Secure Token Service
Holder-of-Key profile
Cookbook
Version 1.6

eHealth platform

Willebroekkaai 38 – 1000 Brussel

38, Quai de Willebroek – 1000 Bruxelles

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
1.0	31/08/2010	eHealth platform	Initial version
1.1	31/08/2017	eHealth platform	Update information related to Self-Signed certificate
1.2	13/04/2018	eHealth platform	Updated links
1.3	18/07/2018	eHealth platform	BCP guidelines
1.4	24/02/2021	eHealth platform	Correction references
1.5	13/07/2022	eHealth platform	§ 3.2 Status (added) § 5.2 Tracing (updated)
1.6	25/01/2023	eHealth platform	Remove support for SHA-1

2. Introduction

2.1 Goal of the service

The goal of this service is to offer a web service based single-sign-on solution (SSO) for the health care sector. The health care party, a web service consumer (WSC), contacts this service to obtain a session ticket (SAML token), which can be used to invoke the services offered by a web service provider (WSP).

2.2 Goal of the document

This document is not a development or a programming guide for internal applications: the partners of the eHealth platform always keep a total freedom within those fields. Nevertheless, in order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, they must commit to comply with specifications, data format, and release processes described within this document. In addition, our partner 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 eHealth platform document references

On the portal of the eHealth platform, you can find all the referenced documents.[1]. Partners can use these versions or any following versions for services of the eHealth platform.

ID	Title	Version	Date	Author
1.	Glossary.pdf	1.0	01/01/2010	eHealth platform
2.	STS – Annex: Mapping Certificate holder	1.0	20/01/2023	eHealth platform

2.4 External document references

ID	Title	Source	Date	Author
1.	*SAML*	***http://www.oasis-open.org/specs/index.php#samlv1.1***	2010-08-30	
2.	*SAMLTokenProfile*	***http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-pr-SAMLTokenProfile-01.html***	2010-08-30	
3.	Basic Profile Version 1.1	***http://www.ws-i.org/Profiles/BasicProfile-1.1-2004-08-24.html***	24/08/2004	Web Services Interoperability Organization

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

3. Support

3.1 Helpdesk eHealth platform

3.1.1 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 chapter about the eHealth Certificates on the portal of the eHealth platform

- ***https://www.ehealth.fgov.be/ehealthplatform/nl/ehealth-certificaten***
- ***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***

3.1.2 For issues in production

eHealth platform contact centre:

- Phone: 02 788 51 55 (on working days from 7 am till 8 pm)
- 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)

3.1.3 For issues in acceptance

Integration-support@ehealth.fgov.be

3.1.4 For business issues

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

3.2 Status

The website ***https://status.ehealth.fgov.be*** is the monitoring and information tool for the ICT functioning of the eHealth services that are partners of the Belgian eHealth system.

3.3 Business continuity plan

In order to limit impacts if serious incidents occur on eHealth components, we recommend the integrator to follow these instructions for I.AM STS:

1. Apply the sliding window principle
2. Persist current SAML token on disk

A SAML token is valid for at most 24 hours. During this period, the SAML token can be reused multiple times and it is not required for the WSC to ask for a new one. If I.AM STS cannot provide new SAML token, the WSC should still be able to reach its usual services while the SAML token is valid.

If the health care professional device restarts, the SAML token previously obtained must be reused.

The sliding window principle can be applied only with a valid SAML token. In this situation, you can avoid some short troubles on I.AM STS.

