socialsecurity.be

# Social Security

# Service Specification:

## Integration and use of the Service

# SecurityTokenService (STS)

## Version 2

## Into an external application

**To the attention of: “IT expert” aiming to integrate this Service**

# Table of Contents

# 1 Goal of the service

## 1.1 Description of the service

The STS (Security Token Service) will allow service consumers to request the issuance of a security token. This security token will be used by the consumer application as a trusted identity assertion in order to authenticate itself to other service providers. The STS works as an brokered authentication service for service consumers on the SOA platform. It allows consumer applications to authenticate once and access multiple service providers with the same token ("SSO for webservices").

## 1.2 Service consumers

The target consumers are the service consumers of the Social Security SOA platform which are known as "expeditor" in the user management application (a service consumer of the STS should be able to provide its "expeditor number" in order to authenticate via the STS).

# 2 Document management

## 2.1 Document history

*Comment: For publication in the registry, only the major release is published with the author <InstitutionName>. Inside the repository, the release history is published.*

| Version | Date | Author | Description of changes / remarks |
|---|---|---|---|
| 0.5 | 04/10/2010 | A. Clerbaut, G. Lemaire | First draft. |
| 0.6 | 14/12/2010 | W. Salembier | Additions |
| 2.0 | 20/10/2021 | G. Lemaire | STS v2 + SHA-1 deprecation |
| | | | |

## 2.2 Validation

| Reviewers | Name | Version Reviewed | Remarks |
|---|---|---|---|
| Project manager | Denis Vandersteene | x.y | |
| Architect | Willem Salembier | x.y | |
| | | | |

## 2.3 Document references

| ID | Title | Version | Date | Author |
|---|---|---|---|---|
| WS-Trust | WS-Trust OASIS standard (link) | 1.3 | 19/03/2007 | OASIS |
| SAML Binding | Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) (link) | 1.1 | 02/09/2003 | OASIS |
| UMan | Beveiligde toegang voor werkgevers RSZ (link nl) (link fr) | | | RSZ/ONSS |

## 2.4 Goal of the document

This document provides functional and technical information on calling the Service STS, as provided by the social security.

In this service specification document, we explain the structure and content aspects of the possible service requests and replies. An example illustrates each of those messages. Also, the list of possible errors is included in this document.

This information should allow (the IT department of) an organization to develop and use the service.

Some technical and legal requirements must be satisfied in order to allow the integration of the services in client applications; this document was written in order to provide you with an overview of requirements which have to be met in order to integrate correctly with the services offered by the Social Security.

This document is not a development or a programming guide for internal applications. In order to interact in a smooth, homogeneous and risk controlled way with a maximum of partners, partners must commit to comply with specifications, data format, and release processes described within this document.

# 3 Service history

This chapter contains the list of changes to the service since the previous publication.

Remark: If only the minor(y) number has changed, the Service is backward compatible with the previous version. Existing consumers with no need to use the new functionality do not have to change their implementation.

**Previous version number:**

None

**Previous release date:**

None

**List of changes:**

Not applicable

# 4 Global overview of the Service Group

The STS implements the OASIS WS-Trust 1.3 standard (ref WS-Trust) to issue security tokens which can be reused as a proof of identity in order to access business services on the Social Security SOA platform. The STS provides a kind of single-sign-on token to the service consumer who wants to access various service providers in a short lap of time. In order to obtain this token a service

consumer should be able to claim its identity (which means that the the consumer should provide informations that can be processed and verified by the STS and which constitute a proof of identity).

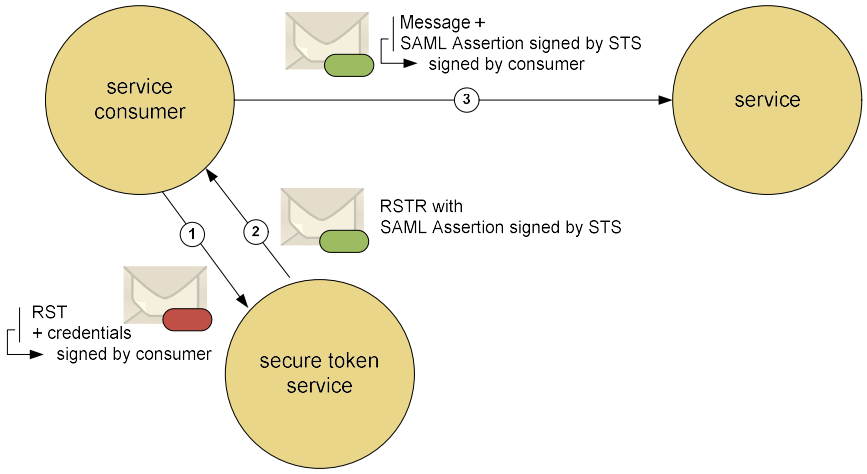
The claims currently supported are:

- The expeditor number
- The pair of the organisation identifier and quality

The STS will consult the User Management database in order to verify the claims of the service consumer and retrieve informations about the service consumer.

