



**Social Security
ONSS/RSZ**

**Place Victor Horta 11 - 1060 BRUXELLES - Tél 02 509 31 11 -
Fax 02 509 30 19 - Internet: www.onss.fgov.be**

**Victor Hortaplein 11 -1060 BRUSSEL - Tel. 02 509 31 11 -
Fax 02 509 30 19 - Internet: www.rsz.fgov.be**

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

Service Specification:

Integration and use of the service

**EboxMessage
Version 1.5 (*Validated Document*)**

into an external application

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

TABLE OF CONTENTS

1. GOAL OF THE SERVICE	6
1.1 e-Box, from the point of view of institutions – applications	6
1.2 e-Box for Enterprises, Professionals, Citizens	6
2. DOCUMENT MANAGEMENT	8
2.1 Document goal	8
2.2 Document history	8
2.3 Document references	9
3. SERVICE HISTORY	9
4. PREREQUISITES	10
4.1 Business prerequisites	10
4.2 Technical prerequisites	11
4.2.1 Security	11
5. DESCRIPTION OF THE SERVICE OPERATIONS	11
5.1 Operation commitLinkEboxMessage	12
5.1.1 Request message construction	12
5.1.1.1 Example	12
5.1.2 Reply message interpretation	12
5.1.2.1 Example	12
5.1.3 Error codes	12
5.2 Operation getEboxInfo	12
5.2.1 Request message construction	12
5.2.1.1 Example	13
5.2.2 Reply message interpretation	13
5.2.2.1 Example	13
5.2.3 Error codes	13
5.3 Operation getMessageStatus	13
5.3.1 Request message construction	13
5.3.1.1 Example	13
5.3.2 Reply message interpretation	14
5.3.2.1 Example	14
5.3.3 Error codes	14
5.4 Operation hasAnEbox	14
5.4.1 Request message construction	14
5.4.1.1 Example	15
5.4.2 Reply message interpretation	15
5.4.2.1 Example	15
5.4.3 Error codes	16
5.5 Operation healthCheck	16
5.5.1 Request message construction	16
5.5.1.1 Example	16
5.5.2 Reply message interpretation	16
5.5.2.1 Example	17
5.5.3 Error codes	17
5.6 Operation linkEboxMessage	17
5.6.1 Request message construction	17
5.6.1.1 Example	17
5.6.2 Reply message interpretation	17
5.6.2.1 Example	18

5.6.3	Error codes	18
5.7	Operation publishEboxMessage	18
5.7.1	Request message construction	18
5.7.1.1	Example	21
5.7.2	Reply message interpretation	22
5.7.2.1	Example	22
5.7.3	Error codes	22
5.8	Operation storeEboxMessage	22
5.8.1	Request message construction	23
5.8.1.1	Example	23
5.8.2	Reply message interpretation	23
5.8.2.1	Example	23
5.8.3	Error codes	23
6.	COMMON TYPES	23
6.1	EboxMessage_types_v1	24
6.1.1	Application	24
6.1.2	CBE	24
6.1.3	Denomination	24
6.1.4	DenominationValue	24
6.1.5	Digest	25
6.1.6	DigestMethod	25
6.1.7	DigestValue	25
6.1.8	EboxEntityQuality	25
6.1.9	EboxInfo	26
6.1.10	Email	26
6.1.11	EntityID	27
6.1.12	ErrorStatusCode	27
6.1.13	File	27
6.1.14	FileName	28
6.1.15	FreeInformation	28
6.1.16	FreeText	28
6.1.17	GetMessageStatusErrorType	29
6.1.18	GetMessageStatusResponseType	29
6.1.19	GetMessageStatusType	29
6.1.20	HasAnEboxResponseType	30
6.1.21	INSS	30
6.1.22	Language	30
6.1.23	LeftCell	30
6.1.24	MessageID	31
6.1.25	MessageType	31
6.1.26	Meta	31
6.1.27	MIME	32
6.1.28	Name	32
6.1.29	NIHII	32
6.1.30	Person	33
6.1.31	QualityCode	33
6.1.32	RecipientID	33
6.1.33	RightCell	34
6.1.34	Row	34
6.1.35	SenderID	34
6.1.36	Sequence	35
6.1.37	ShortString	35
6.1.38	ShortTitle	35

6.1.39	SubentityId	35
6.1.40	Table	36
6.1.41	Ticket	36
6.1.42	Title	36
6.2	Monitoring-v1	36
6.2.1	Component	37
6.2.2	Environment	37
6.2.3	Extensions	37
6.2.4	HealthCheckType	38
6.2.5	Location	38
6.2.6	MavenReference	39
6.2.7	Resource	39
6.2.8	ResourceType	39
6.2.9	SanityCheck	40
6.2.10	Status	40
6.2.11	StatusLevel	41
7.	COMMON TYPES SUPPLEMENTARY INFORMATION	41
7.1	complexType: MessageID	41
7.2	complexType: RecipientID	42
7.3	complexType: File	42
7.4	complexType: FreeInformation	43
7.5	complexType Table	43
7.6	complexType: Meta	44
8.	COMMON ERROR CODES	45
8.1	System Error Codes	45
8.2	Business Error Codes	46
9.	ANNEX	47
9.1	WS EBoxMessage EndPoint URI	47
9.2	Common Quality Codes	47
9.3	Using the X.509 certificate	48
9.3.1	Dedicated certificate by sending application	48
9.3.2	Single certificate shared by several sending applications	48
9.4	Contact	49

1. GOAL OF THE SERVICE

Service that allows partners from the Social Security sector to send messages to the e-Box of enterprises, professionals or citizens. It also allows to verify the existence and activity of an e-Box Enterprise.

1.1 E-BOX, FROM THE POINT OF VIEW OF INSTITUTIONS – APPLICATIONS

e-Box allows institutions of **Belgian** Social Security to publish messages of different types for the attention of companies, professionals or citizens. The e-Box system provides a secured Web Service called WS EBoxMessage. This electronic channel is available for the Belgian Social Security Institutions, in order to communicate official documents to companies, professionals or citizens.

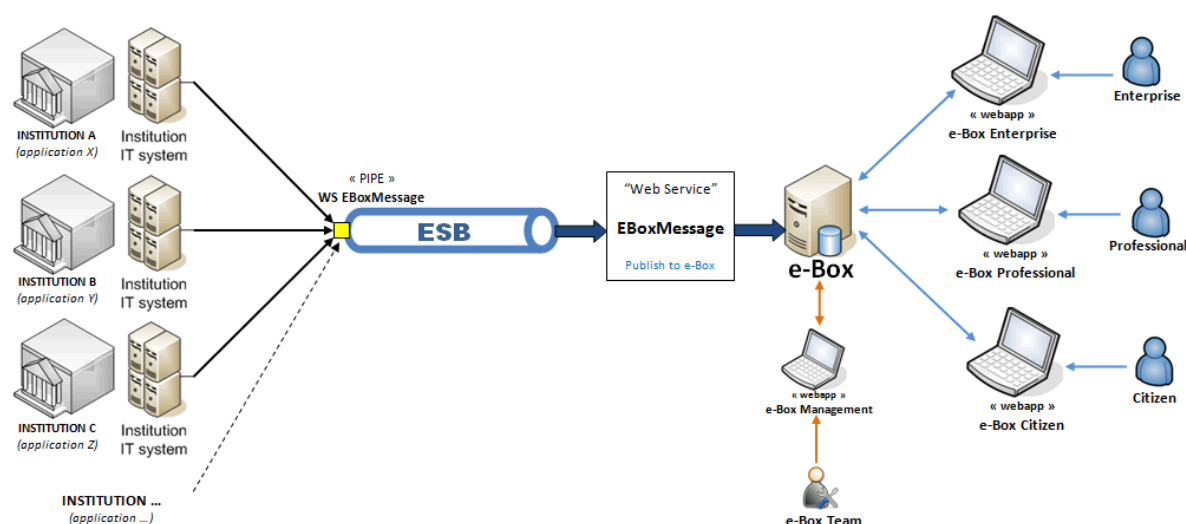


Figure 1 - WS EBoxMessage: overview

1.2 E-BOX FOR ENTERPRISES, PROFESSIONALS, CITIZENS

e-Box is an electronic box, secured and centralized, available as a web application. This protected space is reserved and guaranteed by the Social Security Institutions, for Enterprises, Professionals and Citizens.

- **e-Box Enterprise:** an e-Box is available for each enterprise registered on the Social Security Portal (authentic source = User Management Enterprise, NSSO sector). An e-Box exists for each “quality”¹ defined for the registered enterprises.

More info: https://www.socialsecurity.be/site_fr/general/helpcentre/ebox/index.htm

- **e-Box Professional “Institution”:** an e-Box is available for each institution registered in the authentic source “User Management Professional” (CBSS sector). An e-Box exists for each “quality”² defined for the registered institutions.

- **e-Box Professional “Personal”:** each civil servant of Belgian Social Security (each user registered below a professional quality in the “User Management Professional”) has access to a personal e-Box.
More info: https://www.socialsecurity.be/site_fr/civilservant/Applics/epp/index.htm

- **eBox Citizen:** an eBox is available for each citizen in Belgium.

