



**Social Security
ONSS/RSZ**

**Place Victor Horta 11 - 1060 BRUXELLES - Tél 02 509 59 59
Internet: www.onss.fgov.be**

**Victor Hortaplein 11 -1060 BRUSSEL - Tel. 02 509 59 59
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

**RegisteredMail
Version 1.6 (*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	5
2. DOCUMENT MANAGEMENT	5
2.1 Document goal	5
2.2 Document history	5
2.3 Document references.....	5
3. SERVICE HISTORY	6
4. PREREQUISITES.....	7
4.1 Business prerequisites	7
4.2 Technical prerequisites.....	7
4.2.1 Security.....	7
5. DESCRIPTION OF THE SERVICE OPERATIONS.....	8
5.1 Operation getStatus.....	9
5.1.1 Request message construction	9
5.1.1.1 Example.....	9
5.1.2 Reply message interpretation.....	9
5.1.2.1 Example.....	9
5.1.3 Error codes	9
5.2 Operation healthCheck.....	10
5.2.1 Request message construction	10
5.2.1.1 Example.....	10
5.2.2 Reply message interpretation.....	10
5.2.2.1 Example.....	10
5.2.3 Error codes	11
5.3 Operation sendDocument.....	11
5.3.1 Request message construction	11
5.3.1.1 Example.....	11
5.3.2 Reply message interpretation.....	12
5.3.2.1 Example.....	12
5.3.3 Error codes	13
6. COMMON TYPES	14
6.1 Monitoring-v1.....	14
6.1.1 Component	14
6.1.2 Environment	15
6.1.3 Extensions	15
6.1.4 HealthCheckType	15
6.1.5 Location	15
6.1.6 MavenReference	16
6.1.7 Resource	16
6.1.8 ResourceType	16
6.1.9 SanityCheck	17
6.1.10 Status	18
6.1.11 StatusLevel	18

6.2 RegisteredMail_types_v1	18
6.2.1 AnnexFile	18
6.2.2 CBE	19
6.2.3 CheckEboxStatus	19
6.2.4 Citizen	19
6.2.5 Digest	19
6.2.6 DigestMethod	20
6.2.7 DigestValue	20
6.2.8 DocumentRequest	20
6.2.9 DocumentResponse	20
6.2.10 Enterprise	21
6.2.11 Entity	21
6.2.12 File	21
6.2.13 FileDescription	21
6.2.14 FileName	22
6.2.15 INSS	22
6.2.16 Language	22
6.2.17 Meta	22
6.2.18 MIME	23
6.2.19 QualityCode	23
6.2.20 Receipt	23
6.2.21 ReceiptContent	24
6.2.22 RegisteredMailItem	24
6.2.23 Sender	25
6.2.24 SenderApplicationID	25
6.2.25 ShortString	25
6.2.26 SignedFile	25
6.2.27 Status	25
6.2.28 StatusRequest	26
6.2.29 StatusResponse	26
6.2.30 Ticket	26
6.2.31 Title	26
6.2.32 Token	27
6.2.33 Type	27
6.3 Sweref.....	27
6.3.1 swaRef	27
7. COMMON ERROR CODES.....	27
7.1 System Error Codes.....	27
7.2 Business Error Codes.....	28
8. ANNEX	28

1. GOAL OF THE SERVICE

Service that allows the sending of a registered mail under the Act of of March 19, 2013.

The service RegisteredMail will allow the partners in the Social Security sector to send digital documents as registered mail. This electronic version of registered mail has the same juridical value as its paper equivalent.

2. DOCUMENT MANAGEMENT

2.1 DOCUMENT GOAL

This document provides functional and technical information on calling RegisteredMail, as provided by the Crossroads Bank for Social Security (CBSS). 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 Crossroads Bank for Social Security (CBSS).

