* to *true*.
- *BoxId* allows you to specify which of your eHealthBoxes (if you have more than one) you want to use.
- *CustomMeta* (*Key*, *value*) allow you to add free Meta Information to your message. It will be transparently transported to the recipient.
- Fields names have been improved for better understanding.

2.4 eHealth platform document references

All the document references can be found in the technical documentation on the portal of the eHealth platform[1]. These versions or any following versions can be used for the eHealth platform service.

[1] ***https://www.ehealth.fgov.be/ehealthplatform***

ID	Title	Version	Date	Author
1	Glossary.pdf	1.0	01/01/2010	eHealth platform
2	Cookbook STS	1.0	31/08/2010	eHealth platform
3	Cookbook ETEE voor bekende bestemmeling/destinataire connu	2.3	06/05/2011	eHealth platform
4	Cookbook eHealthBox Consultation	2.2	30/05/2013	eHealth platform
5	eHBox_Quality	1.01	23/04/2018	eHealth platform
6	eHBox_SSO	1.01	23/04/2018	eHealth platform

2.5 Service history

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

Previous version	Previous release date	Changes
1.0	03/02/2011	Major changes: Encryption, Multi-box, Publication WebApplication, Consultation WebApplication reviewed, general reliability

3. Support

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*** (Dutch)
  - ***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***

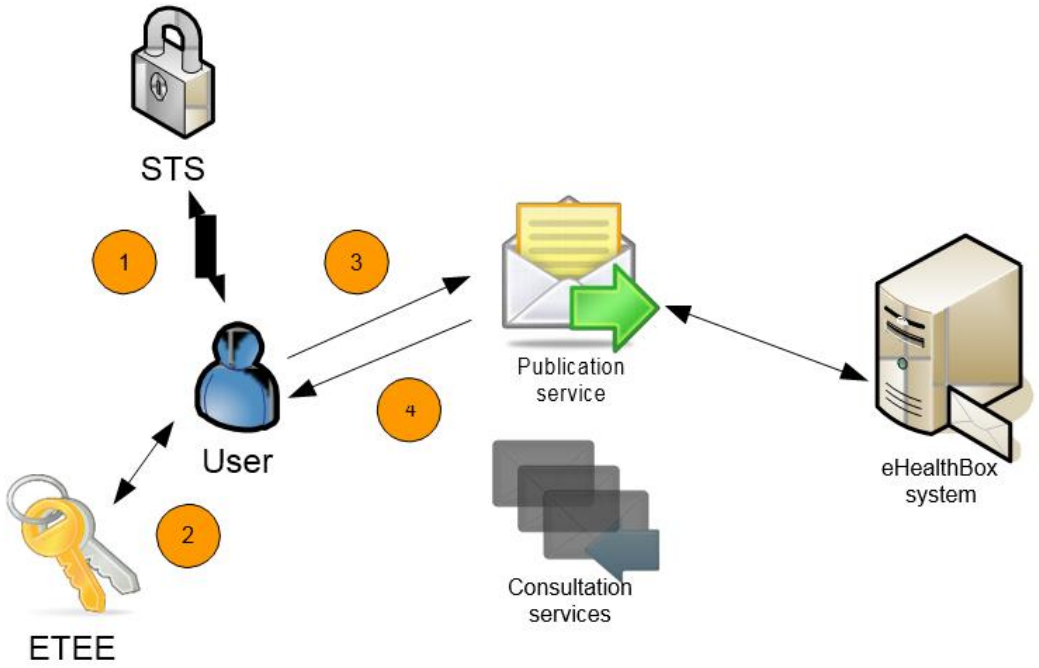
3.4 Certificates

- 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***
  - French version: ***https://www.ehealth.fgov.be/ehealthplatform/fr/certificats-ehealth***
- 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 how the publication web service is used.

Step 1. To use the Publication WS, you have to contact the STS WS to get a secure token containing the identification of the user (see 5.2 and cookbook STS).

Step 2. Optionally, if you want to encrypt the content of your message, you have to call method Get ETK of the ETK Depot WS (synonymous to the ETEE WS, see the cookbook: "ETEE for unknown recipient" on the eHealth platform portal) in order to get the public key(s) of the recipient(s) and use the Crypto Library.

Step 3. Once you have your secure token, you are able to use and contact the "Publication service" to publish your message(s).

Step 4. Once your message sent, the system will respond to you with a response message.

5. Step-by-step

5.1 Technical requirements

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

5.2 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 and eHBox_Quality.

To access the eHealth WS, 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.

5.2.1 Encryption

The message to send to the WS must be encrypted.

To encrypt the message, you should retrieve the public key on the ETK (eHealth Token Key) depot. Then, encrypt the message using this public key via eHealth encryption librairies.

Or/and

The message send by the WS shall be encrypted with your public key.

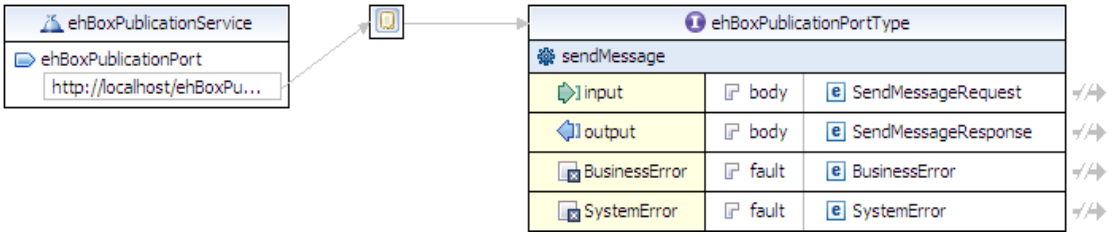
All the information about the use of the encryption libraries and the call to the eHealth Token Key (ETK) depot are described in the cookbooks available on the eHealth platform website.

Encrypted message convention: If an encrypted message is to be sent, ALL "*Encryptable*" fields MUST contain (one and all) encrypted content. In other words, first encrypt the content of each of those fields separately, then convert them to xs:base64Binary and finally set *IsEncrypted* to *True*. If *IsEncrypted* is set to *True*, and non-encrypted content is being transported, unexpected errors can occur! Conversely, if *IsEncrypted* is set to *False*, and encrypted content is being transported, unexpected errors can occur! You cannot choose to encrypt some "*Encryptable*" fields solely.

5.3 Process overview

Technical information is to be found on the Registry website of the eHealth platform,

https://services.ehealth.fgov.be/registry/uddi/bsc/web



The important sections of the WSDL (Web Service Definition Language) of the Publication Web Service are:

- The applicable **Policies**, which cover the **MTOM** (file upload) and **security** aspects.
- The types (**SendMessageRequest** and **SendMessageResponse**) that are used by the **sendMessage** method. The fault message is also defined.

- The **sendMessage** method. (The ping method is only used for the monitoring of the WS)

5.4 eHealthBox Publication WS

5.4.1 Lifetime of a message

- When the expiration date of a message is reached and it has already been placed in the recycle bin after it has been read, the message 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.