More info: <https://www.mysocialsecurity.be/>

¹ A quality is a business characteristic of an organisation. Example of quality for a company: *Employeur ONSS/ Werkgever RSZ; Employeur ONSS-APL / Werkgever RSZPPO; Secrétariat Social Agréé/ erkende sociale secretariaat; Prestataire de service / dienstverrichter; ...*

² A “professional” quality is a business characteristic of an institution. Example of “professional” quality: *Service Administratif de l’institution../ Administratieve dienst...; Service Sécurité de l’institution... / Veiligheidsdienst...; Administration communale / Gemeente bestuur; Action sociale (CPAS)/ Maatschappelijk Welzijn (OCMW)...*

The screenshot shows the e-Box Enterprise web application interface. At the top, there are language options (DE, FR, NL) and a 'Contact' link. The user's name 'ANTOINE JEAN MOULART' and a 'Se déconnecter' link are visible. The e-Box logo is on the left. Below the header, there are navigation tabs: 'Tous' (selected), 'Documents', and 'Notifications'. A search bar with 'Recherche personnalisée' is on the right. The main content is a table of messages:

Date	Institution	Emetteur	Titre	A	
24/06/2014	ONSS	WIDE	Identification auprès de l'ONSS (démon)		Télécharger Détail
24/06/2014	ONSS	DIMONA	Notification Dimona (démon) Expire bientôt		Télécharger Détail
03/06/2014	ONSS	EDE	EDER501 - Ticket : 340002YV3K008 Recommandé		Télécharger Détail
16/05/2014	ONSS	MAHIS	Modification d'un mandat		Télécharger Détail
16/05/2014	ONSS	GOTOT GT	Detachement de Dominique Demonique (VOLKSKREDIET DE TOREN NV)		Télécharger Détail
16/05/2014	ONSS	GOTOT GT (old)	Detachement de Dominique Demonique (VOLKSKREDIET DE TOREN NV)		Télécharger Détail
16/05/2014	ONVA	COVA	Données de vacances (attestation définitive)		Télécharger Détail
16/05/2014	ONVA	COVA	Données de vacances		Télécharger Détail
16/05/2014	ONSS	MAHIS	Modification d'un mandat		Télécharger Détail
16/05/2014	ONSS	MAHIS	Création d'un mandat provisoire		Télécharger Détail

At the bottom, it shows 'Résultats 1 à 10 (11 résultats trouvés)' and a pagination control with '1' selected and '2' available, followed by 'Suivant'.

Figure 2 - e-Box Enterprise (preview of the web application)

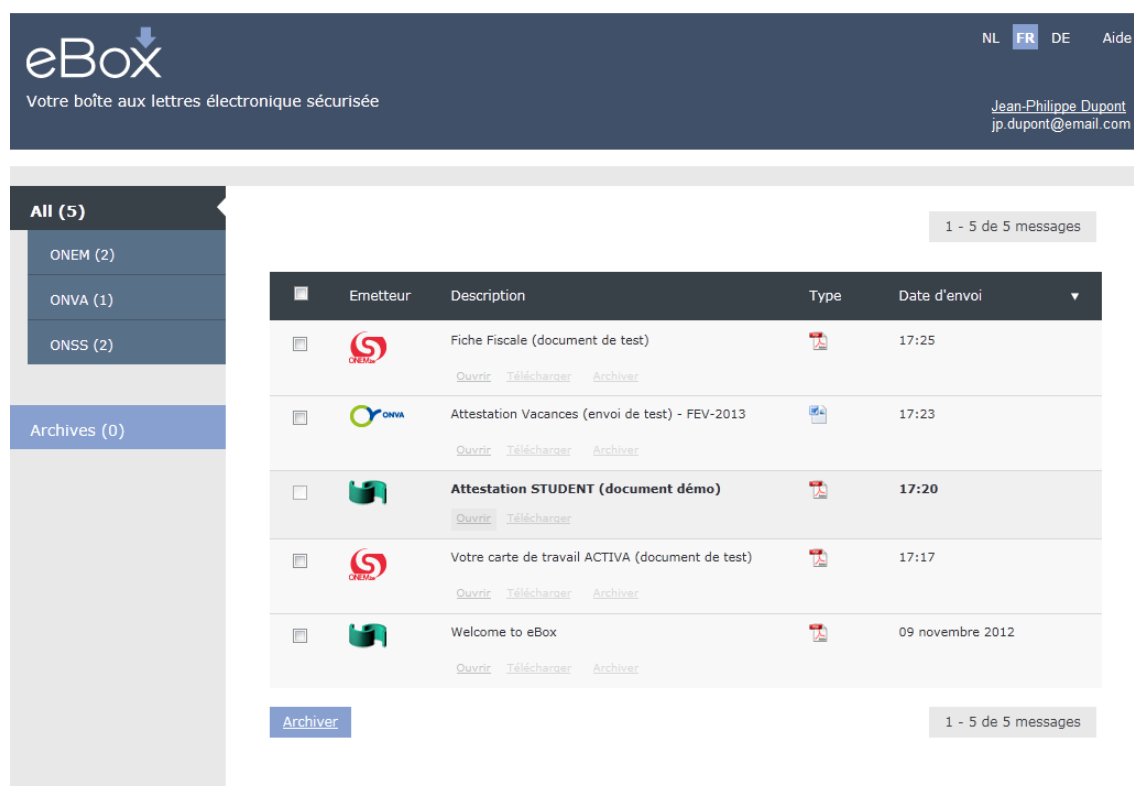


Figure 3 - eBox Citizen (preview of the web application)

2. DOCUMENT MANAGEMENT

2.1 DOCUMENT GOAL

This document provides functional and technical information on calling EboxMessage, as provided by the National Social Security Office (NSSO). 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 National Social Security Office (NSSO).

2.2 DOCUMENT HISTORY

Version	Date	Author	Description of changes / remarks
0.1	27/09/2011	A. Moulart	First version
0.2	21/09/2012	A. Moulart	New operations for broadcast publication
0.3	1/07/2014	A. Moulart	New operation "hasAnEbox"
0.4	18/03/2015	D. Thomas / P. Fustin	First version of this documentation generated based on WSDL/XSD version 1.4

0.5	23/06/2015	V. Adam / A. Clerbaut	Version of this documentation generated based on a new WSDL/XSD version 1.4
0.6	28/07/2015	V. Adam / A. Clerbaut	Version of this documentation generated based on a new WSDL/XSD version 1.5
0.7	9/11/2015	V. Adam / A. Clerbaut	Version of this documentation generated based on a new WSDL/XSD version 1.5

2.3 DOCUMENT REFERENCES

General information about the web services data exchange is available on the Social Security Portal:

https://www.socialsecurity.be/site_nl/general/helpcentre/soa/index.htm

https://www.socialsecurity.be/site_fr/general/helpcentre/soa/index.htm

Detailed information about this service is available on the Social Security Portal:

http://portalinfo.smals.be/fr/docs/Navigation_root/Portal_Knowledge/downloads.html?categ=08ebox#08ebox

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 release date (production): 12/06/2014

Previous version number (production): 1.3

List of changes:

05/07/2012 version 1.0: First version

16/08/2012 version 1.1: support URN as application ID

26/03/2013 version 1.2: support of certificates not dedicated to a single application

12/06/2014 version 1.3:

Add "recipient validation" in operation "publish EboxMessage"

Operation "hasAnEbox" added

24/06/2015 version 1.4:

Operation getMessageStatus added

EntityId modified with a choice (added) and SubentityId added in this choice

==> - GetEboxInfo Request/Response impacted

- HasAnEbox Request/Response impacted

28/07/2015 version.1.5:

simpleType name="SubentityId" regular expression changed

- ==>
- GetEboxInfoRequest/Response impacted
 - GetMessageStatusRequest/Response impacted
 - HasAnEboxRequest/Response impacted

Operation healthCheck added

4. PREREQUISITES

4.1 BUSINESS PREREQUISITES

Access to the service is granted as defined in the "Unique Dossier". This "Unique Dossier" centralizes all required information to authorize a new online application in accordance with the legal provisions on information security and data protection.

Here are the steps an institution must follow to publish for the first time to e-Box.

1. Gather information about e-Box, contact the e-Box Team and organize a meeting if necessary
2. Complete the e-Box form (available on <http://portalinfo.smals.be/>)
3. Asking for validation of the e-Box form (via eBoxIntegration@smals.be)
4. After form validation, integrate the WS, order the x.509 certificate
5. Testing (Integration, Acceptance)
6. Publish to e-Box (Production)

Some operations (*getEboxInfo*, *storeEboxMessage*, *linkEboxMessage*, *commitLinkEboxMessage*) appear in the WSDL but are restricted for internal use. These operations are not documented in this Service Specification Guide.

Each e-Box message is coupled to a **message type**.

A message type is associated with some essential pieces of information:

- "Application – institution" couple(s) that is (are) authorized to publish messages of this type.
- Validity period for messages of this type (the message is deleted from e-Box after its validity period has elapsed).
- Metadata linked to this message type (optional or mandatory + their format). Metadata can be used to add structured information about the message (for example: a NSSO number; a quarter date; a request date; etc.). Metadata can be used as search criteria among messages of a same type.
- Additional information about message updates; category...

