

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

Service Specification:

Integration and use of the service

Citizen profile

Version 1.1

To the attention of: "IT expert" aiming to integrate this Service

Table of Contents

1	Goal of the service	3
1.1	Description of the service.....	3
1.2	Service consumers.....	3
2	Document management.....	3
2.1	Document history	3
2.2	Validation	3
2.3	Document references.....	3
2.4	Goal of the document.....	4
3	Service history.....	4
4	Global overview of the Service Group.....	4
5	Prerequisites	5
5.1	Business prerequisites	5
5.2	Technical prerequisites	5
6	Test, release and exploitation procedures	5
7	Description of the service operations	6
7.1	WSDL.....	6
7.2	Operation: CreateOrUpdateProfile.....	6
7.3	Operation: <i>GetCitizenProfiles</i>	9
7.4	Operation: <i>GetRestrictedCitizenProfiles</i>	11
8	<i>Description of the common types</i>	14
8.1	Simple types.....	14
8.2	complexTypes.....	16
9	Common error codes.....	19
10	Annex.....	19

1 Goal of the service

1.1 Description of the service

Citizen profile is a webapp allowing citizens to create a profile. This profile is mainly used by e-Box citizen in order to get informations over the citizen such as the e-mail address.

It is composed of 3 operations :

- CreateOrUpdateCitizenProfile : used by the WA to create or update citizen profiles (used by citizens)
- GetCitizenProfile : used by e-Box in order to retrieve all the informations of citizen profiles
- GetRestrictedCitizenProfile : used by institutions to get a citizen profile without e-mail address and Phone number (sensitive data)

1.2 Service consumers

Citizens are the main target of this service but it can also be used by institutions in order to get restricted information about the citizens.

It can be accessed to these addresses:

INT : <https://professionalservices-int.socialsecurity.be/CitizenProfileVarA/v1>

ACC : <https://professionalservices-acpt.socialsecurity.be/CitizenProfileVarA/v1>

2 Document management

2.1 Document history

Version	Date	Author	Description of changes / remarks
0.1	23/10/2013	M. Janmart	Initial version

2.2 Validation

Reviewers	Name	Version Reviewed	Remarks

2.3 Document references

OASIS WS-Security standards specification :

https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss

OASIS Standard X.509 Certificate Token Profile:

<http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf>

2.4 Goal of the document

This document provides functional and technical information on calling the Service Citizen Profile. This information should allow (the IT department of) an organization to integrate and use the service.

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.

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 CBSS (Crossroad Bank for Social Security).

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: NONE

4 Global overview of the Service Group

Not applicable

5 Prerequisites

5.1 Business prerequisites

The WS Citizen profile could only be used by **institutions** that are eager to publish document for citizens in e-Box citizen. The list of these institutions is included in the *Dossier Unique* of the WS CitizenProfile. If a new institution wants to use the WS, the *Dossier Unique* must be updated.

A SOAP request to the WS CitizenProfile must be signed by means of a *X.509 certificate* (cf. Technical prerequisites, below). The WS CitizenProfile identifies the message sender on the basis of the KBO in this certificate.

5.2 Technical prerequisites

To call the WS you must sign your soap request with a Fedict X509 certificate.
This certificate identifies the institution and the application.

X509	Value	Comment
C	BE	-
O	FEDERAL GOVERNMENT	-
OU	<i>Institution Name</i>	Example OU=National Social Security Office
OU	OU=urn:be:fgov:kbo-bce:organization:cbe-number:\${CBE}	Example for institution with CBE 1234567890 this OU must be OU=urn:be:fgov:kbo-bce:organization:cbe-number:1234567890
CN	<i>Application identifier</i>	-

5.2.1 Security

Transport security

- For transport security one way SSL applies. The SSL connection will provide encryption to the data transported, as well as server authentication of the service provider.

