Secure Token Service
WS Trust
Cookbook
Version 1.0

eHealth platform

Willebroekkaai 38 – 1000 Brussel

38, Quai de Willebroek – 1000 Bruxelles

Table of contents

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	20/01/2023	eHealth platform	Initial version

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 (SAMLtoken), 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 programming guide for internal applications. Instead, it provides functional and technical information and allows an organization to integrate and use the eHealth platform service.

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

Technical and business requirements must be met, to allow the integration and validation of the eHealth platform service in the client application.

2.3 eHealth platform document references

On the portal of the eHealth platform, you can find all the referenced documents.[1]. These versions, or any following ones, can be used for the eHealth platform service.

ID	Title	Version	Date	Author
1	SOA – Error guide	1.0	10/06/2021	eHealth platform
2	Secure Token Service Cookbook Annex Mapping Certificate holder	1.2	20/01/2023	eHealth platform

2.4 External document references

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

ID	Title	Source	Date	Author
1	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
2	SAML	http://www.oasis-open.org/specs/index.php#samlv1.1	2010-08-30	OASIS

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

3	SAML Token Profile	http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-pr-SAMLTokenProfile-01.html	2010-08-30	
4.	WS – trust 1.4	http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.html	2012-04-25	

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 (BCP)

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 SAML token. If I.AM STS cannot provide a 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.

Sliding window principle can be applied only with a valid SAML token. In this situation, you can avoid short interruptions on I.AM STS.

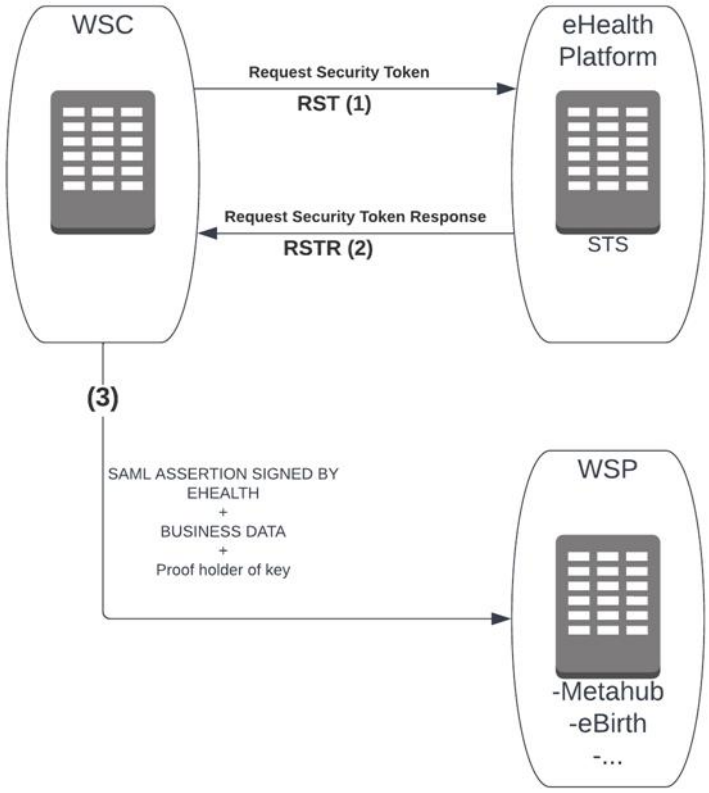
If the WSC has a valid SAML token, the WSC MUST

- After a given period (= X/2), request a new token.
  - If OK: this new SAML token will be valid for X hours from that time.
  - If not OK: the WSC MUST try to get a new one after X/4 hours
    …

The sliding window principle involves multiple eID pin-code introductions 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: The WSC sends a RequestSecurityToken (RST) to STS to get a security token.

Step 2: eHealth returns a signed SAML assertion in a RequestSecurityTokenResponse (RSTR) to the WSC.

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 wants to use two different applications. Every general practitioner (GP) has access to those two applications. The end user requests a SAML token to obtain the proof that he is a GP. The eHealth platform validates this claim against its validated Authentic sources (VAS). When a positive response is received, 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 is expired, the end user must request a new one. When the GP contacts a target application, he must send the session ticket together with the business data. With this session ticket, the application has a certified proof that the requestor is a GP and can execute its business.

5. Step-by-step

5.1 Technical requirements

The WS security model defined in WS-Trust is based on a process in which a WS can require an incoming message to prove a set of claims (e.g., SSIN, CBE number, etc.). If a message arrives without having the required proof of claims, the service will ignore or reject the message.

The web service expects a request composed of a RequestSecurityToken RST and returns a RequestSecurityTokenResponse (RSTR).