When an institution wants to publish a new kind of message to e-Box, the message type must be defined with e-Box GA/TB (eBoxIntegration@smals.be) by means of an **identification form**.

This form is available on <http://portalinfo.smals.be/> .

Depending on the targeted recipients (Enterprises – Professionals – Citizens), an **agreement is required** coming from the owner institution (NSSO or CBSS).

4.2 TECHNICAL PREREQUISITES

The technical prerequisites are described in the Social Security UDDI:

<https://www.socialsecurity.be/web7/registrysocsec/uddi/bsc/web/browse/serviceDetail?serviceKey=uddi:be:fgov:rsz:eboxmessage:v1>

4.2.1 Security

The service has a single endpoint:

* be/socialsecurity/eboxmessage/v1/EboxMessage_v1.wsdl [X.509 Token Profile]

Duplicated WSDL files with explicit policy names

* be/socialsecurity/eboxmessage/v1/EboxMessage-X509_v1.wsdl [X.509 Token Profile]

A SOAP request to the WS EboxMessage must be signed by means of a Fedict X.509 certificate. The WS EboxMessage identifies the message sender on the basis of this 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>	-

Figure 4 - X.509 certificate: parameters to give in the certificate request

5. DESCRIPTION OF THE SERVICE OPERATIONS

The service consists of following operations :

Operations	Description
healthCheck	

getMessageStatus	Check if a message has been readed or downloaded
hasAnEbox	Verify if an e-Box exist and get the last connection date
commitLinkEboxMessage	Commit the link between a message and an e-Box (Internal use only)
linkEboxMessage	Link a message with an e-Box (Internal use only)
storeEboxMessage	Store a message in e-Box (Internal use only)
getEboxInfo	Get information needed for a notification (Internal use only)
publishEboxMessage	Send a message to an e-Box

5.1 OPERATION COMMITLINKeboxMESSAGE

5.1.1 Request message construction

Request of a CommitLinkEboxMessage: commit all link for the StoredEboxMessageID

sequence

<> [0..1] **SenderID** (SenderID)

<> **StoredEboxMessageID** (string)

5.1.1.1 Example

This paragraph is intentionally left blank.

5.1.2 Reply message interpretation

Response of a CommitLinkEboxMessageRequest

sequence

<> **StoredEboxMessageID** (string)

5.1.2.1 Example

This paragraph is intentionally left blank.

5.1.3 Error codes

This paragraph is intentionally left blank.

5.2 OPERATION GETeboxINFO

5.2.1 Request message construction

Request of a GetEboxInfo: for an e-Box list of a same quality.

sequence

<> **QualityCode** (QualityCode)

<> [1..2000] **EboxID** (EntityID)

5.2.1.1 Example

This paragraph is intentionally left blank.

5.2.2 Reply message interpretation

Response of a GetEboxInfo: for an e-Box list of a same quality.

sequence

<> **QualityCode** (QualityCode)

<> [0..2000] **EboxInfo** (EboxInfo)

<> [0..2000] **ErrorCase** (EntityID)

5.2.2.1 Example

This paragraph is intentionally left blank.

5.2.3 Error codes

This paragraph is intentionally left blank.

5.3 OPERATION GETMESSAGESTATUS

5.3.1 Request message construction

Request to know if a message has been readed or downloaded

sequence

<> **QualityCode** (QualityCode)

<> [1..250] **Message** (GetMessageStatusType)

5.3.1.1 Example

Minimal version of a message : 1 document

```
<v1:GetMessageStatusRequest>  
  <v11:QualityCode>CITIZEN</v11:QualityCode>  
  <!--1 to 250 repetitions:-->  
  <v11:Message>
```

```

    <v11:EboxID>
      <v11:INSS>12345678901</v11:INSS>
    </v11:EboxID>
    <v11:MessageID>
      <v11:Ticket>2015-04-23-003</v11:Ticket>
      <v11:Sequence>001</v11:Sequence>
    </v11:MessageID>
  </v11:Message>
</v1: GetMessageStatusRequest>

```

5.3.2 Reply message interpretation

Response to know if a message has been readed or downloaded

sequence

- <> **QualityCode** (QualityCode)

- <> [0..250] **Message** (GetMessageStatusResponseType)

- <> [0..250] **ErrorCase** (GetMessageStatusErrorType)

5.3.2.1 Example

```

<ns5: GetMessageStatusResponse xmlns:ns2="http://socialsecurity.be/eboxmessage/types/v1"
xmlns:ns3="http://socialsecurity.be/errors/v1" xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1"
xmlns:ns5="http://socialsecurity.be/eboxmessage/v1">
  <ns2:QualityCode>CITIZEN</ns2:QualityCode>
  <ns2:Message>
    <ns2:EboxID>
      <ns2:INSS>81041430930</ns2:INSS>
    </ns2:EboxID>
    <ns2:MessageID>
      <ns2:Ticket>welcDocCit20141210</ns2:Ticket>
      <ns2:Sequence>1</ns2:Sequence>
    </ns2:MessageID>
    <ns2:HasBeenConsulted>>false</ns2:HasBeenConsulted>
  </ns2:Message>
</ns5: GetMessageStatusResponse>

```

5.3.3 Error codes

See § 8.2 "Business Error Codes"

5.4 OPERATION HASANEBOX

Check if an e-Box exists (and if the e-Box is already used, get the last connection date, in order to know if this e-Box is really used).

5.4.1 Request message construction

Identification of a (list of) e-Box, for a same quality code. For each e-Box: entity value (ex: CompanyID or INSS).

Request to know if an e-Box exist.

sequence

- <> **QualityCode** (QualityCode)

<> [1..1000] **EboxID** (EntityID)

5.4.1.1 Example

Minimal version of a request

```
<v1:HasAnEboxRequest>
  <v11:QualityCode>QUAL_EMP_NOSS</v11:QualityCode>
  <!--1 to 1000 repetitions:-->
  <v11:EboxID>
    <v11:CompanyID>0207148052</v11:CompanyID>
  </v11:EboxID>
</v1:HasAnEboxRequest>
```

Version with several companies

```
<v1:HasAnEboxRequest>
  <v11:QualityCode>QUAL_EMP_NOSS</v11:QualityCode>
  <!--1 to 1000 repetitions:-->
  <v11:EboxID>
    <v11:CompanyID>9995555999</v11:CompanyID>
  </v11:EboxID>
  <v11:EboxID>
    <v11:CompanyID>9995555998</v11:CompanyID>
  </v11:EboxID>
  <v11:EboxID>
    <v11:CompanyID>9995555997</v11:CompanyID>
  </v11:EboxID>
</v1:HasAnEboxRequest>
```

5.4.2 Reply message interpretation

For each e-Box: available(Y/N)? If e-Box already used, date of last connection*.

*: Only for e-Box Enterprise (since new version of the webapplication in July 2014).

Response to know if an e-Box exist.

sequence

<> **QualityCode** (QualityCode)

<> [1..1000] **Ebox** (HasAnEboxResponseType)

5.4.2.1 Example

Response for 1 ebox

```
<ns5:HasAnEboxResponse xmlns:ns2="http://socialsecurity.be/eboxmessage/types/v1"
xmlns:ns3="http://socialsecurity.be/errors/v1" xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1"
xmlns:ns5="http://socialsecurity.be/eboxmessage/v1">
  <ns2:QualityCode>QUAL_EMP_NOSS</ns2:QualityCode>
  <ns2:Ebox>
    <ns2:Ebox>
      <ns2:CompanyID>0207148052</ns2:CompanyID>
    </ns2:Ebox>
    <ns2:Exists>>false</ns2:Exists>
  </ns2:Ebox>
</ns5:HasAnEboxResponse>
```

Response for more than one ebox

```
<ns5:HasAnEboxResponse xmlns:ns2="http://socialsecurity.be/eboxmessage/types/v1"
```

```

xmlns:ns3="http://socialsecurity.be/errors/v1" xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1"
xmlns:ns5="http://socialsecurity.be/eboxmessage/v1">
  <ns2:QualityCode>QUAL_EMP_NOSS</ns2:QualityCode>
  <ns2:Ebox>
    <ns2:Ebox>
      <ns2:CompanyID>9995555999</ns2:CompanyID>
    </ns2:Ebox>
    <ns2:Exists>true</ns2:Exists>
    <ns2:LastConnectionDate>2015-04-07+02:00</ns2:LastConnectionDate>
  </ns2:Ebox>
  <ns2:Ebox>
    <ns2:Ebox>
      <ns2:CompanyID>9995555998</ns2:CompanyID>
    </ns2:Ebox>
    <ns2:Exists>true</ns2:Exists>
    <ns2:LastConnectionDate>2015-04-03+02:00</ns2:LastConnectionDate>
  </ns2:Ebox>
  <ns2:Ebox>
    <ns2:Ebox>
      <ns2:CompanyID>9995555997</ns2:CompanyID>
    </ns2:Ebox>
    <ns2:Exists>false</ns2:Exists>
  </ns2:Ebox>
</ns5:HasAnEboxResponse>

```

5.4.3 Error codes

See § 8.2 "Business Error Codes"