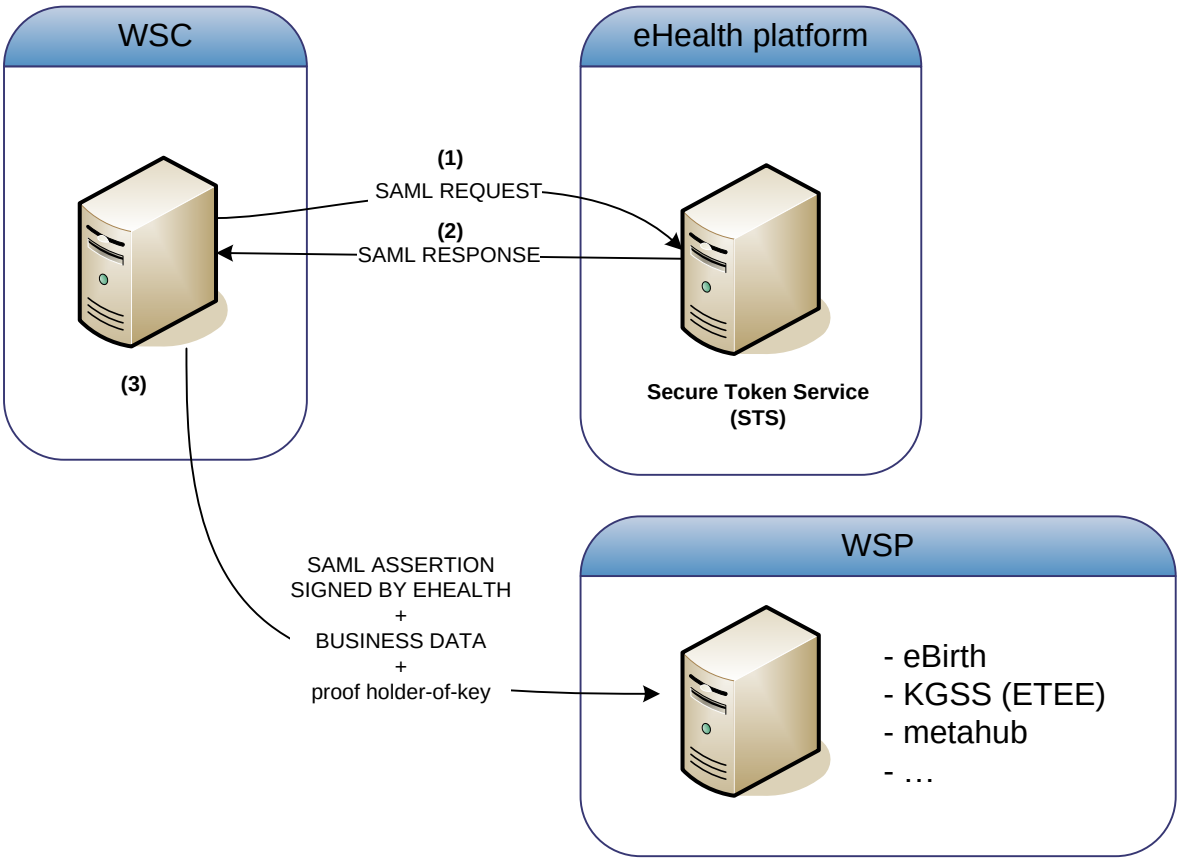
If the WSC has a valid SAML token:

- The WSC must request a new token after a given period (= X/2).
- If I.AM STS delivers a new SAML token, this new SAML token will be valid for X hours from that time.
- If I.AM STS cannot deliver a new SAML token, the WSC must try to get a new one after X/4 hours
- …

The sliding window principle involves multiple eID pin-code introduction within the sliding window.

4. Global overview

Every health care party can contact this service to obtain a session ticket (SAML token) and use this token to communicate with services that are accessible 'through' the eHealth-platform. Only web service based solutions are supported by this service.



Step 1: A web service consumer (WSC) requests a SAML token from the STS. The message contains two parts: identification information relative on the WSC and information to be confirmed by eHealth.

Step 2: eHealth sends a signed SAML assertion to the WSC with the message containing the confirmed information.

Step 3 (out of scope):

From now on, the WSC can use the obtained SAML token for further communication with the different WSP's. When the SAML assertion is invalid, the WSC must request a new SAML token (Step1)

For example, a general practitioner (GP) wants to use two different applications that every GP can access. The end user requests a SAML token to obtain proof of being a GP. The eHealth platform validates this claim against its validated Authentic sources (VAS). In case of a positive response, the STS sends a session ticket to the requestor . This session ticket contains the proof that he is a GP.

Every session ticket has a lifetime: when a session has expired, the end user must request a new one. When the GP contacts a target application, he must send the session ticket along with the business data. Due to the session ticket, the applications have certified proof that the requestor is a GP and can conduct his business.

5. Step-by-step

This web service is expects xml messages and returns xml messages.

5.1 WS-I Basic Profile 1.1

Your request must be WS-I compliant (See Chap 2.4 - External Document Ref).