Message security

- Message security using the OASIS WS-Security standards provides authentication of the service consumer and protects the message integrity end-to-end. See section 2.3 for links to the specifications.
- **Direct authentication (OASIS Standard X.509 Certificate Token Profile):** The consumer includes its X.509 certificate (wsse:BinarySecurityToken) and a timestamp (wsu:Timestamp) within the wsse:Security header. The consumer signs the entire SOAP body, timestamp and BST headers.

6 Test, release and exploitation procedures

/

7 Description of the service operations

7.1 WSDL

WSDL and XSD are available on:

7.2 Operation: CreateOrUpdateProfile

element **CreateOrUpdateCitizenProfileRequest**

diagram	
namespace	http://socialsecurity.be/citizenprofile/v1
type	CreateOrUpdateProfileRequestType
properties	content complex
children	INSS ContactInfo Flags
source	<code><xs:element name="CreateOrUpdateCitizenProfileRequest" type="core:CreateOrUpdateProfileRequestType"/></code>

7.2.1

Explanations:

This operation is used by the webapp whenever a citizen updates or creates a profile.

It has several parameters

- INSS: this is the SSIN (Social Security Identification Number) of the user
- ContactInfo: the information over a citizen (language chosen(mandatory), e-mail address, phone number)
- Flags: the two flags the user can choose when he creates a profile (Conditions accepted, and choose to receive documents rather through e-Box than through paper mail).

Example 1:

Minimal version of a message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/citizenprofile/v1"
xmlns:v11="http://socialsecurity.be/citizenprofile/core/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:CreateOrUpdateCitizenProfileRequest>
      <v11:INSS>999999999979</v11:INSS>
      <v11:ContactInfo>
        <v11:LanguageCode>FR</v11:LanguageCode>
        <!--Optional:-->
        <v11:EmailAddress>test@test.com</v11:EmailAddress>
        <!--Optional:-->
      </v11:ContactInfo>
      <v11:Flags>
        <v11:ConditionsAccepted>true</v11:ConditionsAccepted>
      </v11:Flags>
    </v1:CreateOrUpdateCitizenProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 2:

Complete version of a message:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/citizenprofile/v1"
xmlns:v11="http://socialsecurity.be/citizenprofile/core/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:CreateOrUpdateCitizenProfileRequest>
      <v11:INSS>999999999979</v11:INSS>
      <v11:ContactInfo>
        <v11:LanguageCode>FR</v11:LanguageCode>
        <!--Optional:-->
        <v11:EmailAddress>test@test.com</v11:EmailAddress>
        <!--Optional:-->
        <v11:PhoneNbr>027870000</v11:PhoneNbr>
      </v11:ContactInfo>
      <v11:Flags>
        <v11:ConditionsAccepted>true</v11:ConditionsAccepted>
        <!--Optional:-->
        <v11:ExclusiveEbox>true</v11:ExclusiveEbox>
      </v11:Flags>
    </v1:CreateOrUpdateCitizenProfileRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.2.2 Reply message

element CreateOrUpdateCitizenProfileResponse

diagram	
namespace	http://socialsecurity.be/citizenprofile/v1
type	CreateOrUpdateProfileResponseType
properties	content complex
children	NISS Status
source	<pre><xs:element name="CreateOrUpdateCitizenProfileResponse" type="core:CreateOrUpdateProfileResponseType"/></pre>

Explanations :

The response is a status of whether the update has been performed or not. The NISS is the NISS of the user who has been modified.