5.1.1 Security policies to apply

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

As web service 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 allows eHealth to verify the integrity of the message and the identity of its author.

The STS cookbook, explaining how to implement this security policy, can be found on the eHealth portal.

https://www.ehealth.fgov.be/ehealthplatform/nl/service-iam-identity-access-management (Dutch)

https://www.ehealth.fgov.be/ehealthplatform/fr/service-iam-identity-access-management (French)

5.1.2 WS-I Basic Profile 1.1

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

5.1.3 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***

5.2 Process overview

5.2.1 RequestSecurityToken elements and attributes

A RequestSecurityToken message is composed of multiple parts. The supported elements and attributes are listed below.

Field name	Description
RequestSecurityToken/@***Context***	This is an optional but recommended attribute that specifies an identifier/context for the request. All subsequent RSTR elements relating to this request MUST carry this attribute.
TokenType	Mandatory element that describes the type of security token requested which is, the type of token that will be returned in the RSTR. At the time, eHealth only supports SAML1 response. Value accepted : ***http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1***
RequestType	This mandatory element is used to indicate the class of function being requested. Allowed values : - To issue a new token, use ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue*** - To renew an existing token, use: ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew***
RenewTarget	This element is required when a token needs to be renewed. For more information about renewal, see section 5.2.1.2.
RenewTarget**/SecurityTokenReference**	Optional element that can be used to reference a security token.
RenewTarget**/**SecurityTokenReference **/Embedded**	Optional element that is used to embed an old assertion.
Claims	Optional element used to request a specific set of claims, typically this element contains required/optional claim information identified in a service's policy, see section 5.2.1.1.1
LifeTime	Optional element used to specify the desired valid time range for the returned security token.
LifeTime/**Created**	Optional element, represents the creation time of the security token.
LifeTime/**Expires**	Optional element, represents the validity time period of the requested token. The value should not be in the past and should be a valid date.
KeyType	Optional but recommended element that indicates the type of key desired in the security token. Default value : ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/PublicKey***
Usekey	Optional element used when the requestor wishes to use an existing key rather than the key used in section 6.1.2.It can reference the security token containing the desired key.

UseKey/SecurityTokenReference /X509Data**/X509Certificate**	Optional element to specify the holder-of-key certificate that will be used to proof the ownership of the token in the response.

5.2.1.1 RequestSecurityToken - Issue token

When the requestor does not have any security token, the requestor has to send a RST to STS where the RequestType is set to ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue*** to receive a security token.

Request example :

```
<wst:RequestSecurityToken xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
  <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1</wst:TokenType>
  <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
  <wst:Claims Dialect="http://docs.oasis-open.org/wsfed/authorization/200706/authclaims"></wst:Claims>
  <wst:Lifetime>
    <wsu:Created> 2023-01-10T07:30:10.000Z</wsu:Created>
    <wsu:Expires> 2023-01-10T08:30:10.000Z </wsu:Expires>
  </wst:Lifetime>
  <wst:KeyType>http://docs.oasis-open.org/ws-sx/wstrust/200512/PublicKey</wst:KeyType>
  <wst:UseKey>
    <wsse:SecurityTokenReference>
      <ds:X509Data>
        <ds:X509Certificate>${certificate4}</ds:X509Certificate>
      </ds:X509Data>
    </wsse:SecurityTokenReference>
  </wst:UseKey>
  </wst:RequestSecurityToken>
```

Reponse example :

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
   <SOAP-ENV:Header/>
   <SOAP-ENV:Body>
      <wst:RequestSecurityTokenResponse Context="RC-ab24230b-fae8-43e7-b1b2-c31cedff5235"
xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
         <wst:RequestedSecurityToken>
            <Assertion AssertionID="_7d8e5ce1c2d3b8b0b752efec9f54c3e1" IssueInstant="2023-01-
04T14:41:21.078Z" Issuer="urn:be:fgov:ehealth:sts:1 0" MajorVersion="1" MinorVersion="1"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
               <Conditions NotBefore="2023-01-04T06:25:10.000Z" NotOnOrAfter="2023-01-04T07:30:10.000Z"/>
               <AuthenticationStatement AuthenticationInstant="2023-01-04T14:41:21.078Z"
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="${SUBJECT-CA2}">&{SUBJECT X5093}</NameIdentifier>
                     <SubjectConfirmation>
                        <ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:holder-of-
key</ConfirmationMethod>
                        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                           <ds:X509Data>
                           <ds:X509Certificate>${certificate4}</ds:X509Certificate>
                           </ds:X509Data>
                        </ds:KeyInfo>
                     </SubjectConfirmation>
                  </Subject>
               </AuthenticationStatement>
               <AttributeStatement>
                  <Subject>
                     <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName"
NameQualifier="${SUBJECT-CA2}">${SUBJECT X5093}</NameIdentifier>
                  </Subject>
                  <Attribute AttributeName="urn:be:fgov:kbo-bce:organization:cbe-number"
AttributeNamespace="urn:be:fgov:identification-namespace">
```