5.2 Tracing

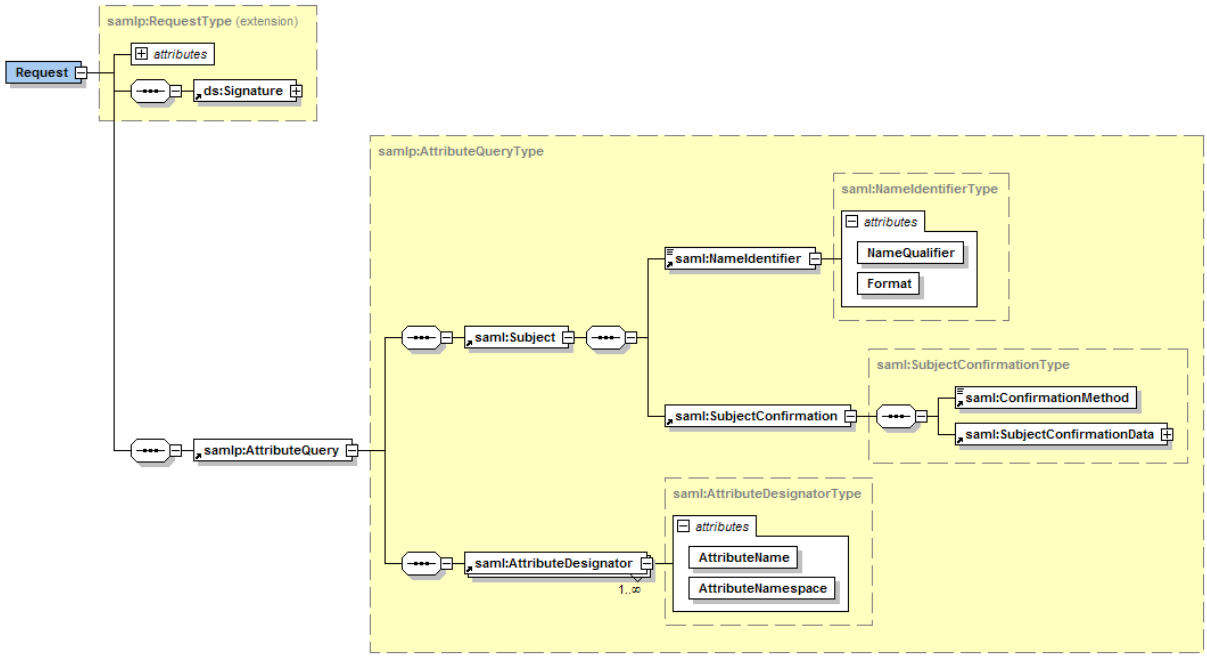
To use this service, the request SHOULD contain the following two http header values (see RFC ***https://datatracker.ietf.org/doc/html/rfc7231#section-5.5.3***):

1. User-Agent: information identifying the software product and underlying technical stack/platform. It MUST include the minimal identification information of the software such that the emergency contact (see below) can uniquely identify the component.
   a. Pattern: {minimal software information}/{version} {minimal connector information}/{connector-package-version}
   b. Regular expression for each subset (separated by a space) of the pattern: [[a-zA-Z0-9-\/]*\/[0-9azA-Z-_.]*
   c. Examples:
      User-Agent: myProduct/62.310.4 Technical/3.19.0
      User-Agent: Topaz-XXXX/123.23.X freeconnector/XXXXX.XXX
2. **From:** email-address that can be used for emergency contact in case of an operational problem
   Examples:
   From: ***info@mycompany.be***

5.3 Formulating a request

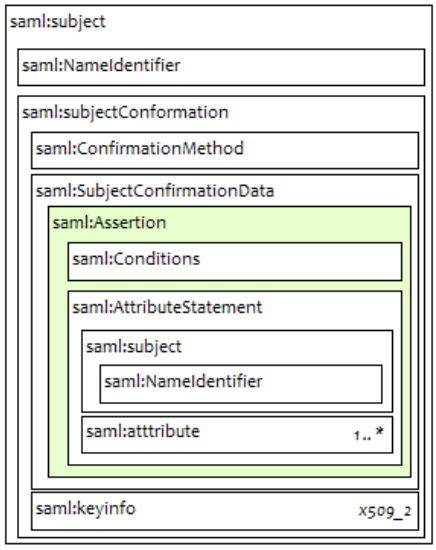
The request of the STS is based on the SAML 1.1 protocol. The full specifications of this protocol can be found on: ***http://www.oasis-open.org/specs/index.php#samlv1.1***. In this document, we will only describe the accepted messages.

We are using the <*AttributeQuery*> element as input for the STS, because the goal is the return of the requested attributes for a given subject. A successful response will be in the form of assertions containing attribute statements. A request of a SAML token from the STS looks as follows:



An <AttributeQuery> messages consists of two main parts. . A successful response will be in the form of assertions containing attribute statements WSC, the saml:subject. The second part of the message contains all of the information, which the STS will confirm in its answer. The following chapters will describe each part.

5.3.1 Part 1: Information on the calling Partner (WSC)



The first part of an <AttributeQuery> message, saml:subject, contains data about the calling partner (WSC).

The <*NameIdentifier*> element specifies a subject as a combination of a name qualifier, a name, and a format. The name is provided as element content. The subject in the STS is the certificate that is used to identify the caller.

The <*SubjectConfirmation*> element specifies a subject by supplying additional data that allows the subject to be authenticated.