2.2 DOCUMENT HISTORY

Version	Date	Author	Description of changes / remarks
0.1	20/11/2013	J. Buyse	First draft
0.2	25/11/2013	J. Buyse	Modification after scope change
0.3	29/04/2014	V. Adam / J. Buyse	This service Specification is generated based on XSD/WSDL 1.2
0.4	13/05/2016	V. Adam / A. Moulart	This service Specification is generated based on XSD/WSDL 1.3
0.5	18/09/2017	V. Adam / A. Moulart	This service Specification is generated based on XSD/WSDL 1.4
0.6	26/07/2021	V. Adam / J. Bury	This service Specification is generated based on XSD/WSDL 1.6

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:

Not applicable

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): 2/08/2018

Previous version number (production): 1.5

List of changes:

7/01/2014 version 1.0: First version

29/04/2015 version 1.2: interaction between back-end and DSP is modified

12/05/2016 version 1.3:

- complexType name="Citizen" added

- complexType name="Sender" added

- simpleType name="CheckEboxStatus" added

- simpleType name="NISS" added

- simpleType name="SenderApplicationID" added

==> SendDocumentRequest impacted

18/09/2017 version 1.4:

- simpleType name="CheckEboxStatus" added

- complexType name="Sender" added

- simpleType name="INSS" added

- simpleType name="SenderApplicationID" added

==> SendDocumentRequest impacted

8/10/2020 version 1.5:

- complexType name = "AnnexFile" added. You can add up to 5 annexes.

==> SendDocumentRequest impacted

22/10/2020 version 1.6:

- The Language tag becomes optional. 'bi' value is accepted for the Language tag.

==> SendDocumentRequest impacted

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.

4.2 TECHNICAL PREREQUISITES

The technical prerequisites are described in the Social Security API catalog:

<https://portal-int.api.socialsecurity.be/api-catalog-internet>

4.2.1 Security

The service has a single endpoint:

* `be/socialsecurity/registeredmail/v1/RegisteredMail_v1.wsdl` [X.509 Token Profile]

Duplicated WSDL files with explicit policy names

* `be/socialsecurity/registeredmail/v1/RegisteredMail-X509_v1.wsdl` [X.509 Token Profile]

To call the WS you must sign your soap request with a *QuoVadis* X509 certificate.

This certificate identifies the institution and the application.

Attention, you need a distinct certificate for each work environment (Acceptance, Production).

The type of certificate requested must be a non-public trust (applicative) certificate.

The certificate format MUST respect the following structure:

Field	Value	Comment / Example
C	BE	C = BE
O	ORGANIZATIONNAME	The legal name of your organization. Example:

		O = SMALS
OU	Belgian Federal Government	OU = Belgian Federal Government
OU	urn:be:fgov:kbo-bce:organization:cbe-number:xxxxxxxxx	Example: for institution with CBE 01234567689 this OU must be: OU = urn:be:fgov:kbo-bce:organization:cbe-number:0123456789
OU	ACC	Possible values: PRD, ACC, INT. Example: OU = ACC
CN	ApplicationID-URN-from-request-document	Example: for application *werkkaart*/*carte de travail* (RVA/Onem), CN must be: **CN** = employment:job-attest:werkkaart
L	Brussels	The city where your organization is located. Example: L = Sint-Gillis
S	Brussel-Hoofdstad	The state/region where your organization is located. This shouldn't be abbreviated. Example: S = Brussel-Hoofdstad

If necessary, the CSR can be sent to us for validation before the order.

Do not forget to have your onboarding form validated (by eBoxIntegration@smals.be) before ordering your certificates.

5. DESCRIPTION OF THE SERVICE OPERATIONS

The service consists of following operations :

Operations	Description
getStatus	This operation allow s an authenticated sender to obtain the proof of dispatch for a previously sent registered mail item
healthCheck	Standard monitoring method. Three level of monitoring : PING, DEFAULT et DEEP
sendDocument	This operation allow s an authenticated sender to send a document as a registered mail item

5.1 OPERATION GETSTATUS

5.1.1 Request message construction

This function allows the sender to obtain the status of his registered mail item.