```
                    <AttributeValue> CBE-number </AttributeValue>
                 </Attribute>
                 <Attribute AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:enterprise:cbe-
number" AttributeNamespace="urn:be:fgov:identification-namespace">
                    <AttributeValue> CBE-number </AttributeValue>
                 </Attribute>
              </AttributeStatement>
              <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/xmldsig-more#rsa-sha256"/>
                    <ds:Reference URI="#_7d8e5ce1c2d3b8b0b752efec9f54c3e1">
                       <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#"/>
                       </ds:Transforms>
                       <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                       <ds:DigestValue>Ks7…V40=</ds:DigestValue>
                    </ds:Reference>
                 </ds:SignedInfo>
                 <ds:KeyInfo>
                 <ds:SignatureValue> W46w+M/KcAYju… YrHEA==</ds:SignatureValue>
                    <ds:X509Data>
                       <ds:X509Certificate>${certificate[4]}</ds:X509Certificate>
                    </ds:X509Data>
                 </ds:KeyInfo>
              </ds:Signature>
           </Assertion>
        </wst:RequestedSecurityToken>
     </wst:RequestSecurityTokenResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.2.1.1.1 Claims

A claim is an information linked to the client, service or other resource. At eHealth it is used to provide information from the WSC to the STS. The STS knows 2 types of claims:

- Claims that are linked to the identity of the WSC (see 5.2.1.1.1.1)
- Claims that must be resolved/certified by the STS (see 5.2.1.1.1.2)

5.2.1.1.1.1 Claims linked to the identity of the WSC

The WSC requests the STS to assert that the value, provided for the claim, is accurate, corresponding to the WSC credentials as provided in the BinarySecurityToken (see section 6.1.2). The link between the identification credential and the claims that are allowed are described in eHealth platform document references - Mapping Certificate Holder (see 2.3). When the claims provided in the request are not linked to the identification credentials, the STS will respond with an error (see Chap8 Error and failure messages).

Example :

An organization hospital, with the identification number '71089914', wants to obtain a SAML token. It sends a request and it uses a X509 certificate (eHealth platform NIHII-HOSPITAL=71089914 certificate) to authenticate.

The request:

```
<wst:RequestSecurityToken Context="RC-${RequestId}" xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-
trust/200512" xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing" xmlns:wsu="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:wsse="http://docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV1.1</wst:TokenType>
         <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
         <wst:Claims Dialect="http://docs.oasis-open.org/wsfed/authorization/200706/authclaims">
```

```
            <auth:ClaimType Uri="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number">
               <auth:Value>71089914</auth:Value>
            </auth:ClaimType>
         </wst:Claims>
         <wst:Lifetime>
            <wsu:Created>2023-01-03T22:55:00.000Z</wsu:Created>
            <wsu:Expires>2023-01-03T23:55:00.000Z</wsu:Expires>
         </wst:Lifetime>
         <wst:KeyType>http://docs.oasis-open.org/ws-sx/wstrust/200512/PublicKey</wst:KeyType>
   </wst:RequestSecurityToken>
```

The response

```
<wst:RequestSecurityTokenResponse Context="RC-a55c3420-6ae3-4d17-bebe-e70daf1f557c"
xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
         <wst:RequestedSecurityToken>
            <Assertion AssertionID="_444b70688b525e6c19f582e282855c48" IssueInstant="2023-01-
11T13:20:46.160Z" Issuer="urn:be:fgov:ehealth:sts:1 0" MajorVersion="1" MinorVersion="1"
xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
               <Conditions NotBefore="2023-01-03T22:55:00.000Z" NotOnOrAfter="2023-01-04T23:00:00.000Z"/>
               <AuthenticationStatement AuthenticationInstant="2023-01-11T13:20:46.160Z"
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=${SUBJECT-CA[2]}>${SUBJECT[3]}</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>${certificate[4]}</ds:X509Certificate>
                           </ds:X509Data>
                        </ds:KeyInfo>
                     </SubjectConfirmation>
                  </Subject>
               </AuthenticationStatement>
               <AttributeStatement>
                  <Subject>
                     <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName"
NameQualifier=${Subject-CA[2]}>${Subject[3]}</NameIdentifier>
                  </Subject>
                  <Attribute AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-
number" AttributeNamespace="urn:be:fgov:identification-namespace">
                     <AttributeValue>71089914</AttributeValue>
                  </Attribute>
               </AttributeStatement>
               <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/xmldsig-more#rsa-sha256"/>
                     <ds:Reference URI="#_444b70688b525e6c19f582e282855c48">
                        <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#"/>
                        </ds:Transforms>
                        <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
                     <ds:DigestValue>zhPBGtUjR4PPRgJmEMQBflbiCGI4QZSf5KBYodo7qd4=</ds:DigestValue>
```