Field name	Descriptions
NameIdentifier	The <*NameIdentifier*> element has the following attributes: ➢ NameQualifier: The security or administrative domain that qualifies the name of the subject. ➢ Format: An URI reference representing the format in which the <*NameIdentifier*> information is provided. Recommended URI: urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName In the STS, the distinguished names of the Certificate Authority of the identification certificate is used as NameQualifier and the distinguished names of this certificate as value of the NameIdentifier. We recommends the use of RFC1779 or RFC2253 to format the distinguished names. example: <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName" NameQualifier="{subject x509_1_CA}"> {subject x509_1} </NameIdentifier>
SubjectConfirmation	It contains the following elements in order: ➢ ConfirmationMethod: A URI reference that identifies a protocol to be used to authenticate the subject. In our case only urn:oasis:names:tc:SAML:1.0:cm:holder-of-key is supported. ➢ SubjectConfirmationData: A SAML assertion issued by the token requestor containing additional identification information, which the STS uses in its verification process.

The SubjectConfirmationData contains a SAML assertion, which is a standard way of supplying additional identification data. This self-generated assertion contains information that helps the STS to make a decision. The STS will verify the links between the different attributes. When the link verification failed, no SAML token will be delivered.

Field name	Descriptions
NameIdentifier	The NameIdentifier of the AttributeQuery is repeated.
Conditions	The WSC can limit the use of the SAML assertion in time. When he wants to use this functionality, he must specify the attributes NotBefore and NotOnOrAfter. The *NotBefore* and *NotOnOrAfter* attributes specify time limits on the validity of the assertion. The *NotBefore* attribute specifies the time instant at which the validity interval begins. The *NotOnOrAfter* attribute specifies the time instant at which the validity interval has ended. The STS will use these attributes to determine the validity of the delivered token. The eHealth platform defines for each attribute the maximum validity, the WSC can only shorten it.
AttributeDesignator	The <*AttributeDesignator*> element identifies an attribute name within an attribute namespace. The <*AttributeDesignator*> element contains the following XML attributes: ➢ AttributeNamespace: The namespace in which the AttributeName elements are interpreted. ➢ AttributeName: The name of the attribute.

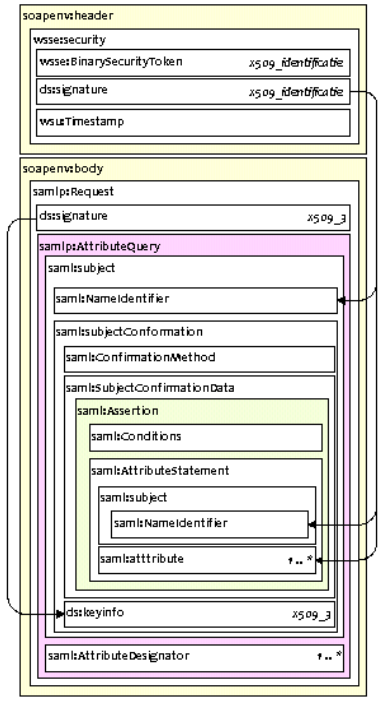
	For every item in this list a definition of the attribute is provided. For every token that has to be delivered for a WSP the required identification attributes are also published.

5.3.2 Part 2: Information on the attributes

The second part of the message contains information on the attributes, which the eHealth platform has to confirm.

Field name	Descriptions
AttributeDesignator	The <*AttributeDesignator*> element identifies an attribute name within an attribute namespace. The <*AttributeDesignator*> element contains the following XML attributes: ➢ AttributeNamespace: The namespace in which the AttributeName elements are interpreted. ➢ AttributeName: The name of the attribute. For every item in this list, a definition of the attribute is provided and the required identification attributes in order to process this attribute correctly.

5.3.3 Information on multiple signatures



The request contains several signatures and references of certificates.

There is a certificate that the requestor uses to identify himself towards the STS. In the schema on the left, this certificate is called X509_1. The contents of this certificate is repeated in every <NameIdentifier> and <Issuer> of the SAML Attribute Query. In the self-generated SAML assertion, an attribute of the type certificateHolder must be present.

With the general availability of eHealth Certificates, self-signed certificates are no longer allowed.

An eID only solution is still possible however, it is allowed to sign the SAML Request using your eID.

The eHealth Connector offers this functionality using the **createSessionEidOnly**() method in the

be.ehealth.technicalconnector.session.impl.SessionManagerImpl class.

In this case, eHealth certificates are not necessary.

