COMMUNICATION AVEC LA BCSS

La préoccupation de l'informatique de la BCSS est d'être interopérable avec ses partenaires.

Elle a opté en 2006 pour une migration de sa plateforme mainframe (zOS) vers une plateforme distribuée (Linux) pour lui permettre d'adopter des standards ouverts non liés à un constructeur. L'emploi de ces différentes normes implique l'adoption d'une série de règles et de comportement de « bonnes pratiques » pour assurer une compatibilité maximum avec les partenaires de son réseau.

L'objectif de ce document est d'aborder les divers choix retenus par la BCSS. Le fil rouge est, en premier lieu, de survoler les différents domaines et, dans une seconde phase, de fournir des détails techniques.

MÉTHODOLOGIE

Nous nous inscrivons dans une approche itérative dans laquelle nous veillons à concilier les besoins du demandeur et ceux de la BCSS. Par exemple, nous voulons disposer d'une architecture robuste et consolider les investissements en matériel et en développement.

La méthodologie est celle du 'Two Track Unified Process' symbolisé par les deux branches de l'Y afin d'arriver à une conception qui tient compte des aspects fonctionnels et non fonctionnels. Elle est également itérative.

Méthodologie

D'autre part, nous nous engageons résolument dans une approche orientée services.

Quatre étapes pour établir une architecture de service.
Du processus des trois premières, émerge un modèle et une direction pour la quatrième.

1. **WHAT** :
    - Quel est le périmètre du service ?
    - Qu'attend-t-on de ce service ?
2. **WHO** :
    - Qui sont les acteurs externes intervenant avec le service ?
    - Quels sont les services en interactions ?
3. **WHY** :
    - Identifier pourquoi un service interagit avec les autres ?
    - Pourquoi des acteurs externes interviennent ?
4. **HOW** :
    - le détail sur les services et les processus coordonnés
    - ainsi que le détail de l'implémentation du service.

Le développement d'un projet (Etape 4) s'appuie sur le document « Projet Initiation Document » approuvé par le donneur d'ordre, le sponsor ainsi que les parties concernées dans le projet. Il a pour but d'établir un contrat sur les objectifs à atteindre, les informations à communiquer, celles à recevoir, les actions à prendre ainsi que le planning des tâches.

Ce document contient les différentes facettes d'un projet, en utilisant un vocabulaire constant mais dont le sens recouvre une vue particulière suivant la discipline couverte.

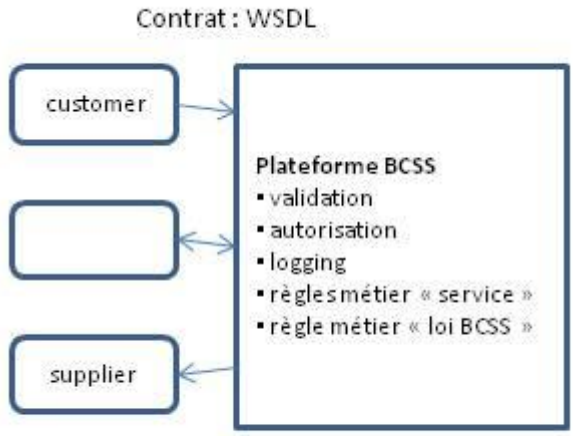
Exemple, le mot « service »

- au niveau du besoin du client : c'est fournir, par exemple, la liste des droits d'une personne,
- au niveau métier : c'est fournir une structure d'éléments structurés par entité,
- au niveau technique : c'est publier un service avec des opérations et des messages SOAP suivant des schémas de définitions (WSDL, XSD),
- au niveau développement : cela peut-être un composant métier ou un utilitaire, par exemple de conversion
- au niveau déploiement : c'est une version d'un ensemble de composants cohérents avec des dépendances.

VUE DU CONTEXTE

Les échanges de la plateforme BCSS se subdivisent en deux groupes : le premier comprend ceux développés pour répondre à des besoins exprimés par des clients et le second, ceux des sources authentiques responsables de leur gestion.

Le rôle de la BCSS est de faciliter les échanges et ou les encourager entre les partenaires de la BCSS tout en veillant au respect de proportionnalité, de confidentialité et des autorisations accordées par le comité sectoriel. La BCSS peut également agréger diverses informations en provenance des partenaires de la sécurité sociale afin de répondre à des projets en appui d'une initiative gouvernementale.



BESOINS DU CLIENT

Nous attendons de notre partenaire l'expression de ses besoins fonctionnels vis-à-vis du réseau de la Sécurité Sociale. La base de cette demande est évidemment fondée sur la législation et étayée par des textes réglementaires. De plus, l'accord du comité sectoriel doit être sollicité et obtenu avant toute mise en production.