[2] The distinguished names of the Certificate Authority of the identification certificate (use RFC1779 or RFC2253)

[3] The distinguished names of the certificate (use RFC1779 or RFC2253)

[4] A Base 64 certificate

```
                    </ds:Reference>
                </ds:SignedInfo>
                <ds:SignatureValue>ZXyW...h9WbOw==</ds:SignatureValue>
                <ds:KeyInfo>
                    <ds:X509Data>
                        <ds:X509Certificate>${certificate[4]}</ds:X509Certificate>
                    </ds:X509Data>
                </ds:KeyInfo>
            </ds:Signature>
        </Assertion>
    </wst:RequestedSecurityToken>
</wst:RequestSecurityTokenResponse>
```

5.2.1.1.1.2 Claims that must be resolved/certified by the STS

The WSC requests STS to resolve a claim value and to assert it. The correct identification claims must be provided in order to resolve claims without a value. If the context or requested claims are incorrect the STS will respond with an error. (see Chap 8 Error and failure messages)

Example:

An organization hospital, with the identification number '71089914', wants to obtain a SAML token.It sends a request and it uses a X509 certificate (eHealth platform NIHII-HOSPITAL=71089914 certificate) to authenticate. The STS must certify that 71089914 is still a recognized hospital.

The request :

```
<wst:RequestSecurityToken Context="RC-${RequestId}"
    xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
    xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706"
    xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy"
    xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing"
    xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
    xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1</wst:TokenType>
    <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
    <wst:Claims Dialect="http://docs.oasis-open.org/wsfed/authorization/200706/authclaims">
        <auth:ClaimType Uri="urn:be:fgov:ehealth:1.0:hospital:nihii-number">
            <auth:Value>71089914</auth:Value>
        </auth:ClaimType>
        <auth:ClaimType Uri="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number">
            <auth:Value>71089914</auth:Value>
        </auth:ClaimType>
        <auth:ClaimType Uri="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-
number:recognisedhospital:boolean"/>
    </wst:Claims>
    <wst:Lifetime>
        <wsu:Created>2023-01-03T22:55:00.000Z</wsu:Created>
        <wsu:Expires>2023-01-03T23:55:00.000Z</wsu:Expires>
    </wst:Lifetime>
    <wst:KeyType>http://docs.oasis-open.org/ws-sx/wstrust/200512/PublicKey</wst:KeyType>
</wst:RequestSecurityToken>
```

The response

```
<wst:RequestSecurityTokenResponse Context="RC-a55c3420-6ae3-4d17-bebe-e70daf1f557c"
    xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
<wst:RequestedSecurityToken>
    <Assertion AssertionID="_444b70688b525e6c19f582e282855c48"
        IssueInstant="2023-01-11T13:20:46.160Z" Issuer="urn:be:fgov:ehealth:sts:1_0"
        MajorVersion="1" MinorVersion="1" xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
        <Conditions NotBefore="2023-01-03T22:55:00.000Z" NotOnOrAfter="2023-01-04T23:00:00.000Z" />
```

```
        <AuthenticationStatement AuthenticationInstant="2023-01-11T13:20:46.160Z"
            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="${SUBJECT-CA[2]}">${SUBJECT X509[3]}</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>${certificate[4]}</ds:X509Certificate>
                        </ds:X509Data>
                    </ds:KeyInfo>
                </SubjectConfirmation>
            </Subject>
        </AuthenticationStatement>
        <AttributeStatement>
            <Subject>
                <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName"
NameQualifier="${SUBJECT-CA[2]}">${SUBJECT X509[3]}</NameIdentifier>
            </Subject>
            <Attribute AttributeName="urn:be:fgov:ehealth:1.0:hospital:nihii-number"
                AttributeNamespace="urn:be:fgov:identification-namespace">
                <AttributeValue>71089914</AttributeValue>
            </Attribute>
            <Attribute
                AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number"
                AttributeNamespace="urn:be:fgov:identification-namespace">
                <AttributeValue>71089914</AttributeValue>
            </Attribute>
                <Attribute
                    AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-
number:recognisedhospital:boolean"
                    AttributeNamespace="urn:be:fgov:certified-namespace:ehealth">
                    <AttributeValue>true</AttributeValue>
                </Attribute>
        </AttributeStatement>
        <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/xmldsig-more#rsa-sha256" />
                <ds:Reference URI="#_444b70688b525e6c19f582e282855c48">
                    <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#" />
                    </ds:Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                    <ds:DigestValue>zhPBGtUjR4PPRgJmEMQBflbiCGI4QZSf5KBYodo7qd4=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>ZXyW...h9WbOw==</ds:SignatureValue>
            <ds:KeyInfo>
                <ds:X509Data>
                    <ds:X509Certificate>${certificate[4]}</ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </ds:Signature>
    </Assertion>
</wst:RequestedSecurityToken>
```