Example :

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:CreateOrUpdateCitizenProfileResponse xmlns="http://socialsecurity.be/citizenprofile/core/v1"
xmlns:ns2="http://socialsecurity.be/errors/v1"
xmlns:ns3="http://socialsecurity.be/errors/serviceprovider/v1"
xmlns:ns4="http://socialsecurity.be/citizenprofile/v1">
      <NISS xsi:type="xs:string" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">99999999979</NISS>
      <Status>ProfileUpdated</Status>
    </ns4:CreateOrUpdateCitizenProfileResponse>
  </S:Body>
</S:Envelope>
```


7.3 Operation: *GetCitizenProfiles*

7.3.1 Request message

element **GetCitizenProfilesRequest**

diagram	
namespace	http://socialsecurity.be/citizenprofile/v1
type	GetProfileRequestType
properties	content complex

Explanation :

The request is simply composed by one or many INSS (1-1000)

Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/citizenprofile/v1"
xmlns:v11="http://socialsecurity.be/citizenprofile/core/v1">
  <soapenv:Header/>
  <soapenv:Body wsu:id="id-3" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd">
    <v1:GetCitizenProfilesRequest>
      <v11:INSS>00000000001</v11:INSS>
      ...
      <v11:INSS>999999999979</v11:INSS>
    </v1:GetCitizenProfilesRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.3.2 Response message

element **GetCitizenProfilesResponse**

diagram	
namespace	http://socialsecurity.be/citizenprofile/v1
type	GetProfileResponseType
properties	content complex
children	CitizenProfile
source	<xs:element name="GetCitizenProfilesResponse" type="core:GetProfileResponseType"/>

Explanation:

The response message contains either a complete citizen profile or a status message in case the NISS is incorrect / doesn't exist. This response is multiplied by the amount of SSIN put as input.

A citizen profile is composed of the following values detailed in the end of the document :

- NISS : the SSIN of the citizen
- Contact info : the language the citizen chose, his e-mail address and his phone number(not mandatory)
- Flags : wether the user has accepted or not the terms and conditions (should be set to "true") and wether the user rather has messages sent in his e-box rather than in his physical mail.
- LastValidationDate : the last time the user updated his profile

Example (for 2 SSIN):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetCitizenProfilesResponse xmlns="http://socialsecurity.be/citizenprofile/core/v1"
  xmlns:ns2="http://socialsecurity.be/errors/v1"
  xmlns:ns3="http://socialsecurity.be/errors/serviceprovider/v1"
  xmlns:ns4="http://socialsecurity.be/citizenprofile/v1">
      <CitizenProfile>
        <INSS>99999999979</INSS>
        <ContactInfo>
          <LanguageCode>FR</LanguageCode>
          <EmailAddress>ssin1@test.com</EmailAddress>
          <PhoneNbr>02/787.00.00</PhoneNbr>
        </ContactInfo>
        <Flags>
          <ConditionsAccepted>true</ConditionsAccepted>
          <ExclusiveEbox>true</ExclusiveEbox>
        </Flags>
        <LastValidationDate>2013-10-22T15:38:59.680</LastValidationDate>
      </CitizenProfile>
      <CitizenProfile>
        <INSS>00000000007</INSS>
        <ContactInfo>
          <LanguageCode>DE</LanguageCode>
          <EmailAddress>ssin2@test.com</EmailAddress>
          <PhoneNbr>0472000000</PhoneNbr>
        </ContactInfo>
        <Flags>
          <ConditionsAccepted>true</ConditionsAccepted>
          <ExclusiveEbox>>false</ExclusiveEbox>
        </Flags>
        <LastValidationDate>2012-10-03T16:48:32.304</LastValidationDate>
      </CitizenProfile>
    </ns4:GetCitizenProfilesResponse>
  </S:Body>
</S:Envelope>
```

7.4 Operation: *GetRestrictedCitizenProfiles*

7.4.1 Request message

element **GetRestrictedCitizenProfilesRequest**

diagram	
namespace	http://socialsecurity.be/citizenprofile/v1
type	GetRestrictedProfileRequestType
properties	content complex
children	INSS
source	<code><xs:element name="GetRestrictedCitizenProfilesRequest" type="core:GetRestrictedProfileRequestType"/></code>

Explanation :

The request is simply composed by one or many NISS (1-1000)

Example :