5.4.2 SendMessage Method

The request that must be sent in order to publish the eHealthBox content for one or more recipients will be discussed here. You can send a message of the type "Document" or "News item". A document contains a PublicationId that you define and cannot be used twice. A News item, on the other hand, can be updated by sending a new news item with the same PublicationId. This is the main difference between a "Document" and "News item" message type.

You can send a message to a particular person or to multiple persons by specifying one *DestinationContext* per person. You can also send a message to a list of persons sharing the same "Quality". Please refer to Annex 1 – Publish a message to a list of professionals. In order to be able to send a message to all persons sharing the same quality (all nurses, or all doctors), prior authorization from the eHealth platform (***info@ehealth.fgov.be***) is required. The eHealth platform will investigate your request, and then allow you to publish this type of message.

When sending a message via the Publication WS, the identification of the sender is retrieved from the STS Token by default. The recipient will see the identifier of the sender when retrieving the message via the consultation WS. If the sender is a healthcare professional, his SSIN will be retrieved from the STS token and used as identifier for the sender. If the sender was a hospital or any other healthcare organization, the identifier in the STS token will be of the type NIHII. Finally, if the sender was an enterprise, the identifier type will be a CBE. Exception is made if BoxId (optional element) is filled in. In that case, the person connected to the WS is sending the message "in name of the person or organization" specified in the element BoxId. Thus, if a healthcare professional (a person) does want to send a message using his NIHII (INAMI) number in place of his SSIN number, he has to submit the request and fill his NIHII number in BoxId.

For example, the sender is connected as a doctor with an STS token containing his INSS number as attribute:

```
Id = 12345678910
Type = INSS
Quality = DOCTOR
```

But the sender does not want to display his SSIN number to the recipient, since he prefers to display his NIHII number; he then uses the BoxId element in the request, and fills it out as follows:

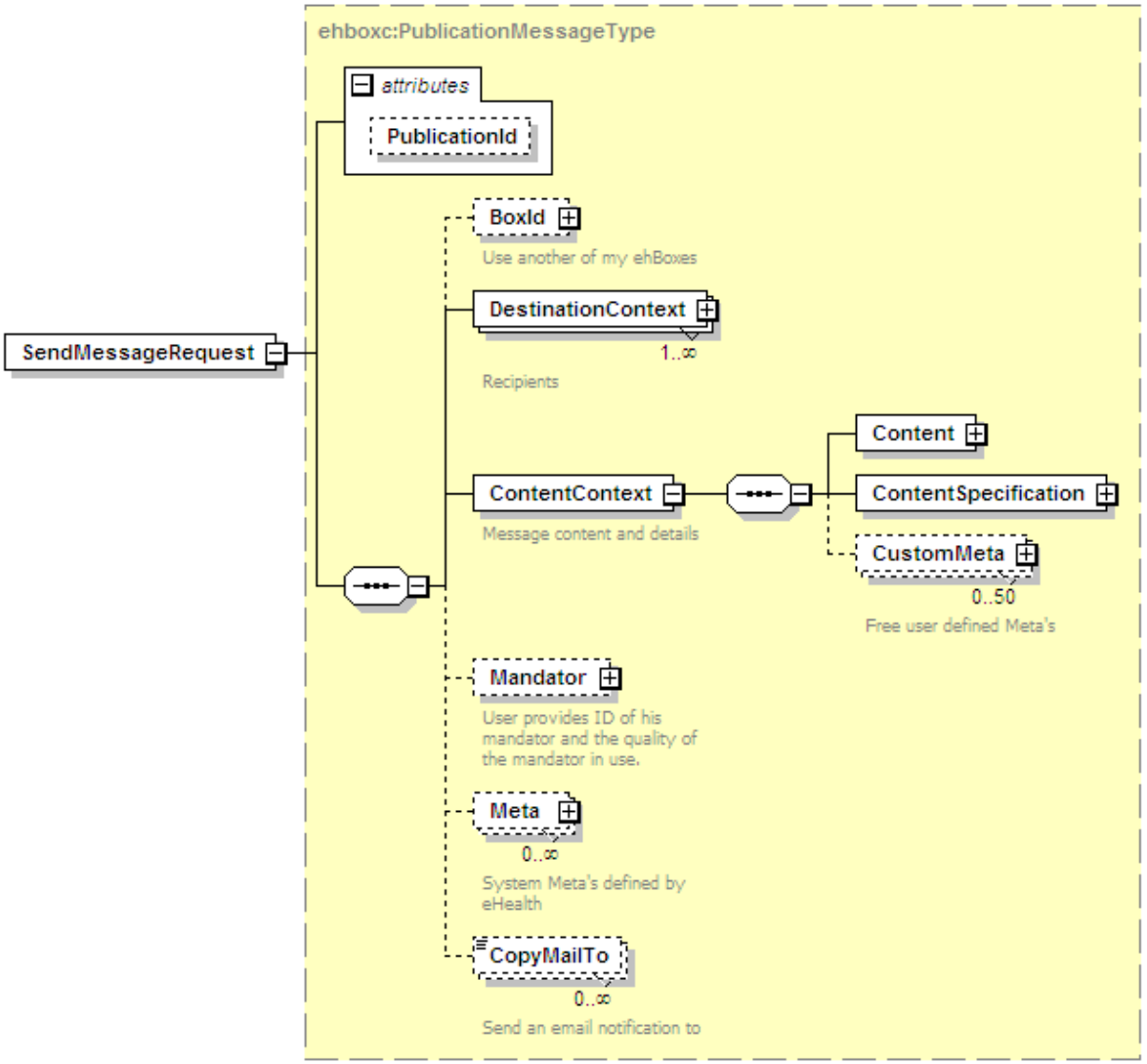
```
<BoxId>
    <Id>104025888</Id>
    <Type>NIHII</Type>
    <Quality>DOCTOR</Quality>
</BoxId>
```

The sender will then be stored in the application with the data found in BoxId (after access validation).

Note: a NIHII number can consist out of eleven numbers (with the three qualification numbers) or eight numbers (without the three qualification numbers).