The STS will then return a signed SAML assertion which contains the verified identity of the service consumer. The fact that the assertion is signed by the STS proves that the data (contained within the SAML assertion) have been validated by the STS.

The service consumer may use this SAML assertion when calling service provider in order to prove its identity. The schema below presents the common use-case of an STS. This kind of usage of the STS is called “SAML Holder Of Key” (ref SAML Binding).




Flow :

1. The service consumer authenticates by sending credentials to the SAML issuer.
2. The STS issues a SAML assertion
3. The service consumer
    1. incorporates the SAML assertion into the message
    - signs both using its own private key (therby proving it’s possession)
    - sends the combined message to the service

# 5 Prerequisites

## 5.1 Business prerequisites

To use the STS with an expeditor number you must :

- be registered in the user management of social security as expeditor;
- have activated the Webservice channel for your expeditor.

To use the STS with an enterprise id and quality you must :

- be registered in the user management of social security as end-user

Documentation about the user management of social security can be found here (ref UMan).

## 5.2 Technical prerequisites

To prove its identity to the STS, a consumer application should be able to provide a signature which matches the certificate registered in the user management of social security. To build this signature, you must possess a certificate and the associated private key.

# 6 Test, release and exploitation procedures

The service is accessible in the acceptance environment for test purpose.

# 7 Description of the common types

Not applicable

# 8 Description of the service operations

## 8.1 Operation: RequestSecurityToken

### 8.1.1 Functional description

**For enterprises**

This function delivers a security token representing an expeditor or enterprise/quality pair based on the expeditor number. The requestor must be registered in the social security usermanagement as expeditor.

To obtain a security token, the requestor claims his expeditor number in the request and signs the request with the certificate registered in the social security usermanagement for this expeditor number.

**For end-users**

This function delivers a security token representing an end-user within an enterprise based on the social security identification number and information about the enterprise and quality. The requestor must be registered in the social security usermanagement as end-user.

To obtain a security token, the requestor claims his existance within an enterprise and quality in the request and signs the request with the eID Authentication certificate.

### 8.1.2 Request message construction

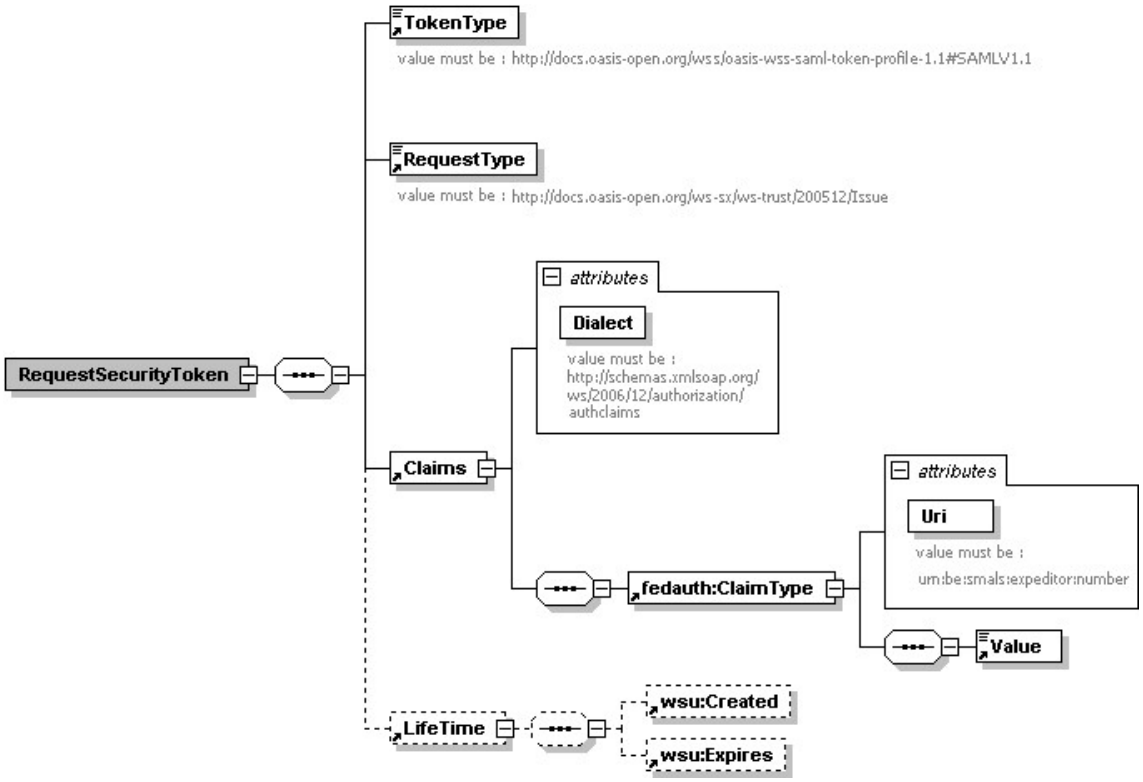
Messages exchanged between a service consumer and the STS are standard SOAP 1.1 messages.