Request message containing a ticket number for which the sender wants the proof of dispatch

sequence

<> **Ticket** (Ticket)

The ticket number that references a registered mail item internally within RegisteredMail

5.1.1.1 Example

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/registeredmail/v1" xmlns:v11="http://socialsecurity.be/registeredmail/types/v1">
  <soapenv:Header/>
  <soapenv:Body>
    <v1:GetStatusRequest>
      <v11:Ticket>001000KG4GECZ</v11:Ticket>
    </v1:GetStatusRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

5.1.2 Reply message interpretation

Response message containing the document status and the rproof of dispatch if available

sequence

<> **Ticket** (Ticket)

The ticket number that references a registered mail item internally within RegisteredMail

<> **Token** (Token)

The token sent by the user in the send request, which serves as a user provided identifier

<> **Status** (Status)

The status of the registered mail item

<> [0..1] **ProofOfDispatch** (SignedFile)

The proof of dispatch

5.1.2.1 Example

```
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns5:GetStatusResponse xmlns:ns2="http://socialsecurity.be/errors/v1"
xmlns:ns3="http://socialsecurity.be/registeredmail/types/v1" xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1"
xmlns:ns5="http://socialsecurity.be/registeredmail/v1">
      <ns3:Ticket>1ZG0000036L3Z</ns3:Ticket>
      <ns3:Token>testToken_sha-256</ns3:Token>
      <ns3:Status>PENDING</ns3:Status>
    </ns5:GetStatusResponse>
  </S:Body>
</S:Envelope>
```

5.1.3 Error codes

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
EPOST_T01	Internal Server Error
EPOST_T01	Unauthorized
EPOST_T01	Business Error
REPO_T01	Repository error
REPO_T02	Repository validation error
REPO_B01	Ticket provided does not match Registered Mail Item

5.2 OPERATION HEALTHCHECK

5.2.1 Request message construction

<> **type** (HealthCheckType)

5.2.1.1 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>
```

5.2.2 Reply message interpretation

sequence

<> **Status** (Status)

<> **Component** (Component)

<> **Location** (Location)

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

<> **timestamp** (dateTime)

5.2.2.1 Example

```

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
      <ns3:HealthCheckResponse timestamp="2021-07-29T11:08:00.180+02:00" xmlns="http://socialsecurity.be/registeredmail/types/v1"
xmlns:ns2="http://socialsecurity.be/errors/serviceprovider/v1" xmlns:ns3="http://services.fgov.be/monitoring/v1"
xmlns:ns4="http://socialsecurity.be/errors/v1" xmlns:ns5="http://socialsecurity.be/registeredmail/v1">
        <ns3:Status>
          <ns3:Level>OK</ns3:Level>
        </ns3:Status>
        <ns3:Component>
          <ns3:Name>epost service ws</ns3:Name>
          <ns3:Version>1.6.0</ns3:Version>
          <ns3:MavenReference artifactId="epost-service-ws" groupId="be.smals.epost"/>
          <ns3:CatalogPart/>
        </ns3:Component>
        <ns3:Location>
          <ns3:Environment>OTHER</ns3:Environment>
          <ns3:Host>epost-service-master-9-qb54r</ns3:Host>
          <ns3:Port>8080</ns3:Port>
          <ns3:URL>http://epost-service-master.int.paas.socialsecurity.be/app006ws/RegisteredMail/v1/RegisteredMailService</ns3:URL>
        </ns3:Location>
        <ns3:SanityCheck description="CheckePost database" id="EPOST-001" type="DEFAULT">
          <ns3:Status>
            <ns3:Level>OK</ns3:Level>
          </ns3:Status>
          <ns3:Resource name="ePost DB" reference="jdbc/epost" type="DB">
            <ns3:Location>
              <ns3:Environment>DEV</ns3:Environment>
              <ns3:Host>CRSTST1-SCAN.dataserv.be</ns3:Host>
              <ns3:Port>2003</ns3:Port>
              <ns3:Name>D4EPOST</ns3:Name>
              <ns3:URL>jdbc:oracle:thin:@CRSTST1-SCAN.dataserv.be:2003/D4EPOST</ns3:URL>
            </ns3:Location>
          </ns3:Resource>
          <ns3:TimeInMillis>340</ns3:TimeInMillis>
        </ns3:SanityCheck>
      </ns3:HealthCheckResponse>
    </soap:Body>
  </soap:Envelope>

```

5.2.3 Error codes

Not applicable

5.3 OPERATION SENDDOCUMENT

This operation allows a requester to send a document as registered mail.

5.3.1 Request message construction

Message format for a DocumentRequest

sequence

- <> **RegisteredMailItem** (RegisteredMailItem)
Envelope containing the document and metadata
- <> [0..1] **CheckEboxStatus** (CheckEboxStatus)

Default = ACTIVATED. Enumeration of the possible values {BYPASS ; ACTIVATED ; PREFERRED}. Interpretation: BYPASS = the system will not check if the recipient ebox profile is already activated or not; ACTIVATED = the recipient ebox profile must be activated ; PREFERRED = the recipient ebox profile must be activated and the recipient has chosen ebox as its preferred communication channel.

5.3.1.1 Example