5.4.2.1 SendMessage Request

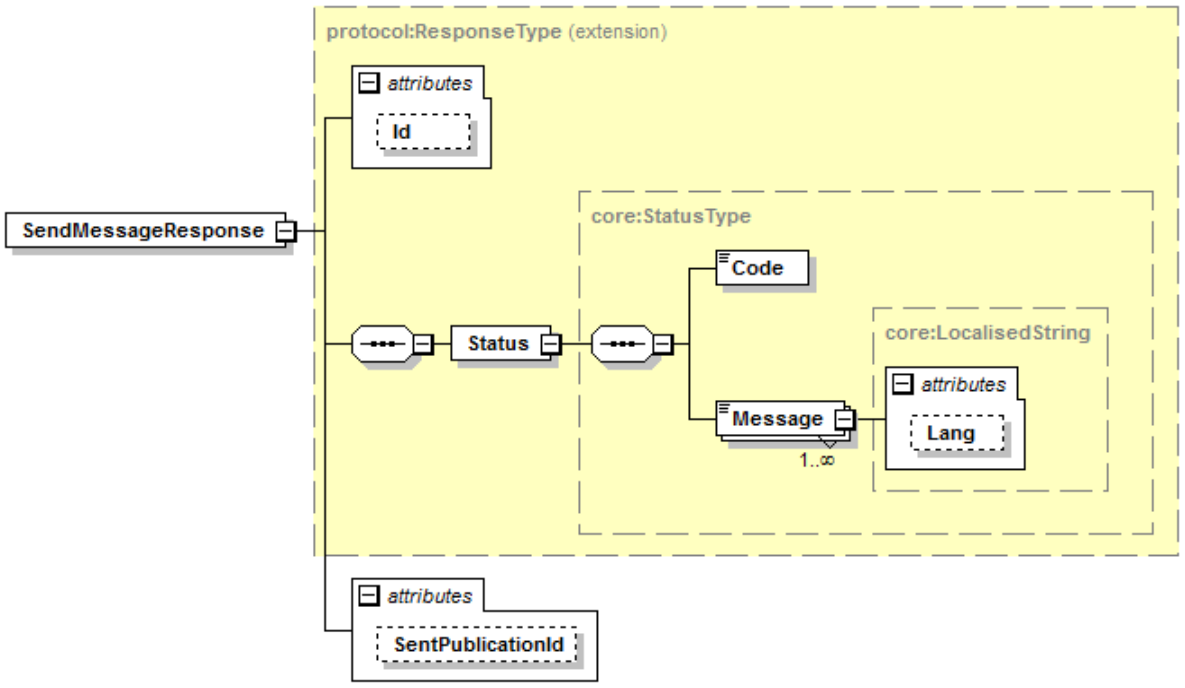
You will find all mandatory information about the allowed combinations Id-Type-SubType-Quality in the eHBox_Quality v1.01.



Field name	Description
PublicationId	The identification of the publication as defined to the client. Any alphanumeric string (minimum 1, maximum 13) can be used. The use of the PublicationId depends on the content type: document or news item (see further). If the content is a "document", the PublicationId must be unique. If you try to publish two documents with the same PublicationId, the second will not be published. If the content is a "news item", you can use the same PublicationId to update your news content. When you publish two news with the same PublicationId, the first will be added as history and replaced by the second in the inbox folder. You have to be careful, if you first send a news item to x recipients, and then update your news item by sending a new content with the same PublicationId, all the recipients will receive the updated content.
BoxId	If the client wants to use another of his eHealthBoxes, he can specify it here. The message will then be sent as if the authentification was made with this eHealthBox. This avoids the client having to re-authenticate himself each time.
DestinationContext	The *DestinationContext* is a complex type that contains information about the recipients. This type is detailed in section 5.4.3.7. A *SendMessageRequest* can have numerous *DestinationContexts* (numerous recipients).
ContentContext	The *ContentContext* is a complex type that contains the message content. This type is detailed in section 5.4.3.4.
Mandator	**Obsolete, do not use.**
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 5.4.3.10).
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 UPPAD . ***https://www.ehealth.fgov.be/fr/esante/professionnels-de-la-sante/uppad*.**

5.4.2.2 SendMessage Response

We describe here how the response returns for the request mentioned above. The response contains a success status code if no major error could be found in the request. Still, **remember that the system is asynchronous: a successful response does not guarantee that the message will be correctly published at the end of the process.** Possible Business Errors returned by the method are described in chapter 8.



Field name	Description
Id	The *Id* is attributed to the eHealthBox message by the eHealthBox system. This is a unique number that is used to identify the message in the eHealthBox system. This Id will be provided to the recipient by the consultation web service.
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'. **Remember that the system is asynchronous: a successful response does not guarantee that the message will be correctly published at the end of the process.** Otherwise, in case of a business error: • The Code will be an error code, which unequivocally identifies the problem (see further). • 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).
SentPublicationId	The PublicationId provided in the SendMessageRequest message allows the grouping of the requests/responses. This is a synonym for the *MessageId* used in all other requests and responses. String 13 digits.

5.4.2.3 Example

The following example does not contain the SAML assertion (See eHBox_SSO v1.1, eHBox_Quality v1.01 and SSO cookbook).

Request:

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:urn="urn:be:fgov:ehealth:ehbox:publication:protocol:v2">
    <soapenv:Header/>
    <soapenv:Body>
        <urn:SendMessageRequest PublicationId="1009927801393">
            <BoxId>
                <Id>13033577799</Id>
                <Type>INSS</Type>
                <Quality>DOCTOR</Quality>
            </BoxId>
            <DestinationContext>
                <Id>10099278</Id>
                <Type>NIHII</Type>
                <SubType>HOSPITAL</SubType>
                <Quality>HOSPITAL</Quality>
            </DestinationContext>
            <DestinationContext>
                <Id>23022211</Id>
                <Type>NIHII</Type>
                <Quality>DOCTOR</Quality>
            </DestinationContext>
            <ContentContext>
                <Content>
                    <Documen
                       t>
                        <Title>DocumentTitle</Title>

<EncryptableBinaryContent>UjBsR09EbGhjZ0dTQUxNQUFBUUNBRU1tQ1p0dU1GUXhEUzhi
<
/EncryptableBinaryContent>
                        <DownloadFileName>test.txt</DownloadFileName>
                        <MimeType>application/octet-stream</MimeType>
                        <Signing SigningType="sha256">

<SigningDownloadFileName>signature.sha</SigningDownloadFileName>

<TextSigningContent>c2lnbmF0dXJl</TextSigningContent>
                        </Signin
                           g>
                    </Document>
                    <FreeInformation
                           s>

<EncryptableFreeText>SW5mb3JtYXRpb24=</EncryptableFreeText>
                    </FreeInformations>

<EncryptableINSSPatient>ODQxMjA4MjI3NjI=</EncryptableINSSPatient>
                    <Annex>
                        <EncryptableTitle>YW5uZXggMQ==</EncryptableTitle>
```

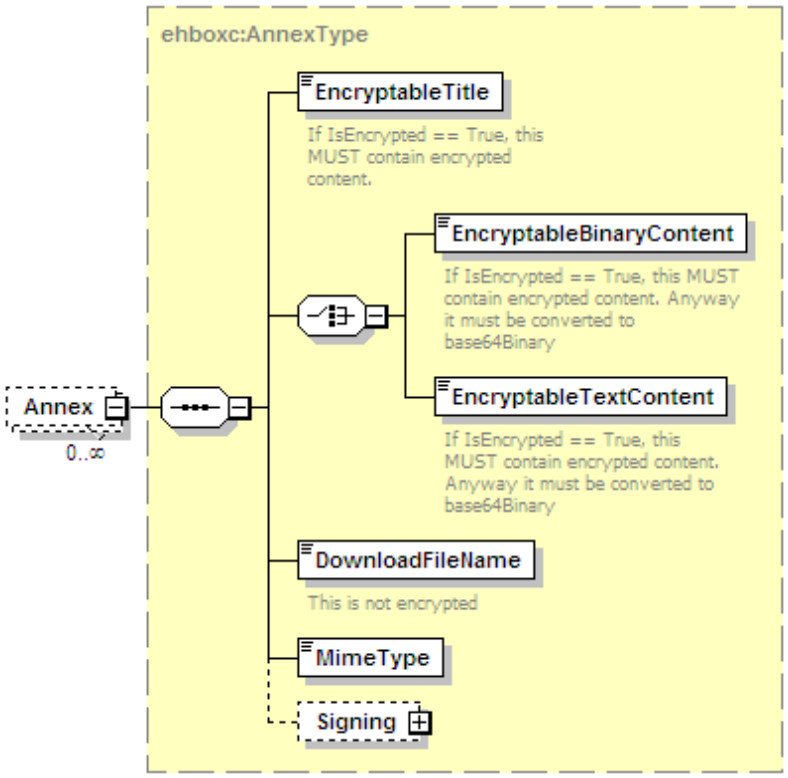
```
<EncryptableTextContent>YW5uZXg=</EncryptableTextContent>
                        <DownloadFileName>annexname.txt</DownloadFileName>
                        <MimeType>text/plain</MimeType>
                    </Annex>
                </Content>