The content of the SOAP body respects the OASIS standard : WS-Trust 1.3.

SPEC : http://docs.oasis-open.org/ws-sx/ws-trust/v1.3/ws-trust.html
XSD : http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3.xsd

Note: the namespace prefixes used in this document are definied in the WS-Trust 1.3 specification document.

The body's content of the SOAP request is a **<wst:RequestSecurityToken>** element (RST). The structure of the XML must be compliant with the graphically-represented XSD below :




### *8.1.2.1 Description of XML elements and attributes*

#### 8.1.2.1.1 Element wst:TokenType

This element is used to specify the type of token which we want to receive in the response. There are various type of token (Kerberos, Username, X509, SAML, …) but only one type of token is currently supported by the STS.

```
<wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1</wst:TokenType>
```

The RST **MUST** contain a **<wst:TokenType>** element with value
**http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1**

#### 8.1.2.1.2 Element wst:RequestType

This element allows the service consumer to tell the STS which action we want to be done (token issuance, token renewal, token validation). At present, the only action that is supported by the STS is"token issuance".

```
<wst:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</wst:RequestType>
```

The RST **MUST** contain a **<wst:RequestType>** element with value
**http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue**

#### 8.1.2.1.3 Element wst:Claims

This element must be used in order to send informations (claims) about the service consumer to the STS. Those informations will be validated by the STS (some verifications will be done in order to be sure that the service consumer is what it claims to be).

```
<wst:Claims Dialect="http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims">
        ...
</wst:Claims>
```

The RST **MUST** contain a **<wst:Claims>** element with Dialect attribute value equal to **http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims**

**For enterprises**

The **<wst:Claims>** tag **MUST** contain

- exactly one **<auth:ClaimType>** element with **Uri** attribute equal to **urn:be:smals:expeditor:number** and an **<auth:Value>** sub-element containing the expeditor number of the requestor (integer value).

```
<wst:Claims Dialect="http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims">
        <auth:ClaimType Uri="urn:be:smals:expeditor:number">
                <auth:Value>100035</auth:Value>
        </auth:ClaimType>
</wst:Claims>
```

**For end-users**

The **<wst:Claims>** tag **MUST** contain

- exactly one **<auth:ClaimType>** element with **Uri** attribute equal to one of the enterprise level identifiers and an **<auth:Value>** sub-element containing the identifier value
- exactly one **<auth:ClaimType>** element with **Uri** attribute equal **urn:be:smals:um:entity:quality** and an **<auth:Value>** sub-element containing a value from the quality lists.

```
<wst:Claims Dialect="http://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims">
        <auth:ClaimType Uri=" urn:be:fgov:kbo-bce:organization:cbe-number">
                <auth:Value>202239951</auth:Value>
        </auth:ClaimType>
        <auth:ClaimType Uri="urn:be:smals:um:entity:quality">
                <auth:Value>QUAL_EMP_NOSS</auth:Value>
        </auth:ClaimType>
</wst:Claims>
```

| Enterprise level identifiers | |
|---|---|
| urn:be:fgov:kbo–bce:organization:cbe-number | Company id |
| urn:be:smals:um:entity:ssin | Social security identification nr |
| **Quality identifiers** | |
| QUAL_COMPANY | Onderneming zonder personeel / Entreprise sans personnel |
| QUAL_EMP_NOSS | Werkgever RSZ / Employeur ONSS |
| QUAL_EMP_NOSSPLA | Werkgever RSZPPO / Employeur ONSSAPL |
| QUAL_FSC | Full service secretariaat / Secrétariat full service |
| QUAL_SP_LEG | Dienstverlener(rechtspersoon) /<br>Prestataire de services(personne morale) |
| QUAL_SSC | Erkend sociaal secretariaat / Secrétariat social agréé |
| QUAL_SP_IND | Dienstverlener(natuurlijke persoon) /<br>Prestataire de services(personne physique) |
| QUAL_CUR | Beheer van de curateles / Gestion des curatelles |