5.2.1.2 RequestSecurityToken - Renew token

To renew an expired security token, the requestor must send a RST request where the parameter ***RequestType*** is set to ***http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Renew***

The identification credentials used in the original request must be used again (see section 6.1.2).
By sending this request, a token is returned with a new lifespan.

Request example :

```
<wst:RequestSecurityToken Context="RC-124" xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
xmlns:auth="http://docs.oasis-open.org/wsfed/authorization/200706"
xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy" xmlns:wsa="http://www.w3.org/2005/08/addressing"
xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
         <wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV1.1</wst:TokenType>
         <wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Renew</wst:RequestType>
         <wst:RenewTarget>
            <wsse:SecurityTokenReference>
               <wsse:Embedded wsu:Id="token-e4f5c593533e313baddb270183e13753">
    <Assertion AssertionID="_444b70688b525e6c19f582e282855c48"
        IssueInstant="2023-01-11T13:20:46.160Z" Issuer="urn:be:fgov:ehealth:sts:1_0"
        MajorVersion="1" MinorVersion="1" xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
        <Conditions NotBefore="2023-01-03T22:55:00.000Z" NotOnOrAfter="2023-01-04T23:00:00.000Z" />
        <AuthenticationStatement AuthenticationInstant="2023-01-11T13:20:46.160Z"
            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="${SUBJECT-CA[2]}">${SUBJECT X509[3]}</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>${certificate[4]}</ds:X509Certificate>
                        </ds:X509Data>
                    </ds:KeyInfo>
                </SubjectConfirmation>
            </Subject>
        </AuthenticationStatement>
        <AttributeStatement>
            <Subject>
                <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName"
NameQualifier="${SUBJECT-CA[2]}">${SUBJECT X509[3]}</NameIdentifier>
            </Subject>
            <Attribute AttributeName="urn:be:fgov:ehealth:1.0:hospital:nihii-number"
                AttributeNamespace="urn:be:fgov:identification-namespace">
                <AttributeValue>71089914</AttributeValue>
            </Attribute>
            <Attribute
                AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number"
                AttributeNamespace="urn:be:fgov:identification-namespace">
                <AttributeValue>71089914</AttributeValue>
            </Attribute>
                <Attribute
                    AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-
number:recognisedhospital:boolean"
                    AttributeNamespace="urn:be:fgov:certified-namespace:ehealth">
                    <AttributeValue>true</AttributeValue>
                </Attribute>
        </AttributeStatement>
        <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/xmldsig-more#rsa-sha256" />
                <ds:Reference URI="#_444b70688b525e6c19f582e282855c48">
                    <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#" />
                    </ds:Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                    <ds:DigestValue>zhPBGtUjR4PPRgJmEMQBflbiCGI4QZSf5KBYodo7qd4=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>ZXyW...h9WbOw==</ds:SignatureValue>
            <ds:KeyInfo>
                <ds:X509Data>
                    <ds:X509Certificate>${certificate4}</ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </ds:Signature>
    </Assertion>
               </wsse:Embedded>
            </wsse:SecurityTokenReference>
         </wst:RenewTarget>
      </wst:RequestSecurityToken>
```