                <ContentSpecification>
                    <IsImportant>true</IsImportant>
                    <IsEncrypted>true</IsEncrypted>
                    <PublicationReceipt>true</PublicationReceipt>
                    <ReceivedReceipt>true</ReceivedReceipt>
                    <ReadReceipt>true</ReadReceipt>
                </ContentSpecification>
            </ContentContext>
            <CopyMailTo>me@test.be</CopyMailTo>
        </urn:SendMessageReques
                    t>
    </soapenv:Body>
</soapenv:Envelope>
```

Response:

```
<ehboxp:SendMessageResponse Id="5H0111043267W"
SentPublicationId="1009927801393"
xsi:schemaLocation="urn:be:fgov:ehealth:ehbox:publication:protocol:
v2 ehealth-ehBox-publication-schema-protocol-2_0.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ehboxp="urn:be:fgov:ehealth:ehbox:publication:protocol:v2">
    <Status>
        <Code>100</Code>
        <Message Lang="EN">SUCCESS</Message>
    </Status>
</ehboxp:SendMessageResponse>
```

5.4.3 Used types

5.4.3.1 Annex

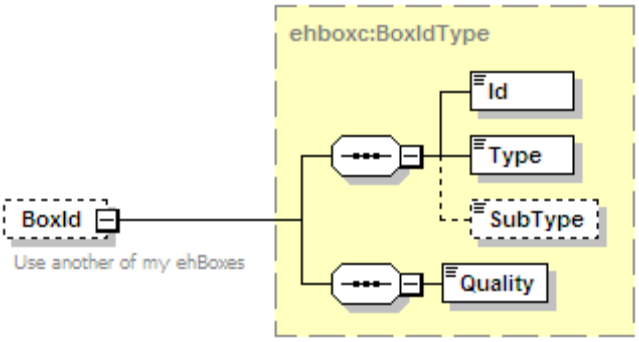


Field name	Descriptions
EncryptableTitle	An *Annex* has an *EncryptableTitle*, a human readable description of its content (string minimum 1, maximum 400). If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
EncryptableBinaryContent	A base64-encoded binary content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
EncryptableTextContent	A base64-encoded text content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
DownloadFileName	E.g. “principal.pdf” (string minimum 1, maximum 80).
MimeType	Represents the mime type of the content. E.g. “application/pdf”,” text/plain”, “application/octet-stream” (string minimum 1, maximum 50).
Signing	See section 5.4.3.12

5.4.3.2 BoxId

A *BoxId* contains all the information on the eHealthBox the client wants to use for the request.

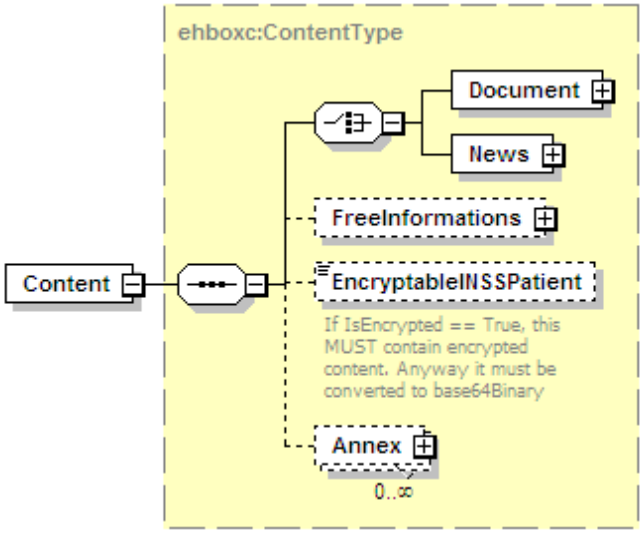
You will find all mandatory information about the allowed combinations Id-Type-SubType-Quality in the eHBox_Quality



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)

5.4.3.3 Content

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

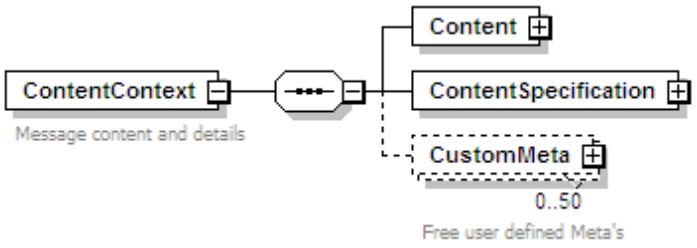


Field name	Descriptions
Document	See section 5.4.3.8
News	See section 5.4.3.11
FreeInformations	See section 5.4.3.9

EncryptableINSSPatient	This optional field allows specifying an INSS number of a patient concerned by the message content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
Annex	See section 5.4.3.1

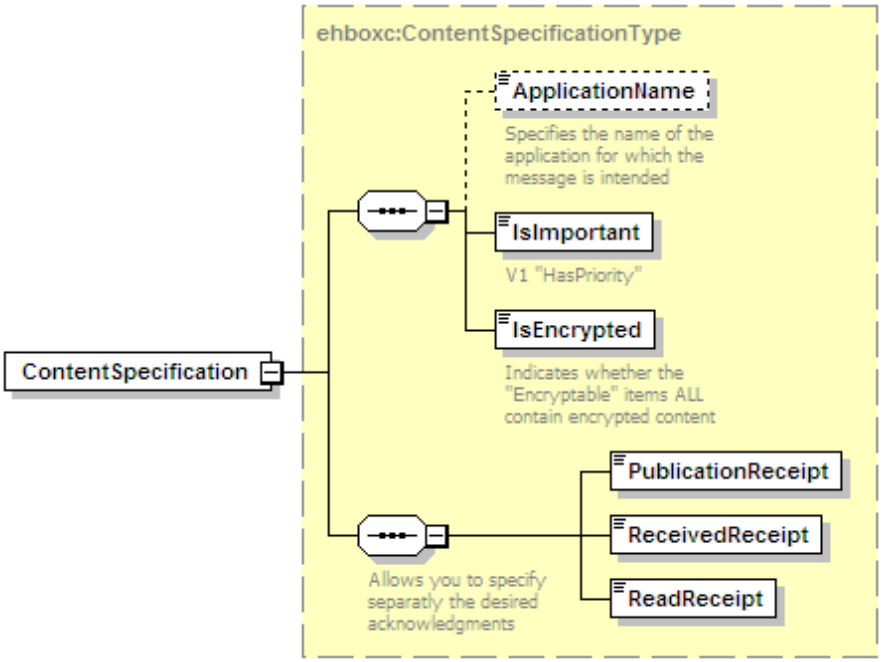
5.4.3.4 ContentContext

A *ContentContext* contains the message content and message details, as well as zero-or-more (50 maximum) free *CustomMeta*s. The user can freely specify these CustomMetas when for internal usage. You can define a Key and a value for each *CustomMeta* (see 5.4.3.6).



Field name	Descriptions
Content	See section 5.4.3.3.
ContentSpecification	See section 5.4.3.5.
CustomMeta	See section 5.4.3.6

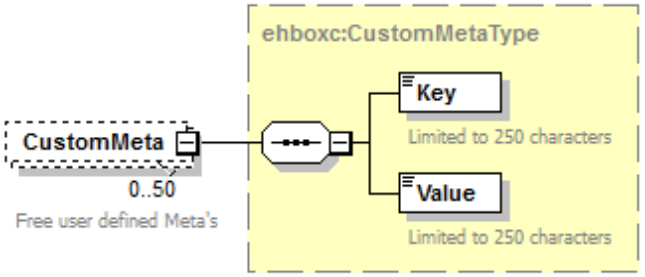
5.4.3.5 ContentSpecification



Field name	Descriptions
ApplicationName	The Application sending the message (optional, string minimum 1, maximum 25).
IsImportant	Boolean (true or false) that indicates if the message is to be considered as important.
IsEncrypted	Boolean (true or false) that indicates if the content has been encrypted.
PublicationReceipt	Boolean (true or false) that indicates if a publication receipt is requested. A message is returned to the sender's eHealthBox when the message is persisted in database. Still, the recipient may not be able to view it if his eHealthBox is full.
ReceivedReceipt	Boolean (true or false) that indicates if a received receipt is requested. A message is returned to the sender's eHealthBox when the recipient viewed the message (when he calls *GetMessagesList*).