#### 8.1.2.1.4 Element wst:Lifetime

```
<wst:Lifetime>
        <wsu:Created>2010-05-06T22:04:34</wsu:Created>
        <wsu:Expires>2011-05-07T10:04:34</wsu:Expires>
</wst:Lifetime>
```

The RST **MAY** have a **<wst:Lifetime>** element.

The **<wsu:Created>** element **MAY** be present. If the **<wsu:Created>** is specified, it's value **MUST** be past [T – 60s] and before [T +60s] where T is the current time. If not present the created timestamp will be the time when the STS receives the request. It's not recommended to use this tag.

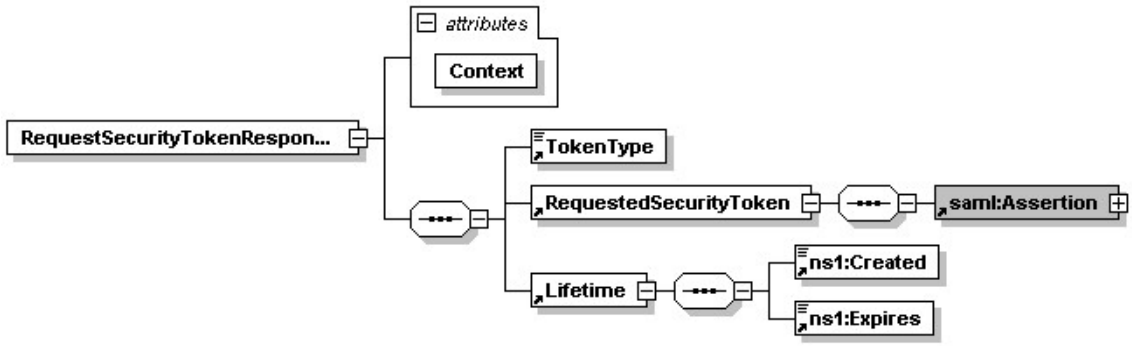
The **<wsu:Expired>** element **MAY** be present. If the **<wsu:Expired>** is specified, it's value **MUST** be between [Created] and [Created + 1h]. If not present the expired timestamp is [Created + 1h].

#### *8.1.2.2 Sample of a full RST*

```
<wst:RequestSecurityToken
   xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
   xmlns:auth="http://schemas.xmlsoap.org/ws/2006/12/authorization"
   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">
  <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://schemas.xmlsoap.org/ws/2006/12/authorization/authclaims">
    <auth:ClaimType Uri="urn:be:smals:expeditor:number">
      <auth:Value>123456</auth:Value>
    </auth:ClaimType>
  </wst:Claims>
</wst:RequestSecurityToken>
```

### 8.1.3 Reply message interpretation

The response provided by the STS is a **<wst:RequestSecurityTokenResponse>** (RSTR). This RSTR will follow the structure of the following schema :




#### *8.1.3.1 Description of XML elements and attributes*

##### 8.1.3.1.1 Attribute Context

If the RST specifies a Context attribute, the exact same value is returned in the RSTR response.

##### 8.1.3.1.2 Element wst:TokenType

This element indicates the type of token which is returned by the STS.

```
<wst:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1</wst:TokenType>
```

The RSTR **will always contain** a **<wst:TokenType>** element with value :

http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1

#### 8.1.3.1.3 Element wst:Lifetime

This element indicates the lifetime of the returned security token. It tells at what time the token will expire. Note that this information is somewhat redundant because it can be extracted from the token itself (in case of a SAML token).

```
<wst:Lifetime>
        <wsu:Created>2010-05-06T22:04:34</wsu:Created>
        <wsu:Expires>2011-05-06T23:04:34</wsu:Expires>
</wst:Lifetime>
```

The RSTR **will always contain** a **<wst:Lifetime>** element.

#### 8.1.3.1.4 Element wst: RequestedSecurityToken

This element contains the security token. In the current version of the STS, this security token will always be a SAML assertion (version 1.1). The SAML assertion must be provided when calling a secure webservice of social security.