Response example :

```
<wst:RequestSecurityTokenResponse Context="RC-a55c3420-6ae3-4d17-bebe-e70daf1f557c"
    xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
<wst:RequestedSecurityToken>
    <Assertion AssertionID="_444b70688b525e6c19f582e282855c48"
        IssueInstant="2023-01-11T13:20:46.160Z" Issuer="urn:be:fgov:ehealth:sts:1_0"
        MajorVersion="1" MinorVersion="1" xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
        <Conditions NotBefore="2023-01-03T22:55:00.000Z" NotOnOrAfter="2023-01-04T23:00:00.000Z" />
        <AuthenticationStatement AuthenticationInstant="2023-01-11T13:20:46.160Z"
            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="${SUBJECT-CA2}">${SUBJECT X5093}</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>${certificate4}</ds:X509Certificate>
                        </ds:X509Data>
                    </ds:KeyInfo>
                </SubjectConfirmation>
            </Subject>
        </AuthenticationStatement>
        <AttributeStatement>
            <Subject>
                <NameIdentifier Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName"
NameQualifier="${SUBJECT-CA2}">${SUBJECT X5093}</NameIdentifier>
            </Subject>
            <Attribute AttributeName="urn:be:fgov:ehealth:1.0:hospital:nihii-number"
                AttributeNamespace="urn:be:fgov:identification-namespace">
                <AttributeValue>71089914</AttributeValue>
            </Attribute>
            <Attribute
                AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-number"
                AttributeNamespace="urn:be:fgov:identification-namespace">
```

```
                <AttributeValue>71089914</AttributeValue>
            </Attribute>
                <Attribute
                    AttributeName="urn:be:fgov:ehealth:1.0:certificateholder:hospital:nihii-
number:recognisedhospital:boolean"
                    AttributeNamespace="urn:be:fgov:certified-namespace:ehealth">
                    <AttributeValue>true</AttributeValue>
                </Attribute>
        </AttributeStatement>
        <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/xmldsig-more#rsa-sha256" />
                <ds:Reference URI="#_444b70688b525e6c19f582e282855c48">
                    <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#" />
                    </ds:Transforms>
                    <ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                    <ds:DigestValue>zhPBGtUjR4PPRgJmEMQBflbiCGI4QZSf5KBYodo7qd4=</ds:DigestValue>
                </ds:Reference>
            </ds:SignedInfo>
            <ds:SignatureValue>ZXyW...h9WbOw==</ds:SignatureValue>
            <ds:KeyInfo>
                <ds:X509Data>
                    <ds:X509Certificate>${certificate[4]}</ds:X509Certificate>
                </ds:X509Data>
            </ds:KeyInfo>
        </ds:Signature>
    </Assertion>
</wst:RequestedSecurityToken>
</wst:RequestSecurityTokenResponse>
```

5.2.2 RequestSecurityTokenResponse elements and attributes

The RSTR element is used to return a security token or response to a RST.

The type of security token depends on what was requested by the WSC as TokenType.

In the case of ***http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1 ,*** this will be a SAML 1.1 Assertion, signed by eHealth as trusted authority, containing the requested claims and the client's public key as Holder-of-Key.

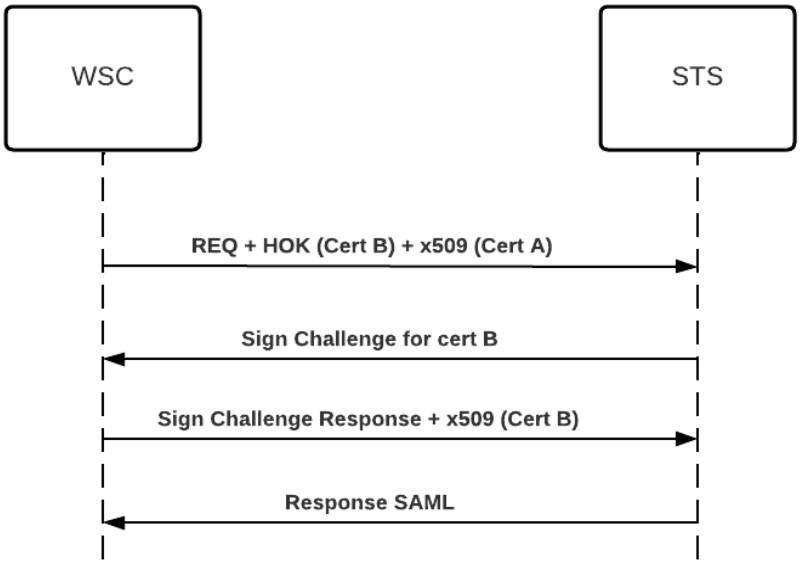
The WSC can use that Assertion in following steps as authentication mechanism to send requests to SOAP services.

In that process, he must take care to change nothing in the Assertion or the signature on it will no longer be valid and his requests will be rejected.

More information about the Holder-of-key mechanism can be found in this document
http://docs.oasisopen.org/wss/v1.1/wss-v1.1-spec-pr-SAMLTokenProfile-01.html

5.2.2.1 RequestSecurityTokenResponse – SignChallenge

All requests sent to STS have to be signed. If the identification credentials do not match the key used in the UseKey field a SignChallenge will be returned.