5.5 OPERATION HEALTHCHECK

5.5.1 Request message construction

<> **type** (HealthCheckType)

5.5.1.1 Example

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:mon="http://services.fgov.be/monitoring/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <mon:HealthCheckRequest type="PING"/>
  </soapenv:Body>
</soapenv:Envelope>

```

5.5.2 Reply message interpretation

sequence

<> **Status** (Status)

<> **Component** (Component)

<> **Location** (Location)

<> [0..*] **SanityCheck** (SanityCheck)

<> **timestamp** (dateTime)

5.5.2.1 Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <mon:HealthCheckResponse xmlns:mon="http://services.fgov.be/monitoring/v1">
      <mon:Status>
        <mon:Level>OK</mon:Level>
      </mon:Status>
      <mon:Component>
        <mon:Name>EmployerAccountBalance WS</mon:Name>
        <mon:Version>1.0.0</mon:Version>
      </mon:Component>
      <mon:Location>
        <mon:Environment>TEST</mon:Environment>
        <mon:Host>host.smals.scloud.be</mon:Host>
      </mon:Location>
    </mon:HealthCheckResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

5.5.3 Error codes

Not applicable

5.6 OPERATION LINKeboxMESSAGE

5.6.1 Request message construction

Complex Type defining a LinkEboxMessageRequest

sequence

<> [0..1] **SenderID** (SenderID)

<> **StoredEboxMessageID** (string)

<> [1..1000] **RecipientID** (RecipientID)

5.6.1.1 Example

This paragraph is intentionally left blank.

5.6.2 Reply message interpretation

Complex Type defining a LinkEboxMessageRequest

sequence

<> **MessageID** (string)

<> [0..1000] **RecipientError** (RecipientID)

5.6.2.1 Example


This paragraph is intentionally left blank.

5.6.3 Error codes


This paragraph is intentionally left blank.

5.7 OPERATION PUBLISH EBOXMESSAGE

The publishEboxMessage operation is used to publish a new message to e-Box.

 Different types of e-Box exist (cf. 1.1).

In the request, a “Quality Code” must be given in the “Recipient” tag. That will automatically correspond to an e-Box type. See also § 8 Annex for the list of commons quality codes.

 The recipient’s validation is now checked for this operation at the ESB level. That means that a publish request with an invalid recipient (example: the Enterprise doesn’t exist) will generate a business error with the code ‘90’ (‘Invalid Recipient’). See also § 7.2 for the complete list of the potential business error codes.

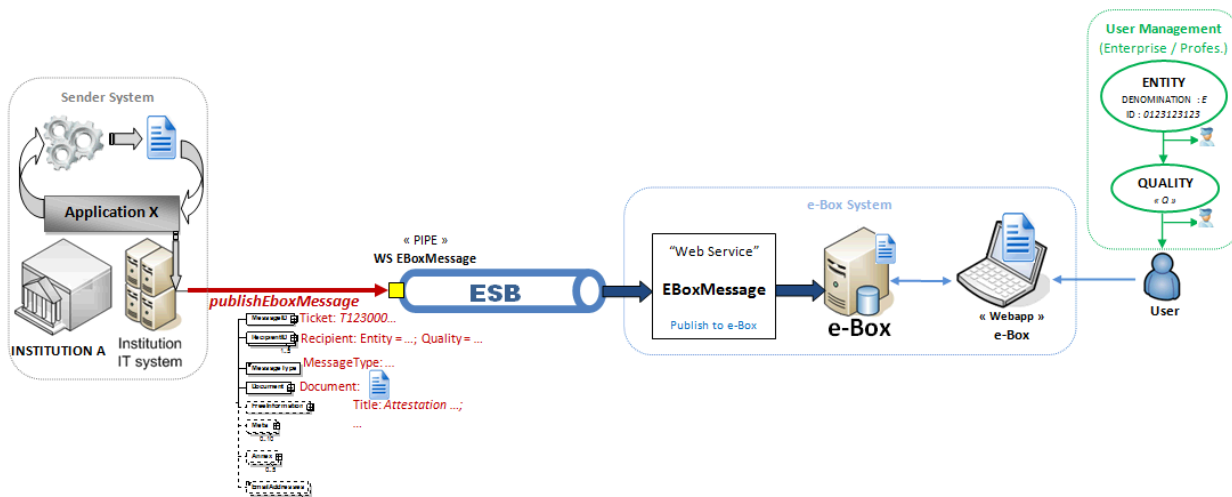


Figure 5 - Operation publishEboxMessage: overview

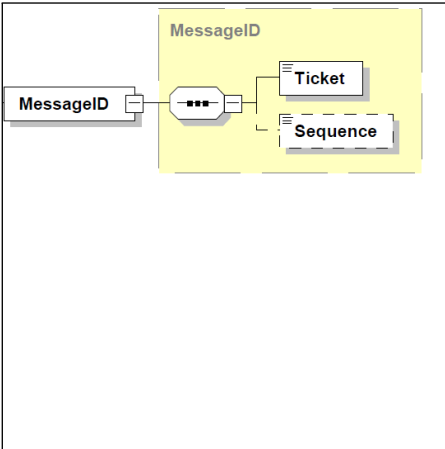
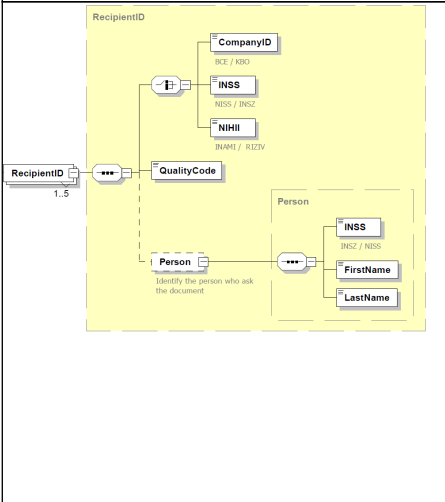
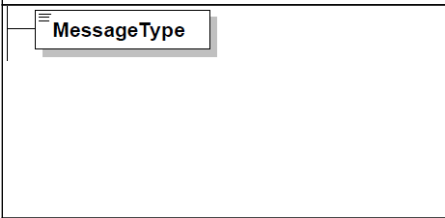
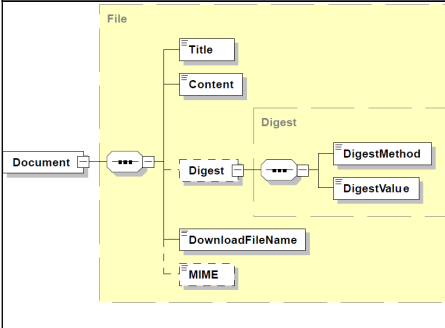
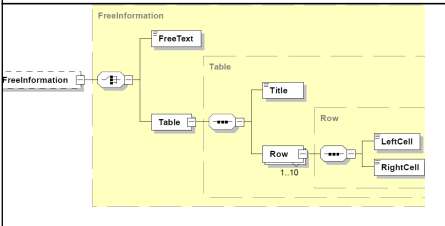
This message contains of course the document that will be published in the recipient’s e-Box, but it is coupled with different pieces of information: *message’s ticket (id), type of message, document’s title, document’s “download file name”, optional free information, metadata or annex...*

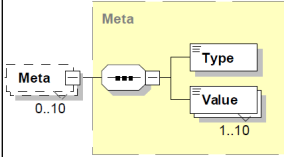
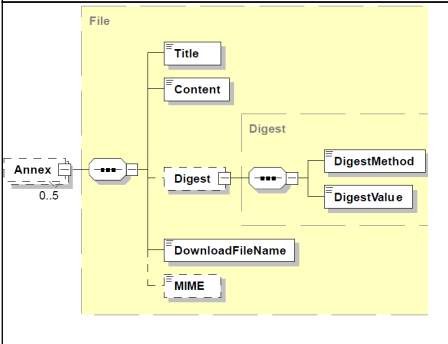

Screenshots of the e-Box web application help the reader to understand how input parameters are used and handled to integrate/show a document in the e-Box. The e-Box’s interfaces are similar between e-Box Enterprise and e-Box Professional. eBox Citizen’s interface looks different, and therefore, some elements that could be provided in a request are not supported with eBox Citizen.

Two request messages (a minimal version and a more detailed one) are explained into details.

5.7.1 Request message construction

PublishEboxMessageRequest	Explanations:
----------------------------------	----------------------

	<p>MessageID (cf. complexType: MessageID)</p> <p>Ticket: functional identifier of the message. Most often, the ticket has a business meaning for the application that sends the message (for example, the reference of a declaration).</p> <p>Sequence [optional]: The sequence number is used to publish different versions of a message having the same ticket number. For example, it is possible to have 3 messages with ticket “12345678”, and sequence “1”; “2” and “3” for each version of the message.</p> <p>Sending a message with an already used ticket/sequence combo will replace the existing message.</p>
	<p>RecipientID [1..5] (cf. complexType: RecipientID)</p> <p>A RecipientID defines an “Entity/Quality couple” that identifies the e-Box where the message will be published.</p> <p>CompanyID or INSS or NIHI: entity id of the recipient, according to the entity type, respectively, an organization (enterprise or institution), an individual (for example a Citizen) or a provider of health care.</p> <p>QualityCode: quality code linked to the given entity. The quality code depends of the “e-Box type”.</p> <p>Person [optional; only for “e-Box Enterprise” and “e-Box Professional Institution”]: a person who is a target recipient among the different users of the specified e-Box.</p>
	<p>MessageType: identifies the type of the message. A message type is defined with eBoxIntegration@smals.be when an institution wants to publish a new sort of message to e-Box (by means of an identification form). Each message type is associated to an institution, a sender application, a validity period, some security checks, and possible metadata.</p>
	<p>Document (cf. complexType: File)</p> <p>Document that will be publish in the specified e-Box. A document has a Title, a Content and a DownloadFileName (with its file extension).</p> <p>A Digest and the MIME-type could be provided.</p>
	<p>FreeInformation: free information in the form of a free text or a table. On the “details page” of the message, the table or the free text is displayed on the top right side of the page (cf. print-screens just below).</p> <p>DEPRECATED (only exploited in e-Box Professional)</p>

	<p>Meta (cf. complexType: Meta)</p> <p>Metadata are used to add more structured information about a message. Metadata are defined for a “message type”, and can be optional or mandatory. In e-Box, metadata can be used as search criteria.</p> <p>NOT USED IN EBOX CITIZEN</p>
	<p>Annex: it is possible to add 1 to 5 annexes in a message. An annex has the same type that a document.</p> <p>In e-Box, a title is followed by a <i>paper clip</i> symbol in order to indicate that some annexes are joined in this message.</p> <p>The “details page” of a message allows a user to consult or download the annexes (cf. print-screen just below).</p> <p>NOT USED IN EBOX CITIZEN</p>
	<p>EmailAddresses: Mail addresses that the sender could add (optional) – max. 5 mail addresses.</p> <p>e-Box sends a notification mail to these addresses in order to indicate that there is a new incoming message in the e-Box of the recipient (not used for eBox Citizen).</p> <p>NOT USED IN EBOX CITIZEN</p>

Complex Type defining an PublishEboxMessageRequest

sequence

<> **MessageID** (MessageID)

<> [0..1] **SenderID** (SenderID)

<> [1..5] **RecipientID** (RecipientID)

<> **MessageType** (MessageType)

<> **Document** (File)

<> [0..1] **FreeInformation** (FreeInformation)

DEPRECATED - NOT SUPPORTED FOR EBOX CITIZEN

<> [0..10] **Meta** (Meta)

NOT SUPPORTED FOR EBOX CITIZEN

<> [0..5] **Annex** (File)

NOT SUPPORTED FOR EBOX CITIZEN

<> [0..5] **EmailAddresses** (Email)

DEPRECATED

5.7.1.1 Example

Minimal version of a message:

```
<PublishEboxMessageRequest
xsi:schemaLocation="http://socialsecurity.be/eboxmessage/v1 EboxMessage_v1.xsd"
xmlns="http://socialsecurity.be/eboxmessage/v1"
xmlns:types="http://socialsecurity.be/eboxmessage/types/v1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <types:MessageID>
    <types:Ticket>testWSebox2012</types:Ticket>
  </types:MessageID>
  <types:RecipientID>
    <types:CompanyID>0123456789</types:CompanyID>
    <types:QualityCode>QUAL_EMP_NOSS</types:QualityCode>
  </types:RecipientID>
  <types:MessageType>DIMONA1</types:MessageType>
  <types:Document>
    <types:Title>titre</types:Title>
    <types:Content>cid:534484181045</types:Content>
    <types:DownloadFileName>test.pdf</types:DownloadFileName>
  </types:Document>
</PublishEboxMessageRequest>
```

Message request with 2 recipients, 2 metas and 3 annexes:

```
<v1:PublishEboxMessageRequest>
  <v1:MessageID>
    <v1:Ticket>12345678</v1:Ticket>
    <v1:Sequence>2</v1:Sequence>
  </v1:MessageID>
  <v1:RecipientID>
    <v1:INSS>999999999999</v1:INSS>
    <v1:QualityCode>PROFESSIONAL</v1:QualityCode>
  </v1:RecipientID>
  <v1:RecipientID>
    <v1:CompanyID>1234567890</v1:CompanyID>
    <v1:QualityCode>QUAL_EMP_NOSS</v1:QualityCode>
    <v1:Person>
      <v1:INSS>999999999999</v1:INSS>
      <v1:FirstName>Francois</v1:FirstName>
      <v1:LastName>Full</v1:LastName>
    </v1:Person>
  </v1:RecipientID>
  <v1:RecipientID>
    <v1:CompanyID>1234567890</v1:CompanyID>
    <v1:QualityCode>QUAL_SP_LEG</v1:QualityCode>
  </v1:RecipientID>
  <v1:MessageType>DocWIDE</v1:MessageType>
  <v1:Document>
    <v1:Title>document de test pour ebox</v1:Title>
    <v1:Content>cid:534484181045</v1:Content>
    <v1:Digest>
      <v1:DigestMethod>SHA-256</v1:DigestMethod>
      <v1:DigestValue>YTONFS1mEb7N01hNH6S3KMsIS5mm4/rf+cgkiuhloys=</v1:DigestValue>
    </v1:Digest>
    <v1:DownloadFileName>document.pdf</v1:DownloadFileName>
    <v1:MIME>application/pdf</v1:MIME>
```

```

</v11:Document>
<v11:Meta>
  <v11:Type>BENOSS</v11:Type>
  <v11:Value>0987654321</v11:Value>
  <v11:Value>1234567890</v11:Value>
</v11:Meta>
<v11:Meta>
  <v11:Type>RequestDate</v11:Type>
  <v11:Value>2012-03-12</v11:Value>
</v11:Meta>
<v11:Annex>
  <v11:Title>title first Annex</v11:Title>
  <v11:Content>cid:534484181045</v11:Content>
  <v11:Digest>
    <v11:DigestMethod>SHA-256</v11:DigestMethod>
    <v11:DigestValue>YTONFS1mEb7N01hNH6S3KMsiS5mm4/rf+cgkuihloys=</v11:DigestValue>
  </v11:Digest>
  <v11:DownloadFileName>ANNEXE01.pdf</v11:DownloadFileName>
  <v11:MIME>application/pdf</v11:MIME>
</v11:Annex>
<v11:Annex>
  <v11:Title>title annex 2</v11:Title>
  <v11:Content>cid:524157812819</v11:Content>
  <v11:DownloadFileName>annex2.csv</v11:DownloadFileName>
  <v11:MIME>text/csv</v11:MIME>
</v11:Annex>
<v11:Annex>
  <v11:Title>Annex 3 Title</v11:Title>
  <v11:Content>cid:524157812819</v11:Content>
  <v11:DownloadFileName>annex3.csv</v11:DownloadFileName>
  <v11:MIME>text/csv</v11:MIME>
</v11:Annex>
</v1:PublishEboxMessageRequest>

```

5.7.2 Reply message interpretation

Response of a PublishEboxMessageRequest

sequence

<> ID (string)

5.7.2.1 Example

