

**eh2ebox  
Cookbook  
Version 1.2**

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

Table of contents .....	2
1. Document management .....	3
1.1 Document history .....	3
2. Introduction .....	4
2.1 Goal of the service .....	4
2.2 Goal of the document .....	4
2.3 eHealth platform document references .....	4
3. Support.....	5
3.1 For issues in production .....	5
3.2 For issues in acceptance .....	5
3.3 For business issues.....	5
3.4 Certificates .....	5
4. Global overview .....	6
5. Step-by-step .....	7
5.1 Technical requirements .....	7
5.1.1 Use of the eHealth SSO solution .....	7
5.1.2 Security policies to apply .....	7
5.2 Web service.....	7
5.2.1 Method SendMessage .....	7
5.3 Technical errors.....	12

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
1.0	15/09/2017	eHealth platform	Initial
1.1	19/04/2018	eHealth platform	Lay-out and updated links
1.2	19/03/2019	eHealth platform	Correction Par 5.2.1.1 ( <b>ContentContext\CustomMeta</b> )

## 2. Introduction

### 2.1 Goal of the service

The eh2ebox is a web service (WS) allowing professionals to send messages to the Social Security eBox via the eHealthBox interface.

The eh2eBox sends messages, according to an analog message structure of the eHealthBox, and translates them so they are compatible with the Social Security eBox. More technically: the incoming request to send a message, coming from the end user, is translated at the level of the eHealth platform and sent to the parallel request on the eBox backend. Depending on the sender, this is the enterprise or citizen backend.

### 2.2 Goal of the document

This document is not a development or programming guide for internal applications. Instead, it provides functional and technical information and allows an organization to integrate and use the service of the eHealth platform.

In order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, the partners of the eHealth platform must commit to comply with the requirements of specifications, data format and release processes described in this document.

The partners must meet the technical and business requirements in order to allow the integration and validation of the eHealth platform service in the client application.

### 2.3 eHealth platform document references

You will find all the document references on the portal of the eHealth platform<sup>1</sup>. These versions or any following versions can be used for this service.

ID	Title	Version	Date	Author
1	Glossary.pdf		p.m.	eHealth platform
2	eHealth certificates		p.m.	eHealth platform
3	Secure Token Service (STS)	1.1	p.m.	eHealth platform
4	eHealthBox Publication WS	3.0	p.m.	eHealth platform

---

<sup>1</sup> <https://www.ehealth.fgov.be/ehealthplatform>

## 3. Support

### 3.1 For issues in production

eHealth platform contact center:

- Phone: 02/788 51 55
- Mail: [support@ehealth.fgov.be](mailto: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](mailto:Integration-support@ehealth.fgov.be)