1. The WSC send a RST containing certificate B as UseKey and signs the request with certificate A.
2. STS will send a RSTR back to the WSC with a sign challenge.
3. The WSC replies the sign challenge (SignChallengeResponse) and signs the request with certificate B
4. If everything matches, STS will respond with a RSTR with an assertion.

SignChallenge description

Field name	Description
SignChallenge	This OPTIONAL element describes a challenge that requires the other party to sign a specified set of information.
*SignChallenge/**Challenge***	This REQUIRED string element describes the value to be signed

SignChallengeResponse description

Field name	Description
SignChallengeResponse	This OPTIONAL element describes a response to a challenge that requires the signing of a specified set of information.
*SignChallengeResponse/**Response***	If a challenge was issued, the response MUST contain the challenge element exactly as received. As well, while the RSTR response SHOULD always be signed, if a challenge was issued, the RSTR MUST be signed

Below, you find an example of the messages exchanged in this process :

Step 1: WSC sends a RST to the STS

```
<wst:RequestSecurityToken>
            <wst:TokenType>
                http://example.org/mySpecialToken
            </wst:TokenType>
            <wst:RequestType>
                http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue
            </wst:RequestType>
              <wst:Lifetime>
                 <wsu:Created>2023-01-10T16:39:58.812Z</wsu:Created>
                 <wsu:Expires>2023-01-10T16:50:58.000Z</wsu:Expires>
              </wst:Lifetime>
            <wst:KeyType>http://docs.oasis-open.org/ws-
            sx/wstrust/200512/PublicKey</wst:KeyType>
            <wst:UseKey>
               <wsse:SecurityTokenReference>
                  <ds:X509Data>
                    <ds:X509Certificate>$(certificate[4])
 </ds:X509Certificate>
                  </ds:X509Data>
               </wsse:SecurityTokenReference>
          </wst:UseKey>
</wst:RequestSecurityToken>
```

Step 2: STS responds with a RSTR with a SignChallenge

```
<wst:RequestSecurityTokenResponse>
            <wst:SignChallenge>
                <wst:Challenge>Huehf...</wst:Challenge>
            </wst:SignChallenge>
        </wst:RequestSecurityTokenResponse>
```

Step 3: WSC response with signed challenge

```
<wst:RequestSecurityTokenResponse>
            <wst:SignChallengeResponse>
                <wst:Challenge>Huehf...</wst:Challenge>
            </wst:SignChallengeResponse>
</wst:RequestSecurityTokenResponse>
```

Step 4: STS responds with a RSTR with an assertion

```
<wst:RequestSecurityTokenResponse Context="RC-a55c3420-6ae3-4d17-bebe-e70daf1f557c"
    xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
<wst:RequestedSecurityToken>
    …
    </Assertion>
</wst:RequestedSecurityToken>
</wst:RequestSecurityTokenResponse>
```

5.2.2.2 RequestSecurityTokenResponse – RequestedSecurityToken

Here a description of these elements.

Field name	Description
@Context	This attribute specifies the identifier from the original request. If the context was mentioned in the RST then it must be echoed in the RSTR.
RequestedSecurityToken	This element is used to return the requested security token and it contains a SAML Assertion with - an AuthenticationStatement - an AttributeStatement - a Signature on the Assertion (signed with eHealth IAM certificate) 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.

The RequestedSecurityToken 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. If 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 period, 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 is asserting 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 is asserting that the requested attributes are applicable to the referenced subject. Only the *NameIdentifier* of the subject is repeated. For every requested attribute an <Attribute> is present. When there is no value for an attribute an empty <Attribute> tag is returned.

For example, when an end user wants the proof that he possesses attributes a and b, an empty tag will be returned for attribute b if the STS could not find any value in its authentic sources for the given end user's identity (or value 'false' in case of attributes of type 'boolean').

```
<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>
```

5.2.3 Service Endpoints

eHealth STS WS-trust has the following endpoints:

- *Acceptation environment:* ***https://services-acpt.ehealth.fgov.be/IAM/SecurityTokenService/v1***
- *Production environment:* ***https://services.ehealth.fgov.be/IAM/SecurityTokenService/v1***

6. Risks and security

6.1 Security

6.1.1 Business security

In case the development adds a use case based on an existing integration, the eHealth platform must be informed at least one month in advance. A detailed estimate of the expected load is necessary to be able to ensure an effective capacity management.

When technical issues occur on the WS, the partner can obtain support from the contact centre (see Chap 3)

If the eHealth platform should find a bug or vulnerability in its software, the partner must update his application with the latest version of the software, within ten (10) business days.

If the partner finds a bug or vulnerability in the software or web service made available by the eHealth platform, he is obliged to contact and inform us immediately. He is not allowed, under any circumstances, to publish this bug or vulnerability.