```

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:v1="http://socialsecurity.be/registeredmail/v1"

```

```

xmlns:v11="http://socialsecurity.be/registeredmail/types/v1 "
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<soapenv:Header/>
<soapenv:Body>
  <v1:SendDocumentRequest>
    <v11:RegisteredMailItem>
      <v11:Token>testToken_sha-256</v11:Token>
      <v11:Recipient xsi:type="v11:Enterprise">
        <v11:CompanyID>0861023666</v11:CompanyID>
        <v11:QualityCode>QUAL_FSC</v11:QualityCode>
      </v11:Recipient>
      <v11:Document>
        <v11:Title>Registered mail example</v11:Title>
        <v11:Digest>
          <v11:DigestMethod>SHA-256</v11:DigestMethod>
        </v11:Digest>
        <v11:DigestValue>Y5/G6JbrPF3sXafd8qS/s51b5vdDVHgmZVX2H+5q0t8=</v11:DigestValue>
        <v11:Digest>
          <v11:DownloadFileName>FileName.pdf</v11:DownloadFileName>
          <!--Optional:-->
          <!--<v11:MIME>application/pdf</v11:MIME>-->
          <v11:Type>EDER501</v11:Type>
          <v11:Language>fr</v11:Language>
          <v11:Content>cid:472982604608</v11:Content>
        </v11:Document>
        <!--you can add 0 to 5 annexes-->
        <v11:Annex>
          <v11:Content>cid:472982604609</v11:Content>
          <v11:Title>Annex title</v11:Title>
          <v11:Digest>
            <v11:DigestMethod>SHA-256</v11:DigestMethod>
            <v11:DigestValue>5juJmzTzWwDC4KZQcFuCRR/0YSuX5RUvMw4d8iFQ4=</v11:DigestValue>
          </v11:Digest>
          <v11:DownloadFileName>annexe.pdf</v11:DownloadFileName>
          <v11:MIME>application/pdf</v11:MIME>
        </v11:Annex>
      </v11:RegisteredMailItem>
    </v1:SendDocumentRequest>
  </soapenv:Body>
</soapenv:Envelope>

```

5.3.2 Reply message interpretation

Response message containing the receipt. This can be used as proof that RegisteredMail has treated the request.

sequence

<> **Receipt** (Receipt)

The receipt confirming that the document was sent

5.3.2.1 Example