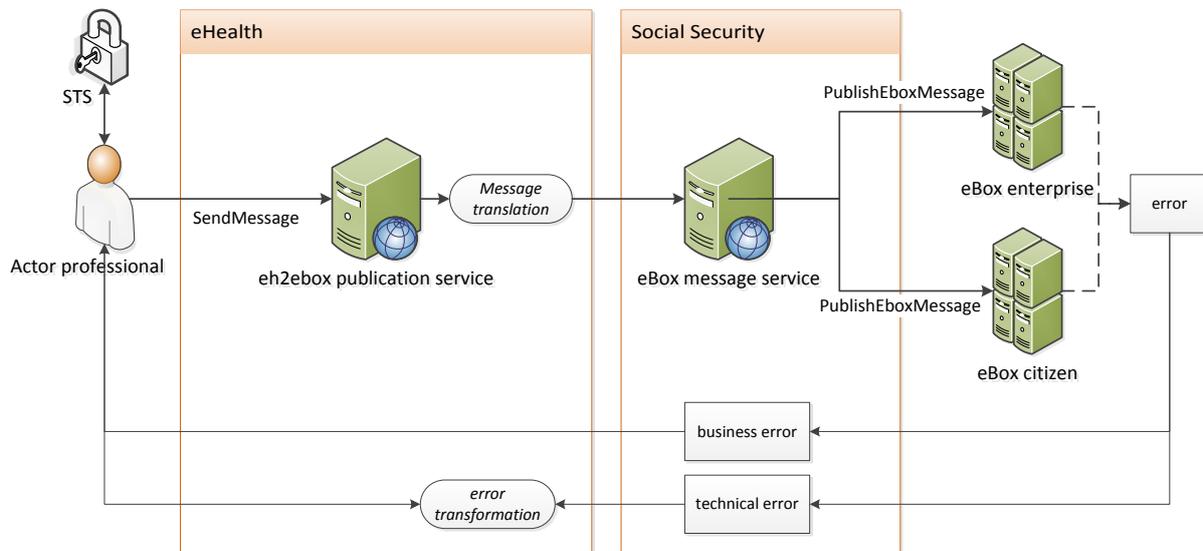
### 3.3 For business issues

- 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](mailto: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 the relevant information on the eHealth Certificates on the portal of the eHealth platform:  
<https://www.ehealth.fgov.be/ehealthplatform/nl/service-ehealth-certificaten> (Dutch)  
<https://www.ehealth.fgov.be/ehealthplatform/fr/service-certificats-ehealth> (French)
- For technical issues regarding eHealth platform certificates  
Acceptance: [acceptance-certificates@ehealth.fgov.be](mailto:acceptance-certificates@ehealth.fgov.be)  
Production: [support@ehealth.fgov.be](mailto:support@ehealth.fgov.be)

## 4. Global overview



In order to send a message to the eh2ebox, the professional needs a secure token containing the identification of the actor, which should be the same one as used for the eHealthBox. For more information, see the STS Cookbook (referred in section 2.3 with ID 3) and the cookbook eHealthBox Publication WS (referred in section 2.3. with ID 4).

The Actor Professional performs one operation: `SendMessage`. This operation, coming from the end-user, is translated at the eHealth platform and sent to the `PublishEboxMessage` operation on the eBox enterprise or eBox citizen backend.

- A business error returned by the eBox Enterprise or eBox Citizen is forwarded to the client. For more information on this error handling: see section `SendMessageResponse` 5.2.1.2.
- A technical error returned by the eBox Enterprise or eBox Citizen, is transformed at the level of the eHealth platform before sending to the client. For more information on this error handling: see section 5.3.

## 5. Step-by-step

### 5.1 Technical requirements

Valid certificate in the correct environment (see eHealthBox requirements)

Request access to the eh2ebox WS at [info@ehealth.fgov.be](mailto:info@ehealth.fgov.be).

Retrieve UDDI information: please consult the registry at the <https://www.ehealth.fgov.be/ehealthplatform>.

All the xml requests submitted to the eh2ebox WS must be encoded in the UTF-8 format.

#### 5.1.1 Use of the eHealth SSO solution

The complete overview of the profile and a systematic implementation to start protecting a new application.

For more information: please consult the cookbook Secure Token Service – HolderOfKey cookbook

[https://www.ehealth.fgov.be/ehealthplatform/Secure\\_Token\\_Service-HOK](https://www.ehealth.fgov.be/ehealthplatform/Secure_Token_Service-HOK)

#### 5.1.2 Security policies to apply

For more information about this security policy, please consult the chapter about the eHealth Certificates on the portal of the eHealth platform.

<https://www.ehealth.fgov.be/ehealthplatform/nl/service-ehealth-certificaten> (Dutch)

<https://www.ehealth.fgov.be/ehealthplatform/fr/service-certificats-ehealth> (French)

We expect that you use SSL one way for the transport layer.

As 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.

### 5.2 Web service

#### 5.2.1 Method SendMessage

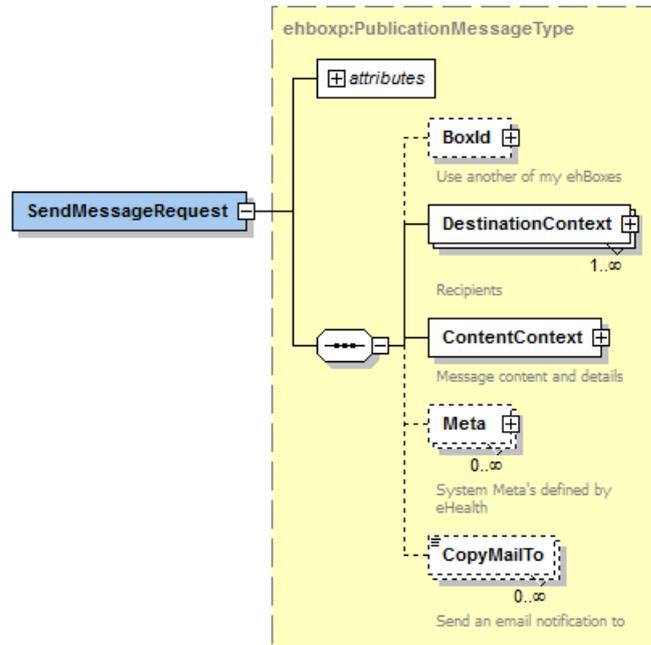
This method is identical to the SendMessage method of the eHealthBox itself. Nevertheless, there are some restrictions in order to be able to pass the message to the eBox of the social security. You will find them explained in section 5.2.1.1. Furthermore, violating the restrictions will return business errors (see section 8.1).