6.1.2 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 allows the eHealth platform to verify the integrity of the message and the identity of its author.
- No encryption on the message.

6.1.3 The use of username, password and token

The username, password, and token are strictly personal.
Every user takes care of his username, password and token, and he is forced to confidentiality of it. It is prohibited to transfer them to partners and clients. Until inactivation, every user is responsible for every use, including the use by a third party.

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 information needed to integrate is published on the portal of the eHealth platform.

Upon request and depending on the case, the eHealth platform provides you with a **test case** in order for you to test your client before releasing it in the acceptance environment.

7.1.3 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 sends his test and performance results with a sample of “eHealth request” and “eHealth answer” by email to his point of contact at the eHealth platform.

Once a release date has been agreed on, 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***.

7.1.4 Operational follow-up

Once in production, the partner using the eHealth platform service for one of his applications will always test in the acceptance environment first 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

There are different types of responses:

- If there are no technical errors, responses as described in section 5.2.2 are returned.
- In case of a technical error, a SOAP fault exception is returned (see table below).

If an error occurs, first please verify your request. The following table contains a list of common system error codes for the eHealth Service Bus/Gateway. For more information, see document *SOA - Error Guide* on eHealth portal.

Table 1 : Description of the possible SOAP fault exceptions.

Error code	Origin	Description
SOA-00001	Not determined	Service error
SOA-01001	Consumer	Service call not authenticated
SOA-01002	Consumer	Service call not authorized
SOA-02001	Provider	Service not available. Please contact service desk
SOA-02002	Provider	Service temporarily not available. Please try later
SOA-03001	Consumer	Malformed message
SOA-03002	Consumer	Message must be SOAP
SOA-03003	Consumer	Message must contain SOAP body
SOA-03004	Consumer	WS-I compliance failure
SOA-03005	Consumer	WSDL compliance failure
SOA-03006	Consumer	XSD compliance failure
SOA-03007	Consumer	Message content validation failure

If the cause is a business error, a SOAP fault exception is returned (see table below). The list is not exhaustive.

Table 2 : Description of the possible error.

Error code	Messages	Solution/Explanation
InvalidRequest	Message not properly encoded Extracting KeyType [XXX] failed	This error is sent in case the referenced RequestType in the request does not have the excepted value. The valid value can be found in the chapter 5.2.1
InvalidRequest	Message not properly encoded Extracting TokenType [XXX] failed	This error is sent in case the referenced TokenType in the request does not have the excepted value. The valid value can be found in the chapter 5.2.1
InvalidRequest	Message not properly encoded Extracting KeyType [XXX] failed	This error is sent in case the referenced KeyType in the request does not have the excepted value. The valid value can be found in the chapter 5.2.1

urn:oasis:names:tc:SAML:2.0:status:InvalidAttrNameOrValue	AttributeAuthority could not resolve attributes Attribute XXX not supported	This error is sent when referencing an attribute that is not supported in the requested claims.
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	Message did not meet security requirements X.509 Attribute Mismatch	This error is sent when using a different certificate holder value than the one referenced in the BinarySecurityToken; (seeeHealth platform document references – STS – Annex Mapping Certificate Holder)
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	Message did not meet security requirements URI of CertificateHolder Attribute in Request [TypeA] does not match URI of CertificateHolder Attribute in Authentication Credential [TypeB].	This error is sent when using a different CertificateHolder Attribute than the one linked to the BinarySecurityToken.
urn:oasis:names:tc:SAML:2.0:status:RequestDenied	Message did not meet security requirements Invalid identity attributes combination.	This error is sent when using a bad combination of claims (see section 5.2.1.1.1)
urn:be:fgov:ehealth:1.0:status:Indeterminate	AttributeAuthority could not resolve attributes Required attribute missing: …	This error indicates a missing Claim (see section 5.2.1.1.1)

Business error example :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <soapenv:Fault>
      <faultcode xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">wst:InvalidRequest</faultcode>
      <faultstring>The request was invalid or malformed</faultstring>
      <detail>
        <urn:BusinessError Id="_86996cb037b4d6105a914879a2d14617" xmlns:urn="urn:be:fgov:ehealth:errors:soa:v1">
          <Origin>Client</Origin>
          <Code>urn:oasis:names:tc:SAML:2.0:status:RequestDenied</Code>
          <Message xml:lang="en">Message did not meet security requirements</Message>
          <Message xml:lang="en">Invalid identity attributes combination.</Message>
          <urn:Environment>Integration</urn:Environment>
        </urn:BusinessError>
      </detail>
    </soapenv:Fault>
  </soapenv:Body>
</soapenv:Envelope>
```