```

<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Body>
    <ns5:SendDocumentResponse xmlns:ns2="http://socialsecurity.be/errors/v1 "
xmlns:ns3="http://socialsecurity.be/registeredmail/types/v1 " xmlns:ns4="http://socialsecurity.be/errors/serviceprovider/v1 "
xmlns:ns5="http://socialsecurity.be/registeredmail/v1 ">
      <ns3:Receipt>
        <ns3:Ticket>1ZG0000036L3Z</ns3:Ticket>
        <ns3:Token>testToken_sha-256</ns3:Token>
        <ns3:Document>
          <ns3:Title>Receipt</ns3:Title>
          <ns3:Digest>
            <ns3:DigestMethod>SHA-256</ns3:DigestMethod>
          </ns3:Digest>
          <ns3:DigestValue>8C3Q59LOG0Mx52pzCJXqjoBcFwpim3OFaN1EMfvR6Ys=</ns3:DigestValue>
          </ns3:Digest>
          <ns3:DownloadFileName>Receipt.xml</ns3:DownloadFileName>
          <ns3:MIME>text/xml</ns3:MIME>
          <ns3:Type>RECEIPT</ns3:Type>
          <ns3:Language>en</ns3:Language>
          <ns3:Content>cid:18ee6a91-a70a-4cae-b0a3-3156129b7719@example.jaxws.sun.com</ns3:Content>
        </ns3:Document>
      </ns3:Receipt>
    </ns5:SendDocumentResponse>
  </S:Body>
</S:Envelope>

```


GENERIC_XML_T02	Cannot build Request from GenericXmlDocument
GENERIC_XML_T03	Cannot generate xml from GenericXmlDocument
GENERIC_XML_B01	GenericXml does not contain all required data
REPO_T01	Repository error
REPO_T02	Repository validation error
FILE_T01	No such digest method algorithm
FILE_T02	Cannot persist file content
FILE_T03	Cannot read file content
FILE_B01	Digest doesn't match document.
FILE_B02	Digest method not accepted (only SHA-1, SHA-256 and SHA-512)
FILE_B03	The SHA-1 digest method cannot be used anymore
SEPIA_T01	Sepia system error
SEPIA_T02	Could not sign receipt
TNG_T01	TNG unreachable - system error
TNG_T02	Cannot create ticket
JMS_T01	Cannot send JMS message
JMS_T02	Problem getting JMS queue
JMS_T03	Problem getting JMS connection factory
JMS_T04	Unable to set JMS message properties
JMS_T05	Unable to get JMS message properties
JMS_T06	Cannot read JMS message
JMS_T07	Cannot build JMS message
JMS_B01	JMS message properties are missing

6. COMMON TYPES

6.1 MONITORING-V1

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

Version:

6.1.1 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)

6.1.2 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	

6.1.3 Extensions

«XSDcomplexType»

An extension point to allow applications to add additional information

sequence

6.1.4 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

6.1.5 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)

6.1.6 MavenReference

«XSDcomplexType»

Describes the artifact as catagorized by maven.

<> **groupId** (string)

<> **artifactId** (string)

6.1.7 Resource

«XSDcomplexType»

The resource that is checked in a sanity check.

sequence

- <> [0..1] **Location** (Location)
- <> **name** (string)
- <> **type** (ResourceType)
- <> **reference** (string)

6.1.8 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	

6.1.9 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)

6.1.10 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)

6.1.11 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.	

6.2 REGISTEREDMAIL_TYPES_V1

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

Version: 1.6

6.2.1 AnnexFile

«XSDcomplexType»

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

sequence

<> **Content** (swaRef)

<> **Title** (Title)

<> **Digest** (Digest)

<> **DownloadFileName** (FileName)

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

6.2.2 CBE

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	pattern	[0-9]?[0-9]{9}	
annotation	The KBO number of the enterprise		

6.2.3 CheckEboxStatus

«Enumeration»

facets	value	annotation
	BYPASS ACTIVATED PREFERRED	
annotation	Enumeration {BYPASS ; ACTIVATED ; PREFERRED}: possible values in order to check the status of the recipient eBox. Interpretation: BYPASS = the system will not check if the recipient ebox profile is already activated or not; ACTIVATED = the recipient ebox profile must be activated ; PREFERRED = the recipient ebox profile must be activated and the recipient has chosen ebox as its preferred communication channel.	

6.2.4 Citizen

«XSDcomplexType»

A citizen

sequence

<> **INSS** (INSS)

The National Number of the citizen

<> **QualityCode** (QualityCode)

The value is "CITIZEN"

6.2.5 Digest

«XSDcomplexType»

Digest calculated on the File

sequence

<> **DigestMethod** (DigestMethod)

<> **DigestValue** (DigestValue)

6.2.6 DigestMethod

«Enumeration»

facets	value	annotation
	SHA-1 SHA-256 SHA-512	SHA-1 will be deprecated as of January 1st, 2018
annotation	Digest Algorithm (SHA-1, SHA-256, SHA-512).	

6.2.7 DigestValue

«XSDsimpleType»

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

6.2.8 DocumentRequest

«XSDcomplexType»

Message format for a DocumentRequest

sequence

<> **RegisteredMailItem** (RegisteredMailItem)

Envelope containing the document and metadata

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