Therefore, the request sent in order to publish the eh2ebox content is schematically identical to the request sent in order to publish to the eHealthBox content.



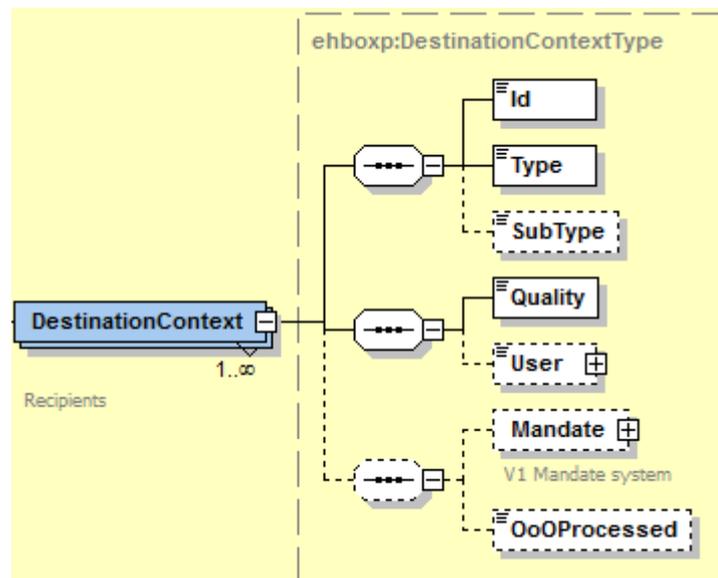
### 5.2.1.1 SendMessageRequest

The schema of SendMessageRequest is:



See section 5.3.5.1 of the cookbook eHealthBox Publication WS for detailed description of the different fields. For the eh2ebox incoming SendMessage request, following restrictions exist concerning the values of these fields:

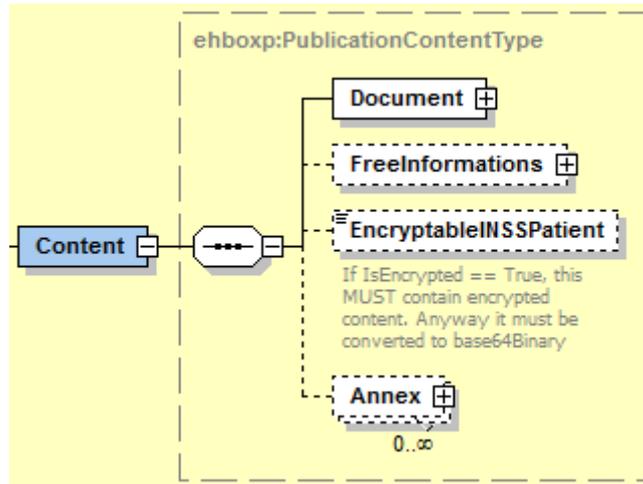
- **BoxId**  
The BoxId field is not used.
- **DestinationContext**



- ➔ For an eh2ebox request, a maximum of five DestinationContext elements can be listed one below the other.
- ➔ The Type element can only be 'CBE' or 'INSS'