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/citizenprofile/v1"
xmlns:v11="http://socialsecurity.be/citizenprofile/core/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetRestrictedCitizenProfilesRequest>
      <v11:INSS>999999999907</v11:INSS>
      ...
      <v11:INSS>000000000007</v11:INSS>
    </v1:GetRestrictedCitizenProfilesRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

7.4.2 Response message

element **GetRestrictedCitizenProfilesResponse**

diagram	
namespace	http://socialsecurity.be/citizenprofile/v1
type	GetRestrictedProfileResponseType
properties	content complex
children	CitizenProfile
source	<code><xs:element name="GetRestrictedCitizenProfilesResponse" type="core:GetRestrictedProfileResponseType"/></code>

Explanation:

The response message contains either a restricted citizen profile or a status message in case the NISS is incorrect / doesn't exist. This response is multiplied by the amount of SSIN put as input.

A citizen profile is composed of the following values detailed in the end of the document :

- NISS : the SSIN of the citizen
- Contact info : the language the citizen chose
- Flags : whether the user has accepted or not the terms and conditions (should be set to "true") and whether the user rather has messages sent in his e-box rather than in his physical mail.
- LastValidationDate : the last time the user updated his profile

Example (for 2 SSIN):

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns4:GetCitizenProfilesResponse xmlns="http://socialsecurity.be/citizenprofile/core/v1"
      xmlns:ns2="http://socialsecurity.be/errors/v1"
      xmlns:ns3="http://socialsecurity.be/errors/serviceprovider/v1"
      xmlns:ns4="http://socialsecurity.be/citizenprofile/v1">
      <CitizenProfile>
        <INSS>99999999979</INSS>
        <ContactInfo>
          <LanguageCode>FR</LanguageCode>
        </ContactInfo>
        <Flags>
          <ConditionsAccepted>true</ConditionsAccepted>
          <ExclusiveEbox>true</ExclusiveEbox>
        </Flags>
        <LastValidationDate>2013-10-22T15:38:59.680</LastValidationDate>
      </CitizenProfile>
      <CitizenProfile>
        <INSS>00000000007</INSS>
        <ContactInfo>
          <LanguageCode>DE</LanguageCode>
        </ContactInfo>
        <Flags>
          <ConditionsAccepted>true</ConditionsAccepted>
          <ExclusiveEbox>false</ExclusiveEbox>
        </Flags>
        <LastValidationDate>2012-10-03T16:48:32.304</LastValidationDate>
      </CitizenProfile>
    </ns4:GetCitizenProfilesResponse>
  </S:Body>
</S:Envelope>
```

8 Description of the common types

This section presents the simple and complex types used to describe the message structure.

8.1 Simple types

simpleType **DateValueType**

namespace	http://socialsecurity.be/citizenprofile/core/v1									
type	restriction of xs:string									
properties	base xs:string									
facets	<table border="0"> <tr> <td>Kind</td> <td>Value</td> <td>Annotation</td> </tr> <tr> <td>minLength</td> <td>5</td> <td></td> </tr> <tr> <td>maxLength</td> <td>8</td> <td></td> </tr> </table>	Kind	Value	Annotation	minLength	5		maxLength	8	
Kind	Value	Annotation								
minLength	5									
maxLength	8									
annotation	documentation date format (between 5 and 8 char).									
source	<pre> <xs:simpleType name="DateValueType"> <xs:annotation> <xs:documentation>date format (between 5 and 8 char).</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:minLength value="5"/> <xs:maxLength value="8"/> </xs:restriction> </xs:simpleType> </pre>									

simpleType **EmailAdressType**

namespace	http://socialsecurity.be/citizenprofile/core/v1									
type	restriction of xs:string									
properties	base xs:string									
used by	element CitizenContactInfoType/EmailAddress									
facets	<table border="0"> <tr> <td>Kind</td> <td>Value</td> <td>Annotation</td> </tr> <tr> <td>minLength</td> <td>0</td> <td></td> </tr> <tr> <td>maxLength</td> <td>60</td> <td></td> </tr> </table>	Kind	Value	Annotation	minLength	0		maxLength	60	