Default = ACTIVATED. Enumeration of the possible values {BYPASS ; ACTIVATED ; PREFERRED}. Interpretation: BYPASS = the system will not check if the recipient ebox profile is already activated or not; ACTIVATED = the recipient ebox profile must be activated ; PREFERRED = the recipient ebox profile must be activated and the recipient has chosen ebox as its preferred communication channel.

6.2.9 DocumentResponse

«XSDcomplexType»

Response message containing the receipt. This can be used as proof that RegisteredMail has treated the request.

sequence

<> **Receipt** (Receipt)

The receipt confirming that the document was sent

6.2.10 Enterprise

«XSDcomplexType»

An enterprise in UMan

sequence

<> **CompanyID** (CBE)

The KBO number of the enterprise

<> **QualityCode** (QualityCode)

The quality under which the entity is known

6.2.11 Entity

«XSDsimpleType»

type				
facets	<table border="1"> <tr> <td>kind</td> <td>value</td> <td>annotation</td> </tr> </table>	kind	value	annotation
kind	value	annotation		
annotation				

6.2.12 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

<> **Content** (swaRef)

Reference to the contents in the attachment

6.2.13 FileDescription

«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)

<> **Digest** (Digest)

<> **DownloadFileName** (FileName)

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

<> **Type** (Type)

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

6.2.14 FileName

«XSDsimpleType»

type	
-------------	--

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

6.2.15 INSS

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	length	11	
annotation	The National Number of the citizen		

6.2.16 Language

«Enumeration»

facets	value	annotation
	fr	French
	nl	Dutch
	de	German
	en	English
	bi	Bilingual
annotation	Language, restricted to the values FR for french, NL for dutch, DE for German	

6.2.17 Meta

«XSDcomplexType»

Meta data.

sequence

<> **Type** (ShortString)

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

6.2.18 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.

6.2.19 QualityCode

«XSDsimpleType»

type													
facets	<table border="1"> <thead> <tr> <th>kind</th> <th>value</th> <th>annotation</th> </tr> </thead> <tbody> <tr> <td>derivation</td> <td>restriction</td> <td></td> </tr> <tr> <td>maxLength</td> <td>125</td> <td></td> </tr> <tr> <td>minLength</td> <td>1</td> <td></td> </tr> </tbody> </table>	kind	value	annotation	derivation	restriction		maxLength	125		minLength	1	
kind	value	annotation											
derivation	restriction												
maxLength	125												
minLength	1												
annotation	For Enterprises, Quality code from the userManagement of social security. Ex : QUAL_EMP_NOSS, QUAL_SSC, QUAL_SP_IND, ... and for Citizens, the value to give is "CITIZEN"												

6.2.20 Receipt

«XSDcomplexType»

The receipt confirming that the document was sent

sequence

<> **Ticket** (Ticket)

The ticket number that will reference the registered mail item internally within RegisteredMail

<> **Token** (Token)

The token sent by the user in the send request, which serves as a user provided identifier

<> **Document** (SignedFile)

6.2.21 ReceiptContent

«XSDcomplexType»

The contents of the SignedFile in the receipt

sequence

<> **Ticket** (Ticket)

The ticket number that will reference the registered mail item internally within RegisteredMail

<> **Token** (Token)

The token sent by the user in the send request, which serves as a user provided identifier

<> **Recipient** (Entity)

The recipient of the document

<> **Sender** (Sender)

The sender of the document

<> **Document** (FileDescription)

FileDescription of the document that was sent as registered mail

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

FileDescription of an annex that was sent as registered mail

6.2.22 RegisteredMailItem

«XSDcomplexType»

Envelope containing the document and metadata

sequence

<> **Token** (Token)

A token, permitting the sender to identify the document

<> **Recipient** (Entity)

The recipient of the document

<> **Document** (File)

The document to be sent as registered mail

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

An annex linked to the document

<> [0..100] **Meta** (Meta)

Meta data linked to registered mail

6.2.23 Sender

«XSDcomplexType»

A Sender is identified with a CompanyID and an ApplicationID (URN).

sequence

<> **CompanyID** (CBE)

<> **ApplicationID** (SenderApplicationID)

6.2.24 SenderApplicationID

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	maxLength	150	
annotation	The applicationID extracted from the subject of the sender certificate (value of the CN tag).		

6.2.25 ShortString

«XSDsimpleType»

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

6.2.26 SignedFile

«XSDcomplexType»

Extension of file containing a signature by SepiaDocumentSigner

sequence

<> **Signature** (base64Binary)

The signature generated by SepiaDocumentSigner for the File content

6.2.27 Status

«Enumeration»

facets	value	annotation
	PENDING	
	SENT	
	ERROR	
annotation	The status of the registered mail item	

6.2.28 StatusRequest

«XSDcomplexType»

Request message containing a ticket number for which the sender wants the proof of dispatch

sequence

<> **Ticket** (Ticket)