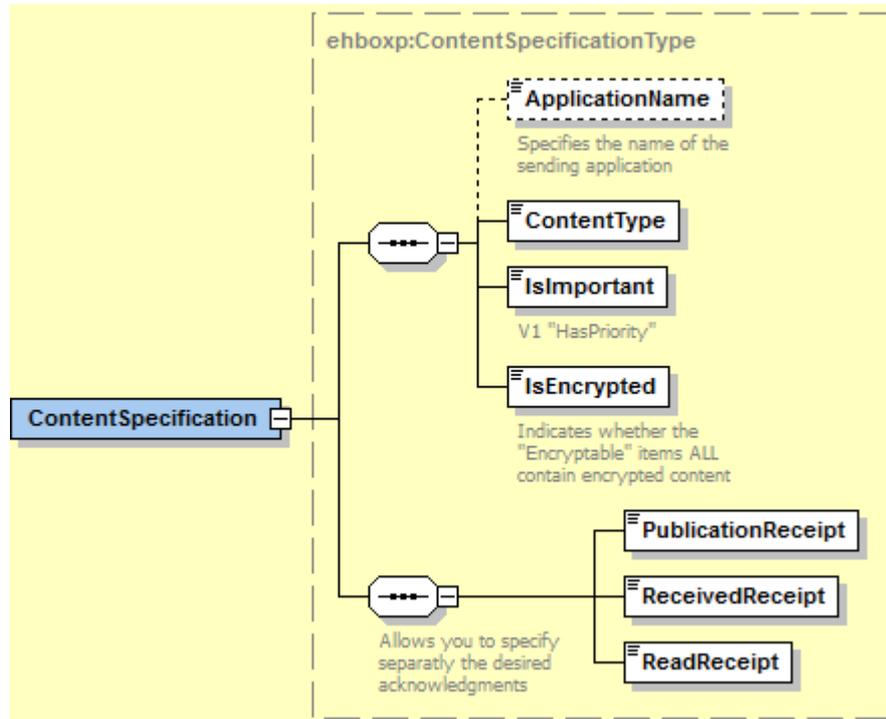
- The SubType element is not used.
- The sub-element **Quality** can only have the **value INSTITUTION or CITIZEN**. This Quality indicates if you are sending to eBox Enterprise or eBox Citizen.
- User element is not allowed.
- Mandate element is not allowed.
- OoOProcessed is not used.

- **ContentContext\Content**



- A document can never contain encrypted content, because this notion does not exist in eBox.
- A Document is immutable.
- MimeType should be one of the following: application/pdf, application/zip, text/csv, text/html, text/plain, text/xml, application/vnd.oasis.opendocument.text, application/vnd.oasis.opendocument.spreadsheet, application/vnd.oasis.opendocument.presentation, application/vnd.oasis.opendocument.graphics, application/vnd.ms-excel, application/vnd.openxmlformats-officedocument.spreadsheetml.sheet, application/vnd.ms-powerpoint, application/msword, application/vnd.openxmlformats-officedocument.wordprocessingml.document. However, only text/plain, text/xml and application/pdf are currently supported.
- The Digest definition differs slightly from the one in the eHBox. Here, the digest is the SHA-256 hash of the content (after base64 decoding for TextContent).
- FreeInformations element is not supported.
- EncryptableINSSPatient element is not supported.
- An Annex can never contain encrypted content. Same restrictions apply is in the Document element.

- **ContentContext\ContentSpecification**



- ➔ The subelement `ApplicationName` is not supported.
- ➔ The subelement `IsEncrypted` MUST be set to false.
- ➔ The subelement `ReceivedReceipt` MUST be set to false.
- ➔ The subelement `ReadReceipt` MUST be set to false.
- ➔ The subelement `PublicationReceipt` MUST be set to false.
- ➔ The subelement `IsImportant` MUST be set to false.

- **ContentContext\CustomMeta**

The only supported CustomMeta has a Key 'project' with an appropriate Value. The value should exactly match the project name for which you are using the eh2ebox. This key-value pair MUST be set. If not, you will receive a 'not authorized' error message. For more information, contact [info@ehealth.fgov.be](mailto:info@ehealth.fgov.be)

- **Meta**

The Meta element is not supported.

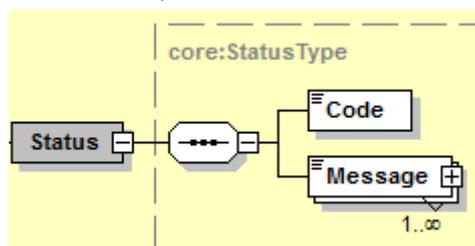
- **CopyMailTo**

A maximum of five elements is supported.

If the above restrictions are not met, a proper business error will be thrown (see section 8.1.)