More information on the Holder-of-Key (HOK) profile can be found: ***http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-pr-SAMLTokenProfile-01.html***

Field name	Descriptions
saml:NameIdentifier	Every NameIdentifier element in the SAML AttributeQuery will follow the same guidelines. The STS expects an X509SubjectName as NameIdentifier. The content is

	specified in the XML Signature Recommendation [xmldsig]. <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName" NameQualifier="CA_x509_1">x509_1</NameIdentifier>
saml:Attribute	In the self-generated SAML assertion, an identification attribute of the certificate holder "must" be declared. For every certificate template, one corresponding certificate holder attribute is defined.
ds:KeyInfo	The contents of this element will be used by the STS to validate/verify the enveloped signature of the SAML Attribute Query. The content of this element is defined in the XML Signature Recommendation [xmldsig]. At this time, only a x509v3 certificate is supported, information will be found in the <ds:X509Data> section. The STS expects a base64 representation of the certificate in the <ds:X509Certificate> section, the other possibilities aren't supported. <ds:KeyInfo> <ds:X509Data><ds:X509Certificate>base64 X509_3</ds:X509Certificate></ds:X509Data> </ds:KeyInfo>

5.4 Interpretation of the Reply

The STS uses the HOK profile in order to achieve its goal. This mechanism protects messages with signed SAML assertion (issued by a trusted authority, in this case the eHealth platform) carrying the client public key and authorization information with integrity and confidentiality protection using mutual certificates.

The HOK method establishes the correspondence between a SOAP message and the SAML assertions added to the SOAP message. The attesting entity includes a signature that can be verified with the key information in the confirmation method of the subject statements of the SAML assertion referenced for key info for the signature. More information about the Holder-of-key method can be found in this document ***http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-pr-SAMLTokenProfile-01.html***

Under this scenario, the WSP does not trust the client directly, but requires the client to send a SAML assertion issued by the eHealth platform. The client knows the recipient's public key, but does not share a direct trust relationship with the recipient. The recipient has a trust relationship with the authority issuing the SAML token. The request is signed with the client's private key and encrypted with the server certificate. The response is signed using the server's private key and encrypted using the key provided within the HOK SAML assertion.

The response of the STS contains a *samlAssertion* with an *AuthenticationStatement* and an *AttributeStatement*. In the *AuthenticationStatement* the subject of the request is repeated. In the *AttributeStatement* an answer to the requested attributes is returned. The complete assertion is signed by the eHealth platform. When an unexpected error occurs while handling the request, no SAML assertion is delivered.

The SAML assertion "container" itself contains the following information

- Issuing information: who issued the assertion, when was it issued and the assertion unique identifier.

```
AssertionID="f887b8101ff23afd3508b9a43cf73cc7"
IssueInstant="2010-03-09T10:55:27.366Z"
Issuer="urn:be:fgov:ehealth:sts:1_0"
MajorVersion="1"
MinorVersion="1"
```

- Conditions information: validity, audience restriction…. The STS will only use the period information. At the time, the other possibilities will not be used.

```
<Conditions
    NotBefore="2010-03-09T10:55:27.366Z"
    NotOnOrAfter="2010-03-09T11:55:27.366Z"/>
```

- AuthenticationStatement: The STS asserts that the subject was authenticated by certain means at a certain time. In our case, the *AuthenticationMethod* will be x509-PKI because the STS is protected by

an x509v3 certificate

```
<AuthenticationStatement
    AuthenticationInstant="2010-03-09T10:55:27.366Z"
    AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X509-PKI">
    <Subject>...</Subject>
</AuthenticationStatement>
```

AttributeStatement: The STS asserts that the given subject with the requested attributes. Only the *NameIdentifier* of the subject is repeated. For every requested attribute an <Attribute> is present. Without value for an attribute, an empty <Attribute> tag is returned.

For example when an end user wants the proof that he possesses two attributes. However, when the STS validates those two claims by its VAS, the conclusion is that he possesses only one attribute. An empty tag will be returned for the other attribute (or 'false' value for boolean attributes).

```
<Attribute
    AttributeName="attribute:a"
    AttributeNamespace="urn:be:fgov:certified-namespace:ehealth">
    <AttributeValue>12345625000</AttributeValue>
</Attribute>
```

```
<Attribute
    AttributeName="attribute:b"
    AttributeNamespace="urn:be:fgov:certified-namespace:ehealth">
    <AttributeValue/>
</Attribute>
```

5.5 Examples

A person, Alice SPECIMEN, with identification number '71715100070' wants to obtain a SAML token proving she is a midwife. With this delivered token, she should have access to the services accessible for midwifes.

X509 v3 Identification certificate

```
SubjectDN:
          C=BE,
          CN=Alice SPECIMEN(Signature)
          SURNAME=SPECIMEN,
        GIVENNAME=Alice Geldigekaart3064
          SERIALNUMBER=71715100070
IssuerDN:
          C=BE,
          CN=SPECIMEN Citizen CA
```

5.5.1 The request

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Header>
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
soapenv:mustUnderstand="1">
      <wsse:BinarySecurityToken
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
        ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
        wsu:Id="CertId-A94241D7943366FABA12683960149041">{eID certificate}</wsse:BinarySecurityToken>
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="Signature-2">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
          <ds:Reference URI="#Timestamp-1">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <ds:DigestValue>{digest message part}</ds:DigestValue>
```