ReadReceipt	Boolean (true or false) that indicates if a read receipt is requested. A message is returned to the sender's eHealthBox when the recipient read the message (when he calls *GetFullMessage*).

5.4.3.6 CustomMeta

CustomMeta was invented 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".

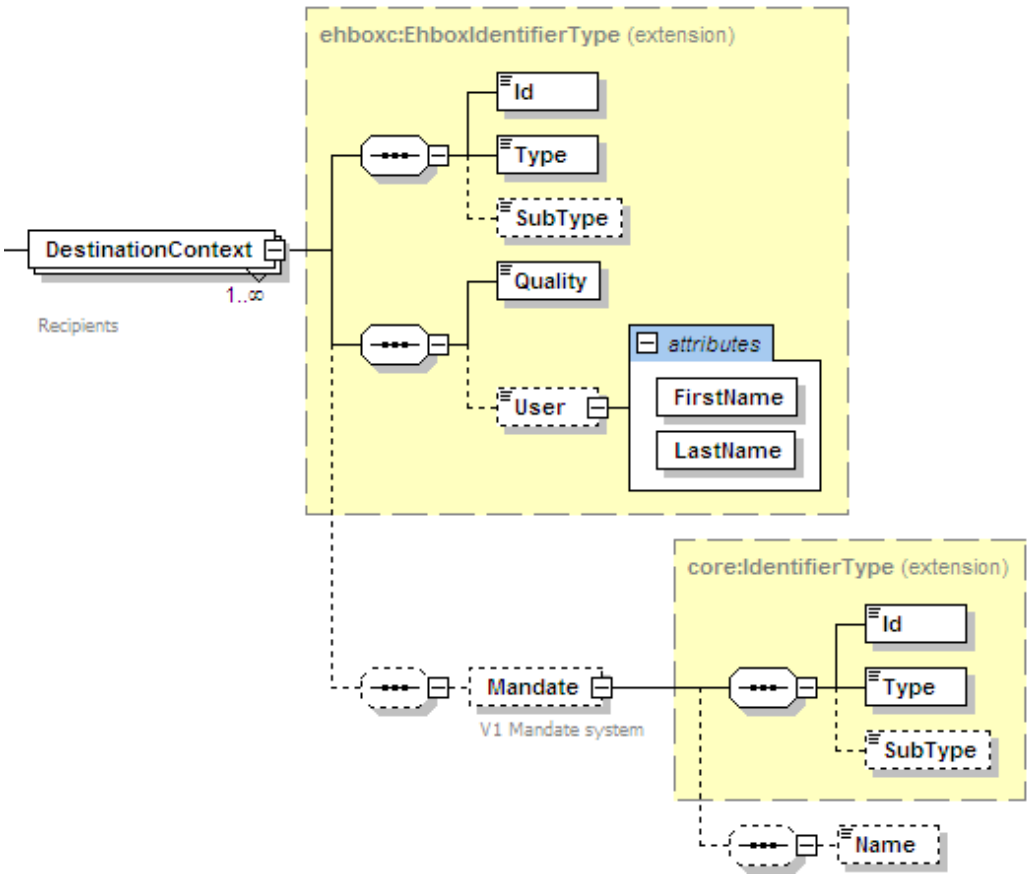


Field name	Descriptions
Key	Alphanumeric string used as a key (string minimum 1, maximum 250).
Value	Alphanumeric string value corresponding to the *Key* (string minimum 1, maximum 250).

5.4.3.7 DestinationContext

A *DestinationContext* contains all the information on the recipient.

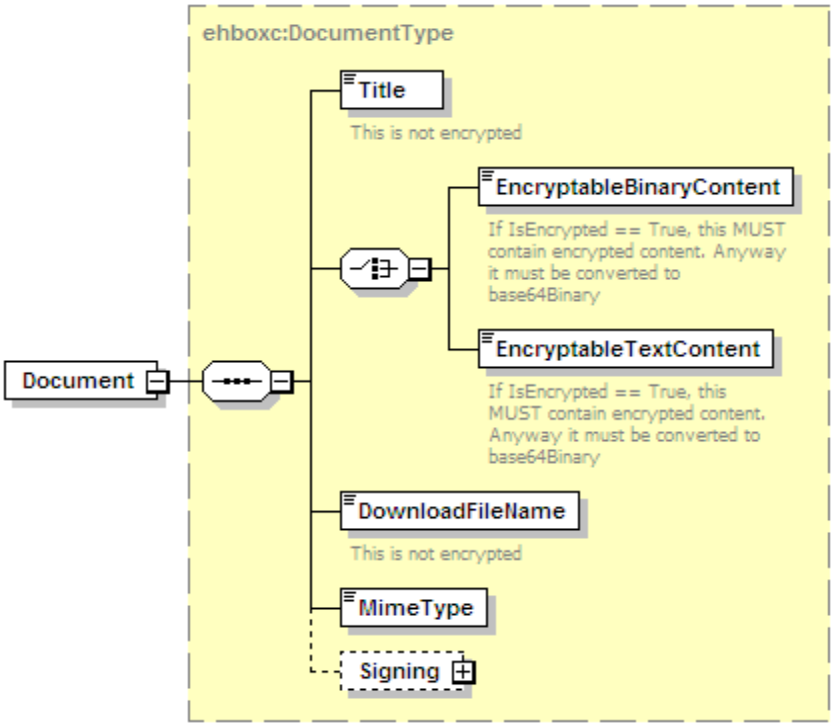
You will find all mandatory information about the allowed combinations Id-Type-SubType-Quality in the eHBox_Quality. If you rather would like to publish your document to a list of recipients sharing the same "Quality" (e.g. to all Nurses), please refer to Annex 1 – Publish a message to a list of professionals.



Field name	Descriptions
ID	The recipient's identification number or "ALL" (publication to a "*Quality*"). This is a digital number representing an INSS, NIHII, FAMPH, or CBE. This number is in String format.
Type	The recipient'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	A *Quality* defines the recipient's eHealthBox and is mandatory. String.
User	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).
Mandate (obsolete)	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 "HOSPITAL" *SubType* for a Hospital *Quality*, or "GROUP" *SubType* for a Group *Quality*). The recipient's name may be specified.

5.4.3.8 Document

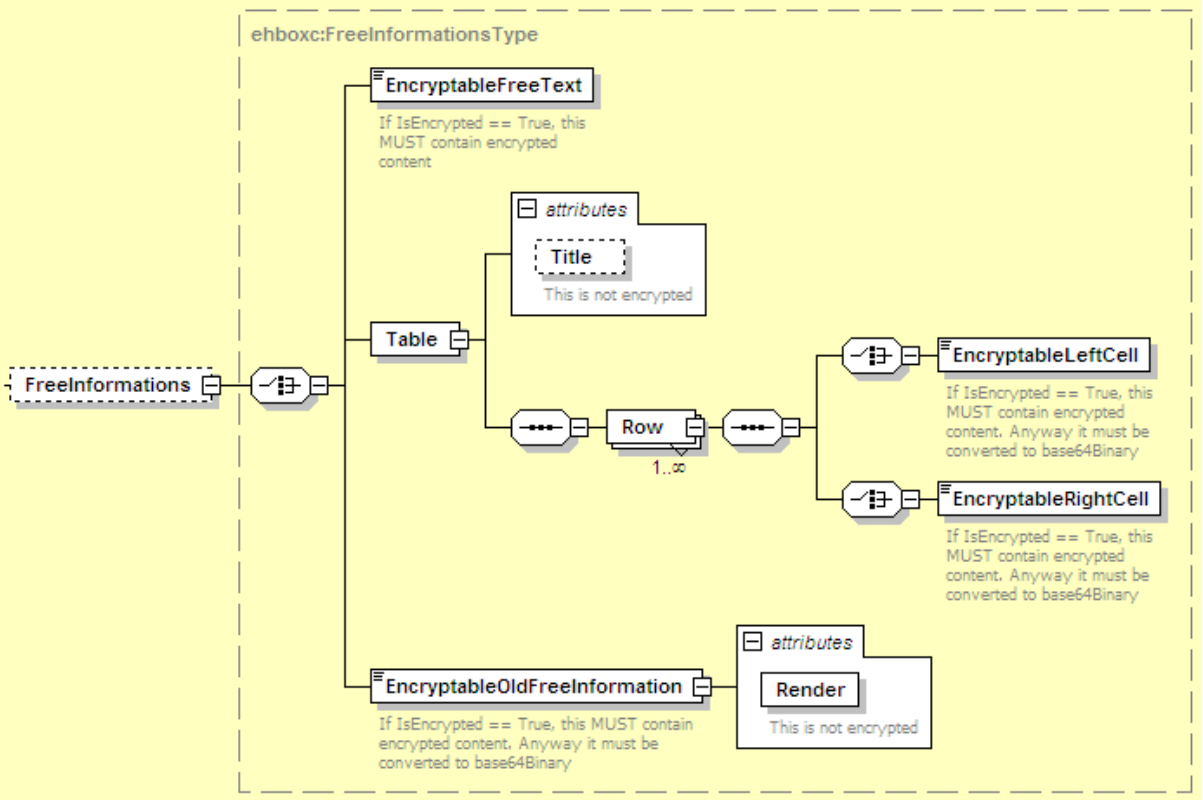
Please note that a message will contain either a News item or a Document, not both.



Field name	Descriptions
Title	A Document has a *Title,* a human readable description of its intent (string minimum 1, maximum 400).
EncryptableBinaryContent	A base64-encoded binary content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
EncryptableTextContent	A base64-encoded text content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
DownloadFileName	E.g. “principal.pdf” (string minimum 1, maximum 80).
MimeType	Represents the mime type of the content. E.g. “application/pdf”,” text/plain”, “application/octet-stream” (string minimum 1, maximum 50).
Signing	See section 5.4.3.12

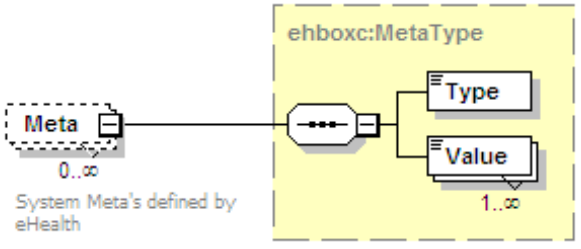
5.4.3.9 FreeInformations

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



Field name	Descriptions
EncryptableFreeText	Contains any alphanumeric string. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
Table	*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 5.4.3.5). The content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
EncryptableOldFreeInformation (obsolete)	A base64-encoded content and a Render attribute (as a string). If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).

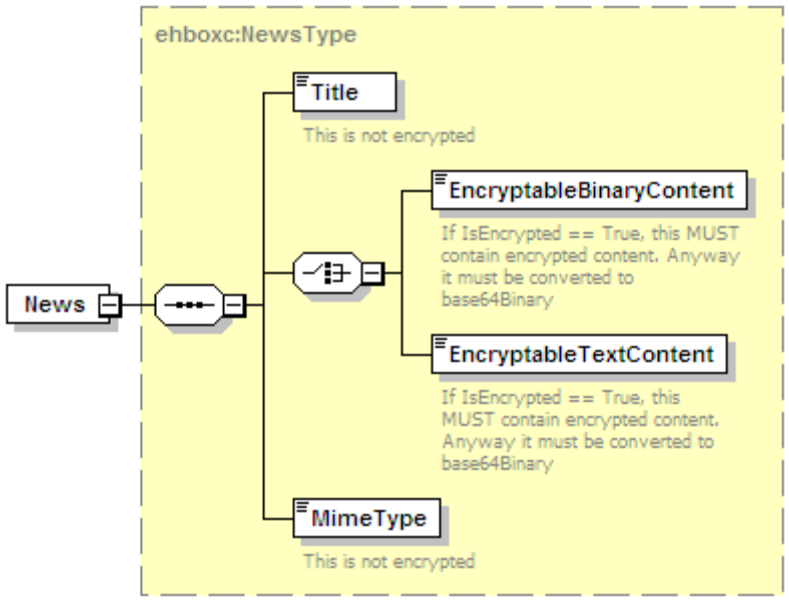
5.4.3.10 Meta



Field name	Descriptions
Type	The type of the meta information (string minimum 1, maximum 250).
Value	A list of *Value*s for this *Type* (string minimum 1, maximum 250).

5.4.3.11 News

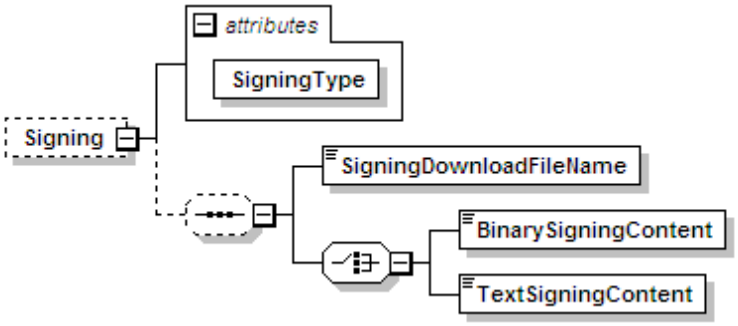
Please note that if you publish a News item, you should not publish a Document at the same time.



Field name	Descriptions
Title	A Document has a *Title*, a human readable description of its intent (string minimum 1, maximum 400).
EncryptableBinaryContent	A base64-encoded binary content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
EncryptableTextContent	A base64-encoded text content. If *IsEncrypted* is true (see Section 5.4.3.5), the content must be encrypted before being converted to xs:base64Binary (see section 5.2.1).
MimeType	Represents the mime type of the content. E.g. “application/pdf”,” text/plain”, “application/octet-stream” (string minimum 1, maximum 50).

5.4.3.12 Signing

To ensure data integrity, the sender can sign the content and provide the following security *Signing* information.