The RSTR **contains** a **<wst:RequestedSecurityToken>** element which contains a SAML assertion.

### *8.1.3.2 Sample of a full RSTR*

```
<wst:RequestSecurityTokenResponse
        Context="abc"
        xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
        xmlns:auth="http://schemas.xmlsoap.org/ws/2006/12/authorization"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
        xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">

        <wst:TokenType>
                http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile1.1#SAMLV1.1
        </wst:TokenType>

        <wst:RequestedSecurityToken>

                <saml:Assertion … >
                        …
                </saml:Assertion>

        </wst:RequestedSecurityToken>
        <wst:Lifetime>
                <wsu:Created>2010-05-06T22:04:34</wsu:Created>
                <wsu:Expires>2010-05-06T22:14:34</wsu:Expires>
        </wst:Lifetime>
</wst:RequestSecurityTokenResponse>
```

Example of Assertion contained in RequestSecurityTokenResponse :

```
<saml:Assertion
        AssertionID="ID_d968adf1-bd09-48aa-b19a-5c6aca32ad9c"
        IssueInstant="2010-09-07T14:28:35.398Z"
        Issuer="http://services.socialsecurity.be/sts"
        MajorVersion="1"
        MinorVersion="1"
        xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
        xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">

        <saml:Conditions
                NotBefore="2010-09-07T14:28:35.194Z"
                NotOnOrAfter="2010-09-07T15:28:35.194Z"/>

        <saml:AuthenticationStatement
                AuthenticationInstant="2010-09-07T14:28:35.398Z"
                AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:X509-PKI">

                <saml:Subject>
                        <saml:NameIdentifier
                                Format="urn:oasis:names:tc:SAML:1.1:nameidformat:X509SubjectName">
                                CN=dg, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState, C=US
                        </saml:NameIdentifier>

                        <saml:SubjectConfirmation>

                                <saml:ConfirmationMethod>
                                        urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
                                </saml:ConfirmationMethod>
```

```
                        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                                <ds:X509Data>
                                        <ds:X509Certificate>
                                                MIIB4zC...Sw==
                                        </ds:X509Certificate>
                                </ds:X509Data>
                        </ds:KeyInfo>
                </saml:SubjectConfirmation>
        </saml:Subject>
</saml:AuthenticationStatement>

<saml:AttributeStatement>
        <saml:Subject>
                <saml:NameIdentifier
                        Format="urn:oasis:names:tc:SAML:1.1:nameidformat:X509SubjectName">
                        CN=dg, OU=FOR TESTING ONLY, O=MyOrganization, L=MyTown, ST=MyState, C=US
                </saml:NameIdentifier>
                <saml:SubjectConfirmation>
                        <saml:ConfirmationMethod>
                                urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
                        </saml:ConfirmationMethod>
                        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                                <ds:X509Data>
                                        <ds:X509Certificate>
                                                MIIB...Sw==
                                        </ds:X509Certificate>
                                </ds:X509Data>
                        </ds:KeyInfo>
                </saml:SubjectConfirmation>
        </saml:Subject>

        <saml:Attribute
                AttributeName="urn:be:smals:um:entity:quality"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>QUAL SP LEG</saml:AttributeValue>
        </saml:Attribute>

        <saml:Attribute
                AttributeName="urn:be:smals:env:attribute-authority"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>NOSS</saml:AttributeValue>
        </saml:Attribute>

        <saml:Attribute
                AttributeName="urn:be:smals:um:entity:quality:external-id"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>sts@E3@QUAL_SP_LEG</saml:AttributeValue>
        </saml:Attribute>

        <saml:Attribute
                AttributeName="urn:be:smals:expeditor:number"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>001221</saml:AttributeValue>
        </saml:Attribute>

        <saml:Attribute
                AttributeName="urn:be:smals:env:authentication-level"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>30</saml:AttributeValue>
        </saml:Attribute>

        <saml:Attribute
                AttributeName="urn:be:smals:env:user-type"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>ENTERPRISE</saml:AttributeValue>
        </saml:Attribute>

        <saml:Attribute
                AttributeName="urn:be:fgov:kbo-bce:organization:cbe-number"
                AttributeNamespace="urn:be:fgov:identification-namespace"
                xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
                <saml:AttributeValue>999124003</saml:AttributeValue>
        </saml:Attribute>
```

```
        </saml: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/2000/09/xmldsig-more#rsa-
sha256"/>
                        <dsig:Reference URI="#ID_d968adf1-bd09-48aa-b19a-5c6aca32ad9c">
                                <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/2000/09/xmlenc#sha256"/>
                                <dsig:DigestValue>
                                        T0jOnZBw4CJCc0qVwWYu2d04+VQ=
                                </dsig:DigestValue>
                        </dsig:Reference>
                </dsig:SignedInfo>
                <dsig:SignatureValue>
                        d+Q5cCAGpB2+wlqHl8kNKGA4Eao1GKYBh1YPJb9JwU8dUlQV+PxDW0CPoe/3PYhL3u6pxyPePWQCZlcrL
                        WqrhYSh8scsOyfWJQaeNjQ+A8oKBMB0TvOX1f8Jyaof4jazW0ndcXaUc8t24dVBgrJINJP98Cuz+nZJqT
                        0FUuH+qrA=
                </dsig:SignatureValue>
        </dsig:Signature>

</saml:Assertion>
```