```
                    </ds:Reference>
                    <ds:Reference URI="#CertId-A94241D7943366FABA12683960149041">
                      <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                      </ds:Transforms>
                      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                      <ds:DigestValue>{digest message part}</ds:DigestValue>
                    </ds:Reference>
                    <ds:Reference URI="#id-3">
                      <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                      </ds:Transforms>
                      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                      <ds:DigestValue>{digest message part}</ds:DigestValue>
                    </ds:Reference>
                  </ds:SignedInfo>
                  <ds:SignatureValue>{signature with eID}</ds:SignatureValue>
                  <ds:KeyInfo Id="KeyId-A94241D7943366FABA12683960149202">
                    <wsse:SecurityTokenReference xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" wsu:Id="STRId-A94241D7943366FABA12683960149363">
                      <wsse:Reference URI="#CertId-A94241D7943366FABA12683960149041"
                      ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
                    </wsse:SecurityTokenReference>
                  </ds:KeyInfo>
                </ds:Signature>
                <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="Timestamp-1">
                  <wsu:Created>2010-03-12T12:13:34.858Z</wsu:Created>
                  <wsu:Expires>2010-03-12T12:18:34.858Z</wsu:Expires>
                </wsu:Timestamp>
              </wsse:Security>
            </soapenv:Header>
            <soapenv:Body xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="id-
3">
              <Request xmlns="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" IssueInstant="2010-03-12T12:13:18.482Z" MajorVersion="1"
MinorVersion="1" RequestID="_81d275d281c4e93a225a7e6d5901d46f">
                <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                  <ds:SignedInfo>
                    <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                    <ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                    <ds:Reference URI="#_81d275d281c4e93a225a7e6d5901d46f">
                      <ds:Transforms>
                        <ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
                        <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                          <ec:InclusiveNamespaces xmlns:ec="http://www.w3.org/2001/10/xml-exc-c14n#"
                          PrefixList="code ds kind rw saml samlp typens #default xsd xsi"/>
                        </ds:Transform>
                      </ds:Transforms>
                      <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                      <ds:DigestValue>{digest message part}</ds:DigestValue>
                    </ds:Reference>
                  </ds:SignedInfo>
```

```
                    <ds:SignatureValue>{signature with proof Holder-Of-Key certificate}</ds:SignatureValue>
                    <ds:KeyInfo>
                      <ds:X509Data>
                        <ds:X509Certificate>{certificate proof Holder-Of-Key}</ds:X509Certificate>
                      </ds:X509Data>
                    </ds:KeyInfo>
                  </ds:Signature>
                  <AttributeQuery>
                    <Subject xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
                      <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName" NameQualifier="C=BE,
CN=SPECIMEN Citizen CA">C=BE, CN=Alice SPECIMEN(Signature), SURNAME=SPECIMEN, GIVENNAME=Alice Geldigekaart3064,
SERIALNUMBER=71715100070</NameIdentifier>
                      <SubjectConfirmation>
                        <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:holder-of-key</ConfirmationMethod>
                        <SubjectConfirmationData>
                          <Assertion AssertionID="_1a8b2689c8a1edd1057a651074783a37" IssueInstant="2010-03-12T12:13:18.029Z"
Issuer="C=BE, CN=Alice SPECIMEN(Signature), SURNAME=SPECIMEN, GIVENNAME=Alice Geldigekaart3064,
SERIALNUMBER=71715100070" MajorVersion="1" MinorVersion="1">
                            <Conditions NotBefore="2010-03-12T12:13:18.029Z" NotOnOrAfter="2010-03-12T13:13:18.029Z"/>
                            <AttributeStatement>
                              <Subject>
                                <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName"
NameQualifier="C=BE, CN=SPECIMEN Citizen CA">C=BE, CN=Alice SPECIMEN(Signature), SURNAME=SPECIMEN, GIVENNAME=Alice
Geldigekaart3064, SERIALNUMBER=71715100070</NameIdentifier>
                              </Subject>
                              <Attribute AttributeName="urn:be:fgov:person:ssin" AttributeNamespace="urn:be:fgov:identification-
namespace">
                                <AttributeValue>71715100070</AttributeValue>
                              </Attribute>
                              <Attribute AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:person:ssin"
AttributeNamespace="urn:be:fgov:identification-namespace">
                                <AttributeValue>71715100070</AttributeValue>
                              </Attribute>
                            </AttributeStatement>
                          </Assertion>
                        </SubjectConfirmationData>
                        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                          <ds:X509Data>
                            <ds:X509Certificate>{certificate proof Holder-Of-Key}</ds:X509Certificate>
                          </ds:X509Data>
                        </ds:KeyInfo>
                      </SubjectConfirmation>
                    </Subject>
                    <AttributeDesignator xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
                      AttributeName="urn:be:fgov:person:ssin"
                      AttributeNamespace="urn:be:fgov:identification-namespace"/>
                    <AttributeDesignator xmlns="urn:oasis:names:tc:SAML:1.0:assertion"
                      AttributeName="urn:be:fgov:person:ssin:midwife:boolean"
                      AttributeNamespace="urn:be:fgov:certified-namespace:ehealth"/>
                  </AttributeQuery>
                </Request>
              </soapenv:Body>
            </soapenv:Envelope>
```

5.5.2 The response