```

<PublishEboxMessageResponse>
  <ID>035000373APHZ</ID>
</PublishEboxMessageResponse>

```

Explanation:

This ID is a technical message ID. This message ID can be used to search the message in e-Box, for example in order to check the publication.

This kind of answer means that the message will be treated. If an error occurs at the WS level, an error code is returned (cf. §7 *Common error codes*).

5.7.3 Error codes

See § 8.2 "Business Error Codes"

5.8 OPERATION STOREEBOXMESSAGE

5.8.1 Request message construction

Complex Type defining a StoreEboxMessageRequest

sequence

<> **MessageID** (MessageID)

<> [0..1] **SenderID** (SenderID)

<> **MessageType** (MessageType)

<> **Document** (File)

<> [0..1] **FreeInformation** (FreeInformation)

<> [0..10] **Meta** (Meta)

<> [0..5] **Annex** (File)

5.8.1.1 *Example*

This paragraph is intentionally left blank.

5.8.2 Reply message interpretation

Response of a StoreEboxMessageResponse

sequence

<> **ID** (string)

5.8.2.1 *Example*

This paragraph is intentionally left blank.

5.8.3 Error codes

This paragraph is intentionally left blank.

6. COMMON TYPES

6.1 EBOXMESSAGE_TYPES_V1

Namespace: <http://socialsecurity.be/eboxmessage/types/v1>

Version: 1.0

6.1.1 Application

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	150	
	minLength	1	
annotation	Identify the application that produces the document.		

6.1.2 CBE

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	pattern	[0-9]{10}	
annotation	BCE - KBO		

6.1.3 Denomination

«XSDcomplexType»

Denomination in NL, FR, DE

sequence

<> **DenominationNL** (DenominationValue)

<> **DenominationFR** (DenominationValue)

<> **DenominationDE** (DenominationValue)

6.1.4 DenominationValue

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	200	
annotation	Denomination of an Entity or a Subdivision. limited to 200 char		

6.1.5 Digest

«XSDcomplexType»

Binary content of a document

sequence

<> **DigestMethod** (DigestMethod)

<> **DigestValue** (DigestValue)

6.1.6 DigestMethod

«Enumeration»

facets	value	annotation
	SHA-1	
	SHA-256	
	SHA-512	
annotation	Digest Algorithm (SHA-1, SHA-256, SHA-512).	

6.1.7 DigestValue

«XSDsimpleType»

type	base64Binary		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	150	
	minLength	1	
annotation	Value of digest		

6.1.8 EboxEntityQuality

«XSDcomplexType»

Identify the e-Box where the message is published.

sequence

<> **QualityCode** (QualityCode)

choice

<> [1..1] **INSS** (INSS)

NISS / INSZ

<> [1..1] **NIHII** (NIHII)

INAMI / RIZIV

sequence

<> [1..1] **CompanyID** (CBE)

BCE / KBO

<> [0..1] **SubentityId** (SubentityId)

6.1.9 EboxInfo

«XSDcomplexType»

Info about an e-Box: Entity ID, Entity Name, Email Address, Language

sequence

<> **EntityID** (EntityID)

<> **EmailAddress** (Email)

<> **Language** (Language)

<> **EntityName** (Denomination)

6.1.10 Email

«XSDsimpleType»

type	string
-------------	--------

facets	kind	value	annotation
	derivation	restriction	
	maxLength	80	
	pattern	[A-Za-z0-9._%\\-]+@[A-Za-z0-9\\.\\-]+ \\.[A-Za-z]{2,4}	
annotation	email adress		

6.1.11 EntityID

«XSDcomplexType»

Identify the Entity. An Entity can be an Enterprise, an Institution, a Professional of Social Security, or a Citizen.

sequence

choice

<> [1..1] **INSS** (INSS)

NISS / INSZ

<> [1..1] **NIHII** (NIHII)

INAMI / RIZIV

sequence

<> [1..1] **CompanyID** (CBE)

BCE / KBO

<> [0..1] **SubentityId** (SubentityId)

6.1.12 ErrorStatusCode

«Enumeration»

facets	value	annotation
	NOT_AUTHORIZED	
	NOT_FOUND	
annotation	Error Code	

6.1.13 File

«XSDcomplexType»

A file must have a title, a content and a "name" to download file, and save it on disk. Mime is optional

sequence

- <> **Title** (Title)

- <> **Content** (swaRef)

- <> [0..1] **Digest** (Digest)

- <> **DownloadFileName** (FileName)

- <> [0..1] **MIME** (MIME)

6.1.14 FileName

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	80	
	minLength	1	
annotation	Name to download the file.		

6.1.15 FreeInformation

«XSDcomplexType»

Free additional information.

choice

- <> **FreeText** (FreeText)

- <> **Table** (Table)

6.1.16 FreeText

«XSDsimpleType»

type	string
-------------	--------

facets	kind	value	annotation
	derivation	restriction	
	maxLength	5000	
	minLength	1	
annotation	Free text. Can be html or simple text.		

6.1.17 GetMessageStatusErrorType

«XSDcomplexType»

Request to know if a message has been readed or downloaded

sequence

<> **EboxID** (EntityID)

<> **MessageID** (MessageID)

<> **ErrorCode** (ErrorStatusCode)

6.1.18 GetMessageStatusResponseType

«XSDcomplexType»

Response to know if a message is readed or downloaded

sequence

<> **EboxID** (EntityID)

<> **MessageID** (MessageID)

<> **HasBeenConsulted** (boolean)

6.1.19 GetMessageStatusType

«XSDcomplexType»

Request to know if a message has been readed or downloaded

sequence

<> **EboxID** (EntityID)

<> **MessageID** (MessageID)

6.1.20 HasAnEboxResponseType

«XSDcomplexType»

Response to know if an e-Box exist.

sequence

<> **Ebox** (EntityID)

<> **Exists** (boolean)

<> [0..1] **LastConnectionDate** (date)

6.1.21 INSS

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	pattern	[0-9]{11}	
annotation	NISS - INSZ		

6.1.22 Language

«Enumeration»

facets	value	annotation
	FR	
	NL	
	DE	
	BI	
annotation	Language code	

6.1.23 LeftCell

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	50	
	minLength	1	
annotation	Left cell of a table row		

6.1.24 MessageID

«XSDcomplexType»

ID of the message.

sequence

<> **Ticket** (Ticket)

<> [0..1] **Sequence** (Sequence)

6.1.25 MessageType

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	80	
	minLength	6	
annotation	Type of document.		

6.1.26 Meta

«XSDcomplexType»

Meta data.

sequence

<> **Type** (ShortString)

<> [1..10] **Value** (ShortString)

6.1.27 MIME

«Enumeration»

facets	value	annotation
	application/pdf	
	application/zip	
	text/csv	
	text/html	
	text/plain	
	text/xml	
	application/vnd.oasis.opendocument.text	
	application/vnd.oasis.opendocument.spreadsheet	
	application/vnd.oasis.opendocument.presentation	
	application/vnd.oasis.opendocument.graphics	
	application/vnd.ms-excel	
	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	
	application/vnd.ms-powerpoint	
	application/msword	
	application/vnd.openxmlformats-officedocument.wordprocessingml.document	
annotation	MIME supported by e-Box.	

6.1.28 Name

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	100	
	minLength	1	
annotation	Name limited to 100 char		

6.1.29 NIHI

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	pattern	[0-9]{8}	

annotation	INAMI - RIZIV
-------------------	---------------

6.1.30 Person

«XSDcomplexType»

Identify a person in an e-Box.

sequence

<> **INSS** (INSS)
INSZ / NISS

<> **FirstName** (Name)

<> **LastName** (Name)

6.1.31 QualityCode

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	125	
	minLength	1	
annotation	Quality code from the userManagement of social security. Ex : QUAL_EMP_NOSS, QUAL_SSC, QUAL_SP_IND, ...		

6.1.32 RecipientID

«XSDcomplexType»

Identify the recipient of the message. The recipient can be an Enterprise, an Institution, a Professional of Social Security or a Citizen.

sequence

<> [0..1] **Person** (Person)

Identify the person who ask the document

<> **QualityCode** (QualityCode) *inherited from* [EboxEntityQuality](#)

choice

<> [1..1] **INSS** (INSS)
NISS / INSZ

- <> [1..1] **NIHII** (NIHII)
INAMI / RIZIV
sequence
- <> [1..1] **CompanyID** (CBE)
BCE / KBO
- <> [0..1] **SubentityId** (SubentityId)

6.1.33 RightCell

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	150	
	minLength	1	
annotation	Righ cell of a table row		

6.1.34 Row

«XSDcomplexType»

Row of a table.

sequence

- <> **LeftCell** (LeftCell)
- <> **RightCell** (RightCell)

6.1.35 SenderID

«XSDcomplexType»

Identify the application and the institution who send the message.

sequence

- <> **Application** (Application)
Identity of application.

<> **Institution** (CBE)

CBE of institution

6.1.36 Sequence

«XSDsimpleType»

type	integer		
facets	kind	value	annotation
	derivation	restriction	
	maxInclusive	99999	
	minInclusive	0	
annotation	Additional Number to publish many different document with the same Ticket.		

6.1.37 ShortString

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	25	
	minLength	1	
annotation	Short string limited to 25 char		

6.1.38 ShortTitle

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	50	
	minLength	1	
annotation	Title of table. Limited to 25 char		

6.1.39 SubentityId

«XSDsimpleType»

type	string
-------------	--------

facets	kind	value	annotation
	derivation	restriction	
	maxLength	50	
	pattern	[A-Za-z0-9]+	
annotation	Sub entity id		

6.1.40 Table

«XSDcomplexType»

Table like html table to display some free value. This table has only 2 column and max 10 row.

sequence

<> **Title** (ShortTitle)

<> [1..10] **Row** (Row)

6.1.41 Ticket

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	25	
	minLength	1	
annotation	Functionnal ID of the document.		

6.1.42 Title

«XSDsimpleType»

type	string		
facets	kind	value	annotation
	derivation	restriction	
	maxLength	400	
	minLength	1	
annotation	Same as subject of mail		

6.2 MONITORING-V1

Namespace: <http://services.fgov.be/monitoring/v1>

Version:

6.2.1 Component

«XSDcomplexType»

Describes the component as created in CMDB

sequence

<> **Name** (string)

The free-text name of the component, validated as the official name by TTF

<> **Version** (string)

The deployed version. In case of maven artifacts, the version should be in standardized X.Y.Z format.

<> [0..1] **MavenReference** (MavenReference)

<> [0..1] **CatalogPart** (string)

<> [0..1] **Extensions** (Extensions)

6.2.2 Environment

«Enumeration»

facets	value	annotation
	PRD	Production
	SIM	Simulation
	ACC	Acceptance
	INT	Integration
	TST	Test
	DEV	Development
	SIC	SIC performance testing
	LOCAL	Local developer pc
	OTHER	Other
annotation	All environments	

6.2.3 Extensions

«XSDcomplexType»

An extension point to allow applications to add additional information

sequence

6.2.4 HealthCheckType

«Enumeration»

facets	value	annotation
	PING	The PING health check does not perform checks on its dependencies, and is used to see if the service is life
	DEFAULT	The DEFAULT health check does perform checks on its local dependencies, eg. database, filesystem, etc
	DEEP	The DEEP health check does perform checks on its local dependencies and external dependencies, eg. remote EJB and web services
annotation	Defines the kind of health check test to execute	

6.2.5 Location

«XSDcomplexType»

Describes the deployment location of the component