### 5.2.1.2 SendMessageResponse

The `SendMessageResponse` will always contain a `SentPublicationId`, linking this response to a `SendMessageRequest`. It will never contain Recipient information.



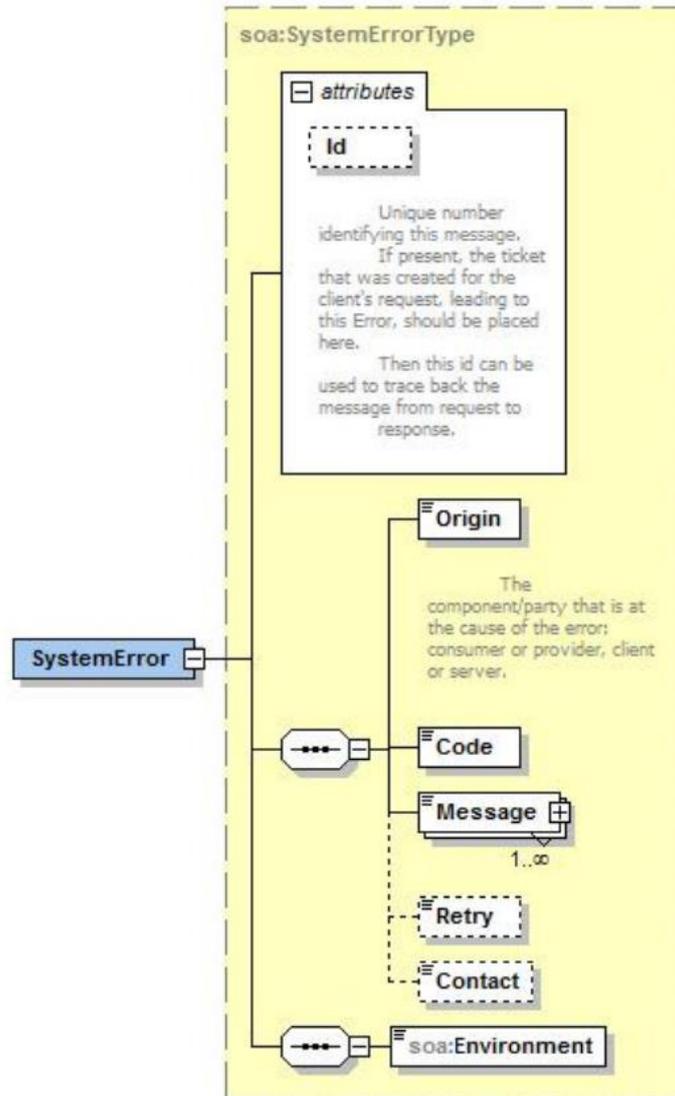
The Status element behaves as described in the cookbook eHealthBox Publication WS. Additional Codes, as listed below, have been added to this service.

### 5.2.1.2.1 Status

Code	Reason	Message returned
831	More than 5 DestinationContext elements exist.	The eBox does not support more than five Recipient elements.
832	The element DestinationContext->Quality is not INSTITUTION or CITIZEN. Furthermore <ul style="list-style-type: none"> <li>- the DestinationContext-&gt;Type has to be CBE in case of INSTITUTION as Quality</li> <li>- the DestinationContext-&gt;Type has to be INSS in case of CITIZEN as Quality</li> </ul>	The eBox only support INSTITUTION or CITIZEN (as Quality Code for the RecipientID element) with the appropriate Type.
833	The element DestinationContext->User exists	The eBox Recipient does not support a User element.
834	The element DestinationContext->Mandate exists	The eBox Recipient does not support a Mandate element.
835	The element ContentContext->Content->FreeInformations exists	The eBox does not support a FreeInformations element.
836	The element ContentContext->Content->EncryptableINSSPatient exists	The eBox does not support an EncryptableINSSPatient element.
837	More than five ContentContext->Content->Annex elements exist	The eBox does not support more than five Annex elements.
838	The element ContentContext->ContentSpecification->IsEncrypted is set to true	The eBox ContentSpecification does not support encryption.
839	The element ContentContext->ContentSpecification->ReceivedReceipt exists	The eBox ContentSpecification does not support a ReceivedReceipt element.
840	The element ContentContext->ContentSpecification->ReadReceipt exists	The eBox ContentSpecification does not support a ReadReceipt element.
841	The element Meta exists	The eBox does not support a Meta element
842	More than five CopyMailTo elements exist	The eBox does not support more than five EmailAddresses elements.
843	The element ContentContext->ContentSpecification->PublicationReceipt exists	The eBox does not support a PublicationReceipt element.
844	The element ContentContext->ContentSpecification->IsImportant is set to true	The eBox ContentSpecification does not support flagging a message as important.
845	The element ContentContext->ContentSpecification->ContentType is not 'DOCUMENT'	The eBox only supports DOCUMENT in the ContentType element.