The ticket number that references a registered mail item internally within RegisteredMail

6.2.29 StatusResponse

«XSDcomplexType»

Response message containing the document status and the rproof of dispatch if available

sequence

<> **Ticket** (Ticket)

The ticket number that references a registered mail item internally within RegisteredMail

<> **Token** (Token)

The token sent by the user in the send request, which serves as a user provided identifier

<> **Status** (Status)

The status of the registered mail item

<> [0..1] **ProofOfDispatch** (SignedFile)

The proof of dispatch

6.2.30 Ticket

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	length	13	
	pattern	[0-9A-HJ-NP-Z]{13}	
annotation	A ticket number as provided by the TicketNumberGenerator		

6.2.31 Title

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	maxLength	400	
	minLength	1	
annotation	The title of the File		

6.2.32 Token

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	maxLength	50	
annotation	A generic token		

6.2.33 Type

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	restriction	
	maxLength	80	
	minLength	1	
annotation	The document type.		

6.3 SWAREF

Namespace: <http://ws-i.org/profiles/basic/1.1/xsd>

Version:

6.3.1 swaRef

«XSDsimpleType»

type			
facets	kind	value	annotation
	derivation	Restriction	
annotation			

7. COMMON ERROR CODES

7.1 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.

7.2 BUSINESS ERROR CODES

These error codes are defined by the application.

Code	Description	Cause	Explication

8. ANNEX

8.1 ComplexType Digest

Functional description

The Digest type contains all necessary information to verify the digest of a document. It has two child items:

- DigestMethod: The algorithm that was used to calculate the DigestValue:
- DigestValue: The digest calculated with the DigestMethod

8.2 ComplexType Enterprise

Functional description

The Enterprise type defines a recipient who is identified as an enterprise. It has two child items:

- CompanyID: The enterprise number, assigned by the CBE, which uniquely identifies an employer, be it a natural person, a group of natural persons or legal person
- QualityCode: The quality under which the entity is known

8.3 ComplexType Entity**Functional description**

The Entity type defines a generic recipient

8.4 ComplexType File**Functional description**

The File complexType extends the FileDescription complexType. In addition, it contains a content element, which is a URI that references the document in attachment (using swaRef standard).

- Content: Contents of the document in an attachment

8.5 ComplexType FileDescription**Functional description**

The FileDescription type contains all the information of a document without containing the actual contents

- Title: Title of the sent document.
- Digest: In order to check the integrity of the message content, the sender must add the hash of the document (DigestValue). The WS will then verify that the provided digest is the same as the document digest computed according to the provided method (SHA-1; SHA-256 or SHA-512).
- DownloadFileName: name used to open or save the document.
- MIME [optional]: MIME type of the document
- Type: The document type
- Language: The language of the document

8.6 ComplexType Meta**Functional description**

The meta type permits to send further specific information about the document under the form of type-value pairs

- Type: the type of the meta data
- Value: the values of the metadata (up to 10)

8.7 ComplexType Receipt**Functional description**

The Receipt type contains the receipt the sender gets after calling the sendDocument operation. It contains three child items:

- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Ticket: The ticket number of this request which identifies this registered mail item and can serve for future reference (e.g. to make a status request)

- Document: The receipt which serves as the proof for the sender that he in fact has sent this registered mail item

8.8 ComplexType ReceiptContent

Functional description

The ReceiptContent type contains the information contained in the content element of the Receipt type.

- Ticket: The ticket number of this request which identifies this registered mail item and can serve for future reference (e.g. to make a status request)
- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Recipient: The recipient of the registered mail item
- Sender: The subject of the x509 certificate of the sender
- Document: The FileDescription of the RegisteredMailItem that was sent

8.9 ComplexType RegisteredMailItem

Functional description

The RegisteredMailItem type identifies a registered mail item. It has three child items:

- Token: The token that was included in the sendDocument request which serves as the identifier for the user
- Document: The document to send as a registered mail item
- Recipient: The recipient of the document
- Meta: Metadata linked to the document

8.10 ComplexType SignedFile

Functional description

The SignedFile complexType extends the File complexType. In addition, it contains a signature element, which is the base64 encoded signature produced by Sepia for the document in Content.