sequence

<> **Environment** (Environment)

<> **Host** (string)

The hostname of the server

<> [0..1] **Port** (int)

The port the application is listening on

<> [0..1] **Name** (string)

The name

<> [0..1] **URL** (string)

The URL representing the filepath location or address of the component

<> [0..1] **Extensions** (Extensions)

6.2.6 MavenReference

«XSDcomplexType»

Describes the artifact as catagorized by maven.

<> **groupId** (string)

<> **artifactId** (string)

6.2.7 Resource

«XSDcomplexType»

The resource that is checked in a sanity check.

sequence

<> [0..1] **Location** (Location)

<> **name** (string)

<> **type** (ResourceType)

<> **reference** (string)

6.2.8 ResourceType

«Enumeration»

facets	value	annotation
	DB	Database
	LDAP	LDAP directory service
	CACHE	Cache
	SOCKET	TCP Socket
	MAINFRAME	Mainframe
	BATCH	Batch system
	FILESYSTEM	File system
	CLASSPATH	Java Classpath

	CERTIFICATE	X.509 Certificate
	QUEUE	JMS Queue
	EJB	JavaEE Enterprise Java Bean
	WS	SOAP Web service
	MAIL	SMTP Mail
	OTHER	Other
annotation	Types of dependent resources	

6.2.9 SanityCheck

«XSDcomplexType»

A check against a single resource to validate its health.

sequence

<> **Status** (Status)

<> **Resource** (Resource)

<> **TimeInMillis** (int)

<> [0..1] **Extensions** (Extensions)

<> [0..1] **StackTrace** (string)

<> **id** (string)

<> **description** (string)

<> **failSafe** (boolean)

<> **type** (HealthCheckType)

6.2.10 Status

«XSDcomplexType»

The status of the component or a test.

sequence

- <> **Level** (StatusLevel)

- <> [0..1] **Code** (string)
A status code defined by the component

- <> [0..1] **Message** (string)
The free-text status message

- <> [0..1] **Extensions** (Extensions)

6.2.11 StatusLevel

«Enumeration»

facets	value	annotation
	OK	
	WARNING	There is an abnormal situation with no impact for the end-user, eg elevated response times, queues filled
	CRITICAL	There is a failure with impact to the end-user on part of the application.
	FATAL	The Critical level is only used for the global health status, and should not be used for sanity checks.
annotation	This is the status of the monitor test result.	

7. COMMON TYPES SUPPLEMENTARY INFORMATION

7.1 COMPLEXTYPE: MESSAGEID

For the e-Box user: ticket / sequence are only displayed in “e-Box Enterprise” and “e-Box Professional”:

- For an e-Box user, the ticket / sequence are displayed on the “details page” of each message. Ticket can be used as search criteria in e-Box Enterprise and e-Box Professional. (Ticket is not displayed in eBox Citizen).

- For the sender and for e-Box Team, the ticket can be used to search the message in the e-Box system (for example, in order to check the message publication). The ticket is also used to inform the sender of the message status [read / unread].

7.2 COMPLEXTYPE: *RECIPIENTID*

A RecipientID defines an “Entity/Quality” couple that identifies the e-Box where the message will be published.

CompanyID or **INSS** or **NIHII**: entity id of the recipient, according to the entity type, respectively, an organization (enterprise or institution), an individual (for example a Citizen) or a provider of health care.

- An organization (enterprise or institution) is identified with a **CompanyID** (*numéro BCE/KBO number*);
- A natural person (Citizen / Professional / Curator ; “Service Provider *Natural Person*”....) is identified with its **INSS** (NISS / INSZ);
- A healthcare provider could be identified by its **NIHII** (n° INAMI / RIZIV).

QualityCode: quality code linked to the given entity. The quality code depends of the “Box type”

è See §8 Annex for a non-exhaustive list of quality codes.

Person: cf. complexType: Person

7.3 COMPLEXTYPE: *FILE*

This complex type is used for a Document or an Annex

Emetteur	Description	Type	Date d'envoi
	Document SIGEDIS pour démo Ouvrir Télécharger Archiver		03 mai 2013
	Test pour ONEM: Décision C62 Ouvrir Télécharger		19 avril 2013

Figure 1 - "Title" of a message (eBox Citizen)

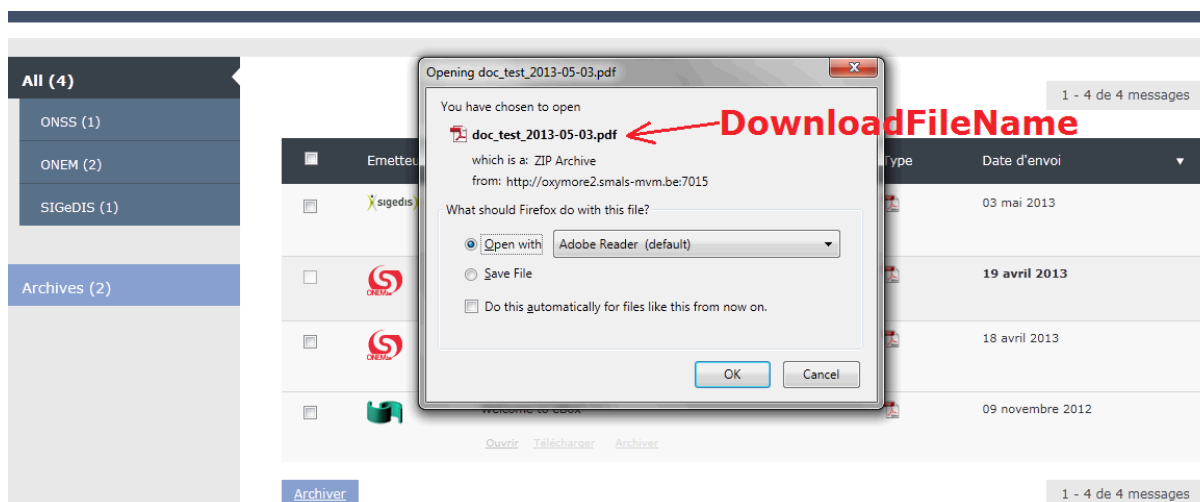



Figure 2 - "DownloadFileName" (eBox Citizen) [browser Mozilla Firefox]

7.4 COMPLEXTYPE: *FREEINFORMATION*

 Détail du document nouveau test avec free text

[Retour](#)


Description


N° de ticket	1298346795
N° de séquence	0
Date d'expiration	27-03-2012
Type de message	DocWIDE
Numéro d'immatriculation ONSS	5110468182
Numéro d'immatriculation ONSS	5110468182
Numéro d'immatriculation ONSS	1234567890

Attention, document confidentiel. Veuillez ne pas divulguer les informations reprises dans le document PDF.

FreeInformation:FreeText

Document principal

Description	Format	Type de signature 
nouveau test avec free text		

Annexes 

Pas d'annexes pour ce message

Figure 3 - "FreeInformation»: free text (e-Box Professional)

7.5 COMPLEXTYPE TABLE

Détail du document Envoi formulaire X-Y.Z pour démo (avec annexe) [Retour](#)

FreeInformation: Table

Description

N° de ticket	TST_WS_2013-05-30_b04
N° de séquence	1
Date d'expiration	30-08-2013
Type de message	Document e-Box

Tableau informatif concernant le formulaire X-Y.Z	
Date demande	26-05-2013
Votre correspondant	Mr. X. XXXXXXXXXXX
Numero de dossier	123-XXXXXXXXXX-XYZ
Date limite de retour	30-06-2013

Document principal

Description	Format	Type de signature
Envoi formulaire X-Y.Z pour démo (avec annexe)	application/pdf	

Annexes

Description	Format	Type de signature
Document Annexe 1 - version .doc du formulaire	application/msword	

Figure 4 - "FreeInformation": table (e-Box Professional)

7.6 COMPLEXTYPE: META

Meta for an e-Box User:

In e-Box, metadata are displayed on the "details page" of a message, into the "Description" section.

Metadata can be used as search criteria among messages of a same type.

DE FR NL Contact ANTOINE JEAN MOULART + | [Se déconnecter](#)

e-Box ENTERPRISE

Détail de la notification [Retour](#)

Modification d'un mandat [Voir](#) [Télécharger](#)

Label(s) assigné(s)	+
Type	Modification d'un mandat
Emetteur	MAHIS
Institution	ONSS
Date de réception	16/05/2014
Date d'expiration	16/08/2014
Numéro de ticket	15052014-demo-acc-021
Id	035001Y9CKWKZ

Informations complémentaires

Numéro d'agrément	600
Numéro d'entreprise de l'employeur	0457527422
Numéro d'entreprise de l'employeur	0123456789
Matricule ONSS	891234567
Matricule ONSS	891234567
Numéro de registre national d'un prestataire de services	55021411199
Type de notification	MAHI003

MetaData

© 2014 Social Security copyright 1.1.4

Figure 5 - "Metadata" displayed on the "message details" page (e-Box Enterprise)

e-Box ENTERPRISE

Retour

Recherche personnalisée

Critères

Date: Sélectionnez

Statut: Sélectionnez

Recommandé: Sélectionnez

Expiration du message: Sélectionnez

A l'attention de: Sélectionnez

Numéro de ticket: [input]

Emetteur: DMFA

Sous-catégorie: Sélectionnez

Critère(s) avancé(s) (1)

Trimestre: [input] ← Metadata (used as search criteria)

Label: Sélectionnez +

Annuler Rechercher

Recherches sauvegardées

Sélectionnez la vue pour afficher les critères

Vue par date

- A l'attention de tout le monde [edit] [delete]
- Dmfa non-lues [edit] [delete]
- ce mois ; non-lu ; pour dominique [edit] [delete]
- drs-chomage [edit] [delete]

© 2014 Social Security copyright 1.1.4

Figure 6 - "Metadata" used as search criteria (e-Box Enterprise)

8. COMMON ERROR CODES

8.1 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.

8.2 BUSINESS ERROR CODES

These error codes are defined by the application.

Code	Description	Cause	Explication
10	Hash computed not equal to the providing hash		
20	The provided institution ID is not authorized to publish for another institution		
30	You are authenticated as MandateSender. A MandateSender must specify a SenderID in SOAP request.		
40	The combo institution/application cannot send this message type		
50	A given meta data is not authorized for this kind of message		
51	A given meta data is not well formed according the regular expression		
52	A required meta is missing according to the constraints		
90	Invalid Recipient		

9. ANNEX

9.1 WS EBOXMESSAGE ENDPOINT URI

Integration:	https://professionalservices-int.socialsecurity.be/EBoxMessage/v1
Acceptation:	https://professionalservices-acpt.socialsecurity.be/EBoxMessage/v1
Production:	https://professionalservices.socialsecurity.be/EBoxMessage/v1

9.2 COMMON QUALITY CODES

Non-exhaustive list of quality codes:

e-Box Type	Quality Code	FR	NL	Expected entity ID type
e-Box Enterprise	QUAL_EMP_NOSS	Employeur ONSS	Werkgever RSZ	CompanyID
	QUAL_EMP_NOSSPLA	Employeur ONSS-APL	Werkgever RSZ-PPO	CompanyID
	QUAL_COMPANY	Entreprise sans personnel	Onderneming zonder personeel	CompanyID
	QUAL_SSC	Secrétariat sociale agréé	Erkend sociaal secretariaat	CompanyID
	QUAL_SP_LEG	Prestataire de services (personne morale)	Dienstverlener (rechtspersoon)	CompanyID
	QUAL_SP_IND	Prestataire de services (personne physique)	Dienstverlener (natuurlijke persoon)	INSS
	QUAL_FSC	Full Service Center APL	Full Service Center PPO	CompanyID
	QUAL_SPPLA_LEG	Prestataire de services ONSSAPL	Dienstverlener ONSSAPL	CompanyID
	QUAL_CUR	Curateur	Curator	INSS
e-Box Professionnal (Institution)	QUAL_MUN_ADM	Administration Communale	Gemeente bestuur	CompanyID
	QUAL_MUT_SOCSEC	Mutuelle en matière de sécurité sociale	Mutualiteit voor de sociale zekerheid	CompanyID
	QUAL_SOC_ACTION	Action Sociale	Maatschappelijk Welzijn	CompanyID

	QUAL_COMP_PENSION	Gestionnaire pensions complementaires	Beheerder aanvullende pensioenen	CompanyID
e-Box Professional (Personnal)	PROFESSIONAL	<i>Professionnel (fonctionnaire)</i>	<i>professioneel (ambtenaar)</i>	INSS
eBox Citizen	CITIZEN	<i>Citoyen</i>	Burger	INSS

If you need to publish to an e-Box with another quality type, you can ask to eBoxIntegration@smals.be to get the valid quality code to use.

9.3 USING THE X.509 CERTIFICATE

Each request to the WS EBoxMessage must be signed with a X.509 certificate (cf.5.2.).

9.3.1 Dedicated certificate by sending application

For this use, each application of a same institution has a dedicated certificate. The sender doesn't have to give a "Sender tag" in the request.

CompanyID of the institution and URN of the application are automatically detected in the request signature made by means of the X.509 certificate.

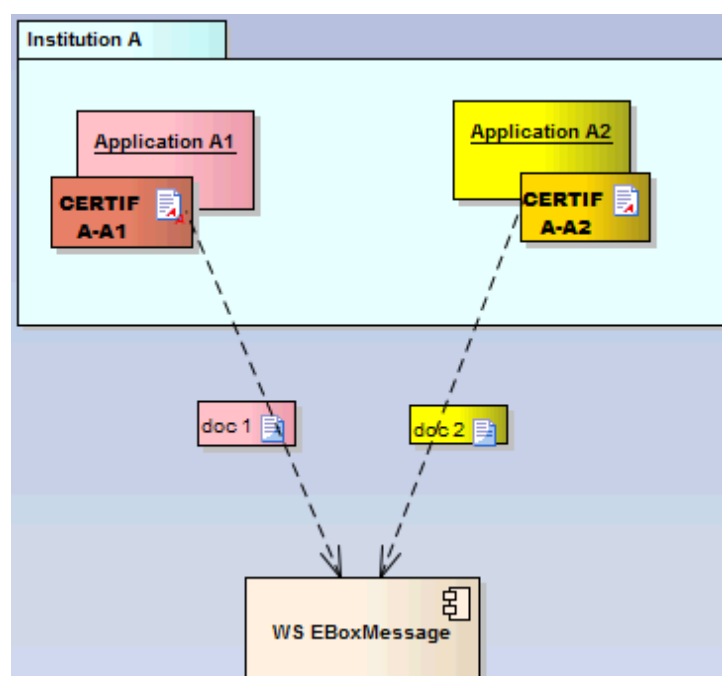
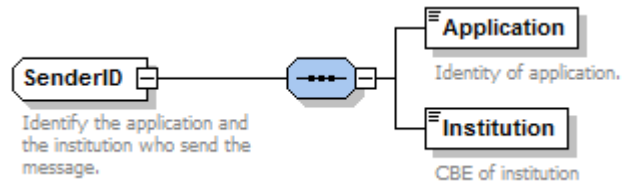


Figure 6 - Dedicated certificate by application

9.3.2 Single certificate shared by several sending applications

For this use, each application of a same institution uses the same certificate. The sender has to give a "Sender tag" in the request, in order to indicate for which application the message will be published.