Field name	Descriptions
SigningType	The type of signature used. E.g. “PKCS”, “sha256” (mandatory, string minimum 1, maximum 50).
SigningDownloadFileName	The name of the signing file. E.g. “signature.sha” (string minimum 1, maximum 80).
BinarySigningContent	The signature of the BinaryContent. xs:base64Binary.
TextSigningContent	The signature of the TextContent. xs:base64Binary.

6. Risks and security

6.1 MTOM Policy

For binary content sending, the "Message Transmission Optimization Mechanism" (MTOM/XOP) should be used.

See ***http://www.w3.org/TR/soap12-mtom/ for the technical specification***.

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 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 Chap 3)

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 web service that the eHealth platform delivered, he is obliged to contact and inform us immediately. He is not allowed to publish this bug or vulnerability in any case.

6.2.2 Web service

WS 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 eHBox_SSO for a more detailed description of the SSO Access in the case of eHealth.
- SSL one way.
- an X.509 certificate. This certificate will contain the identifiers of the caller: INSS or NIHII number or CBE enterprise number. More information on how to obtain a certificate:
  French version: https://www.ehealth.fgov.be/fr/esante/professionnels-de-la-sante/gestion-des-certificats-ehealth.
  Dutch version: https://www.ehealth.fgov.be/nl/egezondheid/beroepsbeoefenaars-in-de-gezondheidszorg/beheer-van-de-ehealth-certificaten/algemene-voorstelling.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

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 doesn't 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)
  - 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.
A document explaining how to implement this security policy can be obtained on the website of the eHealth platform. This STS cookbook can be found on the portal of the eHealth platform.
https://www.ehealth.fgov.be/ehealthplatform/nl/data/file/view/240bf89245752bb69f1e188873c2af4d0c57a889?name=STS_HolderOfKey-Cookbookv1-2-13042018.pdf .

6.2.4 The use of username, password and token

The username, password and token are strictly personal. Partners and clients are not allowed to transfer them. Every user takes care of his username, password and token and he is forced to confidentiality of it. Moreover, every user is responsible of every use, which includes the use by a third party, until the inactivation.

7. Test and release procedure

7.1 Procedure

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

7.1.1 Initiation

If you intend to use the eHealth platform 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 in order to connect to our WS. Most of the required integration info to integrate is published on the portal of the eHealth platform.

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

7.1.3 Create test cases

7.1.3.1 Rules to access the Publication services are the same in test and in production. Access rules:

- To use the Publication services, the user must be part of one of the following profiles: **hospital, nurse, group, institution, doctor, laboratory …**
- authentication with a certificate

All test cases have to be configured by the eHealth development team.

7.1.3.2 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. You can request the test certificate through one of the following URLs:

- Dutch version: ***https://www.ehealth.fgov.be/nl/support/basisdiensten/ehealth-certificaten***
- French version: ***https://www.ehealth.fgov.be/fr/support/services-de-base/certificat-ehealth***

The process is described in the "How to request an eHealth test certificate".

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

7.1.3.3 Obtain SAML token

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

In the case of eHealthBox Publication web service, see **eHBox_SSO *access*** section.