```
<?xml version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <Response xmlns="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" InResponseTo="_313d48d693f8ae8691dd2f6643ed654a" IssueInstant="2010-03-
12T12:19:04.785Z" MajorVersion="1" MinorVersion="1" Recipient="urn:be:fgov:ehealth:ssoclient"
ResponseID="ab2ae0347a94ba9e8211b5aa09928c82">
      <Status>
        <StatusCode Value="samlp:Success"/>
      </Status>
      <Assertion xmlns="urn:oasis:names:tc:SAML:1.0:assertion" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" AssertionID="f887b8101ff23afd3508b9a43cf73cc7" IssueInstant="2010-03-
09T10:55:27.366Z" Issuer="urn:be:fgov:ehealth:sts:1_0" MajorVersion="1" MinorVersion="1">
        <Conditions NotBefore="2010-03-09T10:55:27.366Z" NotOnOrAfter="2010-03-09T11:55:27.366Z"/>
        <AuthenticationStatement AuthenticationInstant="2010-03-09T10:55:27.366Z"
AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X509-PKI">
          <Subject>
            <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName" NameQualifier="C=BE, CN=SPECIMEN
Citizen CA">C=BE, CN=Alice SPECIMEN(Signature), SURNAME=SPECIMEN, GIVENNAME=Alice Geldigekaart3064,
SERIALNUMBER=71715100070</NameIdentifier>
            <SubjectConfirmation>
              <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:holder-of-key</ConfirmationMethod>
              <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                <ds:X509Data>
                  <ds:X509Certificate>{certificate proof Holder-Of-Key}</ds:X509Certificate>
                </ds:X509Data>
              </ds:KeyInfo>
            </SubjectConfirmation>
          </Subject>
        </AuthenticationStatement>
        <AttributeStatement>
          <Subject>
            <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName" NameQualifier="C=BE, CN=SPECIMEN
Citizen CA">C=BE, CN=Alice SPECIMEN(Signature), SURNAME=SPECIMEN, GIVENNAME=Alice Geldigekaart3064,
SERIALNUMBER=71715100070</NameIdentifier>
          </Subject>
          <Attribute AttributeName="urn:be:fgov:person:ssin" AttributeNamespace="urn:be:fgov:identification-namespace">
            <AttributeValue>71715100070</AttributeValue>
          </Attribute>
          <Attribute AttributeName="urn:be:fgov:person:ssin:midwife:boolean" AttributeNamespace="urn:be:fgov:certified-
namespace:ehealth">
            <AttributeValue>true</AttributeValue>
          </Attribute>
        </AttributeStatement>
        <dsig:Signature xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
          <dsig:SignedInfo>
            <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <exc14n:InclusiveNamespaces xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </dsig:CanonicalizationMethod>
            <dsig:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
            <dsig:Reference URI="#f887b8101ff23afd3508b9a43cf73cc7">
              <dsig:Transforms>
                <dsig:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
```