```

<SenderID>
  <Application>${urn:of:application-to-indicate}</Application>
  <Institution>${0123456789}</Institution>
</SenderID>

```

The value given in the “SenderID\Institution” tag must be the CompanyID of the institution, which is also detected and checked in the request signature made by means of the X.509 certificate.

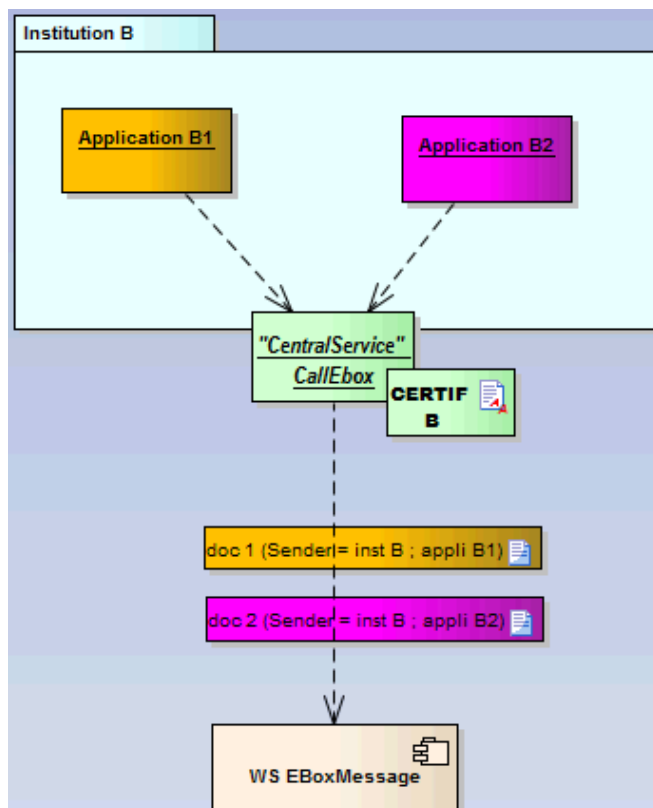


Figure 7 - Single certificate shared by several applications

9.4 CONTACT

e-Box team:	eBoxIntegration@smals.be
CSM Smals:	Catherine Schoetter
CPL Smals:	Roel Biermans
PL Smals (TO&P):	Jérôme Vos

PL Smals (GA/TB):	Barbara Meyers
SPOC Smals "eBox Citizen":	Kristien Plomteux
e-Box analyst (Smals):	Antoine Moulart
e-Box architect (Smals):	Adrien Clerbaut
SPOC NSSO (e-Box Enterprise):	Philippe Benoît (ONSS-RSZ)
SPOC CBSS (e-Box Professional & eBox Citizen):	Claudia Laeremans (BCSS-KSZ)