7.1.3.4 Release procedure

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

For further information and instructions, please contact: ***info@ehealth.fgov.be***.

7.1.3.5 Operational follow-up

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

7.2 Test cases

This section describes a systematic process to test the Publication service. The eHealth platform recommends performing tests for all of the following cases:

- Publish a document message to the sender eHealthBox and get a successful response
- Publish a news item message to the sender eHealthBox and get a successful response.

In addition, the organization should also run negative test cases: publish a wrong xml input and get an error response message: *XSD compliance failure* (see section 8).

8. Error and failure messages

8.1 Send Message Response Status Codes

Error codes originating from the eHealth platform:

These error codes first indicate a problem in the arguments sent, or a technical error.

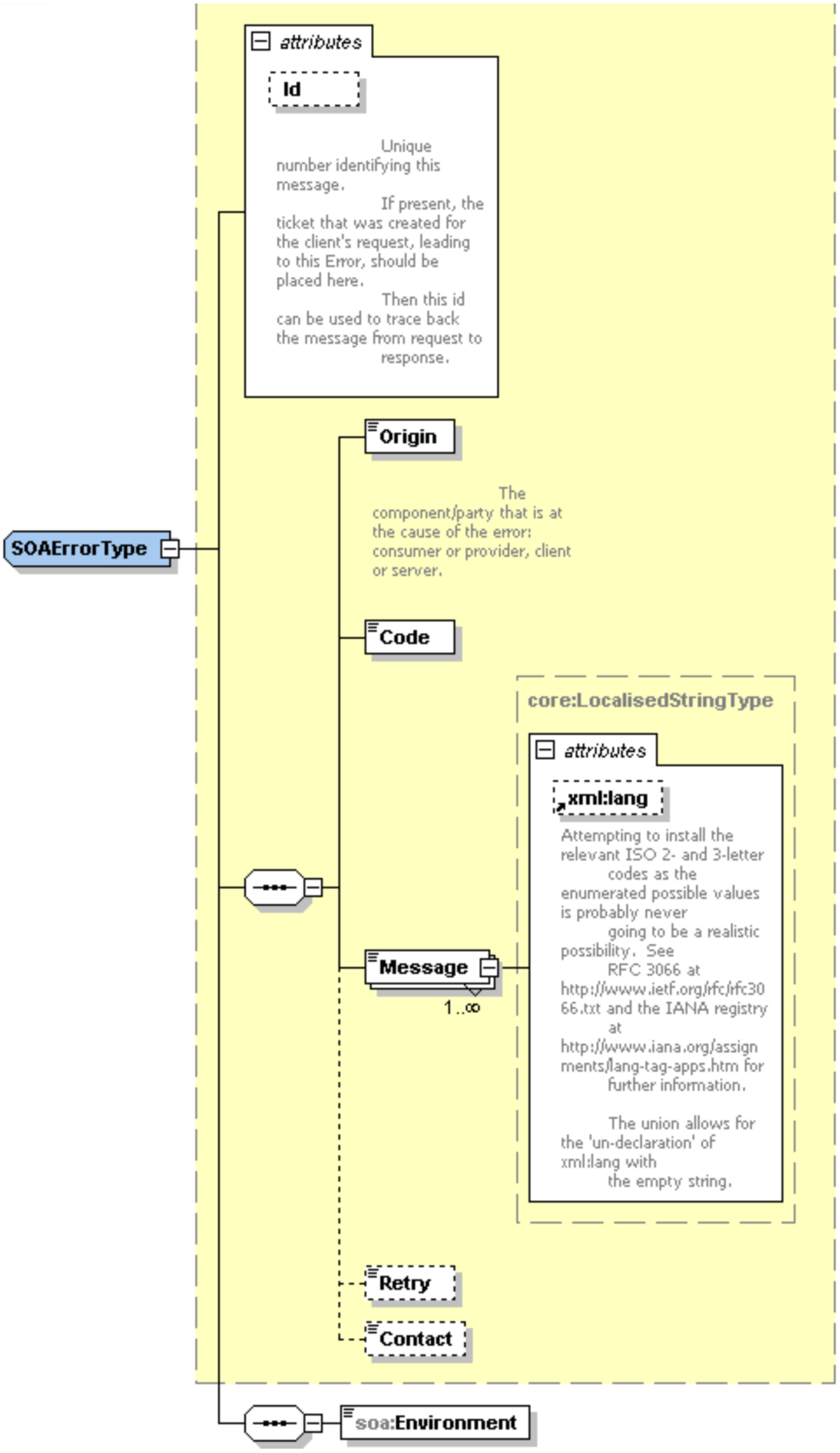
Error code	Component	Description	Solution
100	SendMessage	SUCCESS	
801	SendMessage	Your message exceeds the maximum size of 10485760 bytes (10Mb). Please lower the message size by deleting some appendixes or by splitting the message and send the message again. Please take into account that encryption may have increased the total message size.	Reduce appendixes size. Split the message and send multiple messages.
802	SendMessage/ DestinationContext	The specified Identifier is invalid; please verify the ID.	Is the NISS, NIHII, CBE valid and known by eHealth?
803	SendMessage/ DestinationContext	The specified Quality is invalid; please verify that Quality is a quality recognized by the system.	Verify allowed combination in eHBox_Quality
804	SendMessage/ DestinationContext	The specified Type is invalid; please verify that Type is a type recognized by the system.	Verify allowed combination in eHBox_Quality
805	SendMessage/ DestinationContext	The specified SubType is invalid; please verify that SubType is a subtype recognized by the system.	Verify allowed combination in eHBox_Quality
810	SendMessage/ BoxId	The specified BoxId is invalid; please verify the data and that you can access it.	Can you normally access that eHealthBox?
814	SendMessage/ DestinationContext	You are not authorized to publish to this Quality.	Contact the eHealth platform to get authorization.

8.2 Soap Fault Error Codes

They contain the following attributes:

Field name	Descriptions
Id	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.
Origin	The component/party causing the error: consumer or provider, client or server.
Code	The Error Code
Message	A human readable message

Retry	An optional Boolean that indicates if it is worth resending the same request.
Contact	An optional field specifying a contact description.
Environment	The eHealth environment in which the error occurs: integration, acceptation or production.


attributes
Id
Unique number identifying this message.
If present, the ticket that was created for the client's request, leading to this Error, should be placed here.
Then this id can be used to trace back the message from request to response.
Origin
The component/party that is at the cause of the error: consumer or provider, client or server.
SOAErrorType
Code
core:LocalisedStringType
attributes
xml:lang
Attempting to install the relevant ISO 2- and 3-letter codes as the enumerated possible values is probably never going to be a realistic possibility. See RFC 3066 at http://www.ietf.org/rfc/rfc3066.txt and the IANA registry at http://www.iana.org/assignments/lang-tag-apps.htm for further information.
The union allows for the 'un-declaration' of xml:lang with the empty string.
Message
1..∞
Retry
Contact
soa:Environment

8.2.1 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:

Code	Message
SOA-03006	XSD compliance failure

Example:

```
<?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>
```

8.2.2 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:

Code	Message
SOA-00001	An internal error has occurred. Please contact the Contact Center

This list can evolve.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<env: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 occured. Please contact service desk.</Message>
<soa:Environment>Production</soa:Environment>
</soa:SystemError>
</detail>
</env:Fault>
</env:Body>
</soapenv:Envelope>
```

9. Annex 1 – Publish a message to a list of professionals

It is now possible to publish a message at once to all doctors, dentists, nurses, practical nurses. This list of professionals is kept up to date on a daily basis.

However, there are some limitations of use:

- You need to have permission of the eHealth platform.
- When you send a message to a list of professionals, you may only address the message to one list (e.g. to "all nurses" and not "all nurses and all dentists"). In other words, one "SendMessageRequest" may contain only one "DestinationContext" addressed to a list of professionals (Id = "ALL").
- If needed, you can add some recipients (not containing Id = "ALL") to the same request, like in the example below.
- Please be aware that this type of request requires a lot of system resources and time to complete (up to 1 hour). Please use it with caution and moderation as it could negatively impact other messages.

Example (partial) request:

```
<DestinationContext>
    <Id>ALL</Id>
    <Type>INSS</Type>
    <Quality>NURSE</Quality>
</DestinationContext>
<DestinationContext>
    <Id>13033577799</Id>
    <Type>INSS</Type>
    <Quality>NURSE</Quality>
</DestinationContext>
<DestinationContext>
    <Id>23022211879</Id>
    <Type>INSS</Type>
    <Quality>DOCTOR</Quality>
</DestinationContext>
```