```
                <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                  <exc14n:InclusiveNamespaces xmlns:exc14n="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </dsig:Transform>
              </dsig:Transforms>
              <dsig:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
              <dsig:DigestValue>{digest assertion}</dsig:DigestValue>
            </dsig:Reference>
          </dsig:SignedInfo>
          <dsig:SignatureValue>{signature ehealth}</dsig:SignatureValue>
          <dsig:KeyInfo>
            <dsig:X509Data>
              <dsig:X509Certificate>{certificate ehealth}</dsig:X509Certificate>
            </dsig:X509Data>
          </dsig:KeyInfo>
        </dsig:Signature>
      </Assertion>
    </Response>
  </S:Body>
</S:Envelope>
```

6. Risks and security

6.1 Risks & safety

6.2 Security

6.2.1 Business security

In case the development adds an additional use case based on an existing integration, the partner should inform the eHealth platform at least one month in advance with a detailed estimate of the expected load in order to be able to 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 within 10 business days with the newest version of the software.

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

6.2.2 Web service

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

- SSL one way
- Time-to-live of the message: one minute.
- 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.
- No encryption on the message.

7. Test and release

7.1 Initiation of the procedure

If you intend to use a service from the eHealth platform, please contact ***info@ehealth.fgov.be*** and provide them with detailed information on your project. The project department will inform you about the next steps.

7.2 Development and test procedure

You have to develop a client in order to connect to our WS. You will find most of the info on how to integrate published in the technical library on the portal of the eHealth platform.

Upon request, the eHealth platform can provide you with test cases so you can test your client before the release in the acceptance environment.

7.3 Release procedure

When development tests are successful, you can request to access the acceptance environment of the eHealth platform. From this moment, you start the integration and acceptance tests. The eHealth platform suggests testing during minimum one month.

After successful acceptance tests, the partner mails the test and performance results with a sample of the "eHealth request" and the "eHealth answer" 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 performance tests.

For further information and instructions, please contact: ***integration-support@ehealth.fgov.be***.

7.4 Operational follow-up

Once in production, the partner using the eHealth platform service for one of his applications will always 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.

8. Error and failure messages

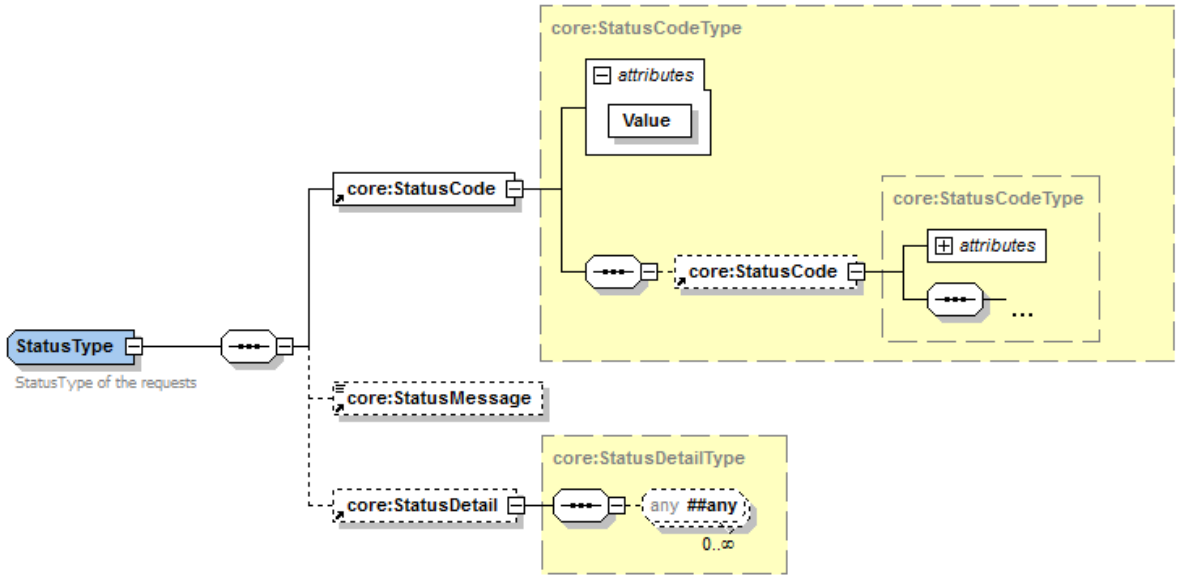
This list of error codes originating from the eHealth platform can be found below. This list is not exhaustive.

In the case of a technical error, a SOAP fault exception is returned (see table below).

Table 1: Description of the possible SOAP fault exceptions.

Error code	Component	Description	Solution/Explanation
SOA-00001	Not determined	Service error	This is the default error sent to the consumer in case more details are missing.
SOA-01001	Consumer	Service call not authenticated	From the security information provided: • or the consumer could not be identified. • or the credentials provided are not correct.
SOA-01002	Consumer	Service call not authorized	The consumer is identified and authenticated but is not allowed to call the given service.
SOA-02001	Provider	Service not available. Please contact service desk	An unexpected error has occurred. • Retries will not work. • Service desk may help with root cause analysis.
SOA-02002	Provider	Service temporarily not available. Please try later	An unexpected error has occurred. • Retries should work. • If the problem persists service desk may help.
SOA-03001	Consumer	Malformed message	This is the default error for content related errors in case more details are missing.
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	From the message content (conform XSD): • Extended checks on the element format failed. • Cross-checks between fields failed.

If there are no technical errors, response will contain a Status element with the following structure :



Status is used to indicate the status of the completion of the request. The status is represented by a StatusCode and optionally the StatusMessage describing the status. Additional StatusDetail gives extra information on the encountered business errors returned by the target service.

The possible values for the Level 1 StatusCode are:

- *urn:be:fgov:ehealth:2.0:status:Success* (everything OK)
- *urn:be:fgov:ehealth:2.0:status:Requester* (error caused by client (consumer))
- *urn:be:fgov:ehealth:2.0:status:Responder* (error caused by provider)

Level 2 StatusCode and StatusMessage is not used.