### 8.1.4 Error codes

See chapter 9.

## 8.2 Operation RequestSecurityToken2

### 8.2.1 Functional description

Operation to issue a collection of security tokens. Currently not supported.

### 8.2.2 Request message construction

Not supported

### 8.2.3 Reply message interpretation

Not supported

### 8.2.4 Error codes

Will always return a SOAP fault with faultcode **wst:BadRequest.**

# 9 Common error codes

Error codes originating from OASIS: [http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html#_Toc162064994](http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html#_Toc162064994)

The error codes listed here define possible business errors whilst processing the request. These error codes indicate a problem in the content of the request, including format errors.

| Error code | Description |
|---|---|
| **wst:InvalidRequest** | The request was invalid or malformed |
| **wst:FailedAuthentication** | Authentication failed |
| **wst:RequestFailed** | The specified request failed |
| **wst:InvalidSecurityToken** | Security token has been revoked |
| **wst:AuthenticationBadElements** | Insufficient Digest Elements |
| **wst:BadRequest** | The specified RequestSecurityToken is not understood. |
| **wst:ExpiredData** | The request data is out-of-date |
| **wst:InvalidTimeRange** | The requested time range is invalid or unsupported |
| **wst:InvalidScope** | The request scope is invalid or unsupported |
| **wst:RenewNeeded** | A renewable security token has expired |
| **wst:UnableToRenew** | The requested renewal failed |

# 10 Security

Service security is following the common standards:

- For authentication, a X.509 certificate applies. The certificate contains the identifier of the caller.
- For transport security, one way SSL applies.
- For integrity and authentication of the message, a signature of the timestamp, body and binary security token is required.

The following WS-Security policy applies to all inbound requests. The outbound responses have no security header. The response contains an already signed SAML assertion.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsp:Policy xmlns:sp="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702"
xmlns:xsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
xsu:Id="Wssp1.2-X.509-Wss1.0">
  <sp:AsymmetricBinding>
    <wsp:Policy>
      <sp:InitiatorToken>
        <wsp:Policy>
          <sp:X509Token
sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/AlwaysToRecipient">
            <wsp:Policy>
              <sp:WssX509V3Token10/>
            </wsp:Policy>
          </sp:X509Token>
        </wsp:Policy>
      </sp:InitiatorToken>
      <sp:RecipientToken>
        <wsp:Policy>
          <sp:X509Token
sp:IncludeToken="http://docs.oasis-open.org/ws-sx/ws-securitypolicy/200702/IncludeToken/Never">
            <wsp:Policy>
              <sp:WssX509V3Token10/>
            </wsp:Policy>
          </sp:X509Token>
        </wsp:Policy>
      </sp:RecipientToken>
      <sp:AlgorithmSuite>
        <wsp:Policy>
          <sp:Basic256/>
        </wsp:Policy>
      </sp:AlgorithmSuite>
      <sp:Layout>
        <wsp:Policy>
          <sp:Lax/>
        </wsp:Policy>
```

```
      </sp:Layout>
      <sp:IncludeTimestamp/>
      <sp:ProtectTokens/>
      <sp:OnlySignEntireHeadersAndBody/>
    </wsp:Policy>
  </sp:AsymmetricBinding>
  <sp:Wss10>
    <wsp:Policy>
      <sp:MustSupportRefKeyIdentifier/>
      <sp:MustSupportRefIssuerSerial/>
    </wsp:Policy>
  </sp:Wss10>
  <sp:SignedParts>
    <sp:Body/>
  </sp:SignedParts>
</wsp:Policy>
```

# 11 Annex