Kind	Value	Annotation								
minLength	0									
maxLength	60									
annotation	documentation Type for the citizen email address.									
source	<pre> <xs:simpleType name="EmailAdressType"> <xs:annotation> <xs:documentation>Type for the citizen email address.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:maxLength value="60"/> <xs:minLength value="0"/> </xs:restriction> </xs:simpleType> </pre>									

simpleType **INSSType**

namespace	http://socialsecurity.be/citizenprofile/core/v1
type	restriction of xs:string
properties	base xs:string
used by	elements CreateOrUpdateProfileRequestType/INSS GetProfileRequestType/INSS GetRestrictedProfileRequestType/INSS CitizenProfileType/INSS RestrictedCitizenProfileType/INSS
facets	Kind Value Annotation pattern \d{11}
annotation	documentation Citizen national number
source	<pre><xs:simpleType name="INSSType"> <xs:annotation> <xs:documentation>Citizen national number</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:pattern value="\d{11}"/> </xs:restriction> </xs:simpleType></pre>

simpleType **LanguageType**

namespace	http://socialsecurity.be/citizenprofile/core/v1
type	restriction of xs:string
properties	base xs:string
used by	elements CitizenContactInfoType/LanguageCode RestrictedCitizenContactInfoType/LanguageCode
facets	Kind Value Annotation enumeration FR enumeration NL enumeration DE enumeration EN
annotation	documentation Type for the language.
source	<pre><xs:simpleType name="LanguageType"> <xs:annotation> <xs:documentation>Type for the language.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:enumeration value="FR"/> <xs:enumeration value="NL"/> <xs:enumeration value="DE"/> <xs:enumeration value="EN"/> </xs:restriction> </xs:simpleType></pre>

simpleType **PhoneNbrType**

namespace	http://socialsecurity.be/citizenprofile/core/v1
type	restriction of xs:string

properties	base <code>xs:string</code>
used by	element CitizenContactInfoType/PhoneNbr
facets	Kind Value Annotation minLength 1 maxLength 20
annotation	documentation Type for the citizen's phone nbr.
source	<pre><xs:simpleType name="PhoneNbrType"> <xs:annotation> <xs:documentation>Type for the citizen's phone nbr.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:minLength value="1"/> <xs:maxLength value="20"/> </xs:restriction> </xs:simpleType></pre>

simpleType StatusType

namespace	<code>http://socialsecurity.be/citizenprofile/core/v1</code>
type	restriction of <code>xs:string</code>
properties	base <code>xs:string</code>
used by	elements CitizenProfileType/Status RestrictedCitizenProfileType/Status
facets	Kind Value Annotation enumeration ProfileNotFound
annotation	documentation Status of an action on a NISS
source	<pre><xs:simpleType name="StatusType"> <xs:annotation> <xs:documentation>Status of an action on a NISS</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"> <xs:enumeration value="ProfileNotFound"/> </xs:restriction> </xs:simpleType></pre>

8.2 complexTypes

complexType citizenprofile/CitizenContactInfoType

diagram	
namespace	<code>http://socialsecurity.be/citizenprofile/core/v1</code>
children	LanguageCode EmailAddress PhoneNbr
used by	elements CreateOrUpdateProfileRequestType/ContactInfo CitizenProfileType/ContactInfo
annotation	documentation

	Identifiers of the citizen requested
source	<pre> <xs:complexType name="CitizenContactInfoType"> <xs:annotation> <xs:documentation>Identifiers of the citizen requested</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="LanguageCode" type="LanguageType" minOccurs="1"/> <xs:element name="EmailAddress" type="EmailAdressType" minOccurs="0"/> <xs:element name="PhoneNbr" type="PhoneNbrType" minOccurs="0"/> </xs:sequence> </xs:complexType> </pre>

complexType RestrictedCitizenContactInfoType

diagram	
namespace	http://socialsecurity.be/citizenprofile/core/v1