### 5.3 Technical errors

Technical errors are errors inherent to the internal working of a WS. They are returned as SOAP Faults. The SOA Standard for Error handling specifies a structure for SystemError and BusinessError, thrown as SOAP Faults. A SystemError MUST be thrown when a system failure occurred. It is not related to the business of the service. The SOA system error structure is as follows:



The SystemError element contains a unique Id attribute for tracing. The origin is set to server or provider. Retry is set to true if the user can try again immediately without interventions.

**Example:**

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode>soapenv:Server</faultcode>
      <faultstring>SOA-02002</faultstring>
      <detail>
        <soa:SystemError Id="9E0-00000P1-00-C" xmlns:soa="urn:be:fgov:ehhealth:errors:soa:v1">
          <Origin>Server</Origin>
          <Code>SOA-02002</Code>
          <Message xml:lang="en">Service is temporarily not available. Please contact service
desk.</Message>
          <Retry>true</Retry>
          <soa:Environment>Test</soa:Environment>
        </soa:SystemError>
      </detail>
    </S:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```

The SOAP Fault element has the following sub elements:

Element name	Description	Required
faultcode	A code for identifying the fault	Yes
faultstring	A human readable explanation of the fault	Yes
Faultactor	Information about who caused the fault to happen (the origin)	No
detail	Holds application specific error information related to the Body element. E.g., it could include a java stack trace or any other kind of trace, used internally, to document on the cause of this error.	No

The default SOAP fault code values are defined in an extensible manner allowing new SOAP fault code values to be defined while maintaining backwards compatibility with existing fault code values.

Element name	Description
versionMismatch	Found an invalid namespace for the SOAP Envelope element.
mustUnderstand	An immediate Child element of the Header element, with the mustUnderstand attribute set to "1", was not understood.
Client	Message incorrectly formed or containing incorrect information.
Server	There was a problem with the server so the message could not proceed.

Description of the possible SOAP fault exceptions:

<	Component	Description	Solution/Explanation
SOA-00001	Undefined	Service error	This is the default error sent to the user in case further details are unknown.
SOA-01001	Consumer	Service call not authenticated	From the security information provided <ul style="list-style-type: none"> <li>• or the user could not be identified</li> <li>• or the credentials provided are not correct</li> </ul>
SOA-01002	Consumer	Service call not authorized	The user is identified and authenticated but is not allowed to call the given service.



SOA-02001	Provider	Service not available. Please contact service desk	<ul style="list-style-type: none"> <li>• An unexpected error has occurred</li> <li>• Retries will not work</li> <li>• Service desk may help with root cause analysis</li> </ul>
SOA-02002	Provider	Service temporarily not available. Please try later	<ul style="list-style-type: none"> <li>• An unexpected error has occurred</li> <li>• Retries should work</li> <li>• If the problem persists service desk may help</li> </ul>
SOA-03001	Consumer	Malformed message	This is default error for content related errors in case more details are lacking.
SOA-03002	Consumer	Message must be SOAP	Message does not respect the SOAP standard
SOA-03003	Consumer	Message must contain SOAP body	Message respects the SOAP standard, but body is missing
SOA-03004	Consumer	WS-I compliance failure	Message does not respect the WS-I standard
SOA-03005	Consumer	WSDL compliance failure	Message is not compliant with WSDL in Registry/Repository
SOA-03006	Consumer	XSD compliance failure	Message is not compliant with XSD in Registry/Repository
SOA-03007	Consumer	Message content validation failure	<p>From the message content (conform XSD):</p> <ul style="list-style-type: none"> <li>• Extended checks on the element format failed</li> <li>• Cross-checks between fields failed</li> </ul>