children	LanguageCode
used by	element RestrictedCitizenProfileType/ContactInfo
annotation	documentation Identifiers of the citizen requested
source	<pre> <xs:complexType name="RestrictedCitizenContactInfoType"> <xs:annotation> <xs:documentation>Identifiers of the citizen requested</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="LanguageCode" type="LanguageType" minOccurs="1"/> </xs:sequence> </xs:complexType> </pre>

complexType CitizenFlagsType

diagram	
namespace	http://socialsecurity.be/citizenprofile/core/v1
children	ConditionsAccepted ExclusiveEbox
used by	elements CreateOrUpdateProfileRequestType/Flags CitizenProfileType/Flags RestrictedCitizenProfileType/Flags
annotation	documentation Identifiers of the citizen requested
source	<pre> <xs:complexType name="CitizenFlagsType"> <xs:annotation> <xs:documentation>Identifiers of the citizen requested</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="ConditionsAccepted" type="xs:boolean" minOccurs="1"/> <xs:element name="ExclusiveEbox" type="xs:boolean" minOccurs="0"/> </xs:sequence> </xs:complexType> </pre>

complexType CitizenProfileType

diagram	<p>The diagram illustrates the structure of the CitizenProfileType complex type. It consists of a sequence container that includes an INSS element and a choice container. The choice container has two options: a sequence of ContactInfo, Flags, and LastValidationDate elements, and a Status element.</p>
namespace	http://socialsecurity.be/citizenprofile/core/v1
children	INSS ContactInfo Flags LastValidationDate Status
used by	element GetProfileResponseType/CitizenProfile
annotation	documentation Complexe type for the global citizen profile type.
source	<pre> <xs:complexType name="CitizenProfileType"> <xs:annotation> <xs:documentation>Complexe type for the global citizen profile type.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="INSS" type="INSSType"/> <xs:choice> <xs:sequence> <xs:element name="ContactInfo" type="CitizenContactInfoType"/> <xs:element name="Flags" type="CitizenFlagsType"/> <xs:element name="LastValidationDate" type="xs:dateTime"/> </xs:sequence> <xs:element name="Status" type="StatusType"/> </xs:choice> </xs:sequence> </xs:complexType> </pre>

9 Common error codes

System Error Codes:

These error codes indicate a problem in the request sent or a system error.

Code	Description	Cause	Explication
SOA-00001	Service error		This is the default error sent to the consumer in case no more details are known.
SOA-01001	Service call not authenticated	Consumer	From the security information provided, either the consumer could not be identified or the credentials provided are not correct.
SOA-01002	Service call not authorized	Consumer	The consumer is identified and authenticated, but is not allowed to call the given service.
SOA-02001	Service not available. Please contact service desk.	Provider	An unexpected error has occurred. Retries will not work. Service desk may help with root cause analysis.
SOA-02002	Service temporarily not available. Please try later.	Provider	An unexpected error has occurred. Retries should work. If the problem persists service desk may help.
SOA-03001	Malformed message	Consumer	This is the default error for content related errors in case no more details are known.
SOA-03002	Message must be SOAP	Consumer	Message does not respect the SOAP standard.
SOA-03003	Message must contain SOAP body	Consumer	Message respects the SOAP standard, but body is missing.
SOA-03004	WS-I compliance failure	Consumer	Message does not respect the WS-I standard.
SOA-03005	WSDL compliance failure	Consumer	Message is not compliant with WSDL.
SOA-03006	XSD compliance failure	Consumer	Message is not compliant with XSD.
SOA-03007	Message content validation failure	Consumer	From the message content (conform XSD): extended checks on the element format failed or cross-checks between fields failed.

10 Annex

/