Pour exprimer ces besoins, en plus de phrases décrivant les objectifs (éviter de parler en terme de solution et, ou d'employer des termes techniques réduisant la vue à un contexte particulier), l'ajout de diagramme permet de visualiser la problématique. L'ensemble : texte et graphique, contribue à une même compréhension du sujet par l'ensemble des acteurs.

Divers diagrammes permettent de comprendre le périmètre du projet, clarifier l'objectif et mesurer l'impact chez les acteurs. Par exemple, le SYSTEM CONTEXT précise les différents acteurs, leur mode d'interaction (haut niveau) ainsi que leur responsabilité respective. Le diagramme 'bassin de natation' permet de décrire les fonctionnalités sous forme d'activités sous le contrôle de chacun des acteurs.

LE CONTRAT DE L'ÉCHANGE D'UN SERVICE À RENDRE

Techniquement, après une réflexion itérative, nous aboutirons à l'élaboration d'un SERVICE comportant une ou plusieurs OPERATIONS, chacune ayant un contenu précis d'une requête [REQUEST] et d'une réponse [RESPONSE]. C'est en fait, le contrat de ce qui sera échangé entre les plateformes « client » et « BCSS ».

La réponse « métier » décrit à la fois les divers types de réponses du service demandé. Elle peut être positive ou négative.

Une distinction supplémentaire est de séparer les erreurs « métiers » des erreurs « techniques ».

Une erreur « métier » survient lorsque le service à rendre n'a pas pu être rendu pour des raisons métiers : un non respect de règles métiers ou simplement, les données demandées n'existent pas. Ce sont des exceptions prévisibles et l'application sait comment poursuivre sa logique.

Une erreur « technique », par contre, devrait être exceptionnelle et couvrir les incidents dont l'application peut tenir compte mais n'est pas capable de régler.

Ces problèmes sont détectés lors des développements (validation des schémas) ou lors de l'exécution (serveur pas disponible, schéma modifié ou pas à jour, …)

Exemples : la non-conformité à la définition de la structure du message :

- élément obligatoire non fourni,
- taille d'un contenu non respectée,
- etc. …

BESOINS DE LA BCSS

La BCSS doit pouvoir être capable de respecter les décisions du comité sectoriel.

Elle doit pouvoir authentifier l'organisation cliente, et, ou l'application cliente et, ou l'utilisateur à la base de la demande. Elle doit obtenir un feu vert pour permettre l'accès à l'application pour le client authentifié. L'application de la BCSS fournira le service demandé tout en filtrant éventuellement les informations hors du champ d'application réglementaire qu'elle obtient des sources authentiques.

Elle doit également prendre une trace de ce qui est échangé (quand, qui, quoi).

ASPECTS TECHNIQUES

Nous n'oublierons pas de tenir compte, en plus des besoins fonctionnels, des besoins non fonctionnels qui nous permettront de passer d'un modèle du maître d'ouvrage à celui d'un modèle d'exécution. En d'autres termes, nous devons prévoir un scénario en temps réel qui tient compte des exceptions.

On ajoute ainsi pour chaque OPERATION un contenu lié aux exceptions techniques. Par exemple, si le message reçu n'est pas conforme au schéma ou que le service n'est pas disponible. Remarque importante : le client peut avoir une exception sans ce message lorsque le problème arrive dans une couche inférieure comme lors de l'établissement de session http : Code erreur HTTP 400 / 500 / etc avant que le niveau applicatif n'ait la main.

WSDL

La BCSS suit la convention suivante, le WSDL (1.1) contient la définition des opérations sous la forme <*verbObject*XXX> et repris au niveau des paramètres.

Service operation		<verbObject>	
	request	**<*verbObject*Request>**	***liés au métier (fonctionnel)***
	response	**<*verbObject*Response>**	***positive ou négative***
	fault	**<*verbObject*Fault>**	***Pré-requis non respectés***
			Système indisponible
			Etc

Pour être compliant au WS-I (Web Service compatibility), nous avons opté pour les options suivantes pour ce contrat :

- Etre au sein d'un message SOAP
- Le style est 'document' et l'encoding (use=)'litteral'
- L'élément '<Envelope>**<Body>**' ne contient qu'un seul élément
  Suivant le stade de l'échange :

  →<*verbObject*Request>
  ←<*verbObject*Response>'
  ← <Fault><Detail><*verbObject*Fault>

- Le message <*verbObject*Fault> est le fils de l'élément '<Envelope><Body><Fault>**<Detail>'**

SCHÉMA

Le schéma associé au WSDL reprend la définition des éléments sous forme de **SimpleType** ou **ComplexType** avec la convention suivante :

- Le nom de ces «<Type » respecte la casse : « Pascal case » :

  Exemple <xs :SimpleType name='***GenderType' …>***

- Les noms des éléments et des attributs suivant la casse « camel case »

  Exemple ***<gender>F</gender>***

Le namespace est **http://kszbcss.fgov.be/intf/<*ServiceName*>/v<n>** : Service name en Pascal case.

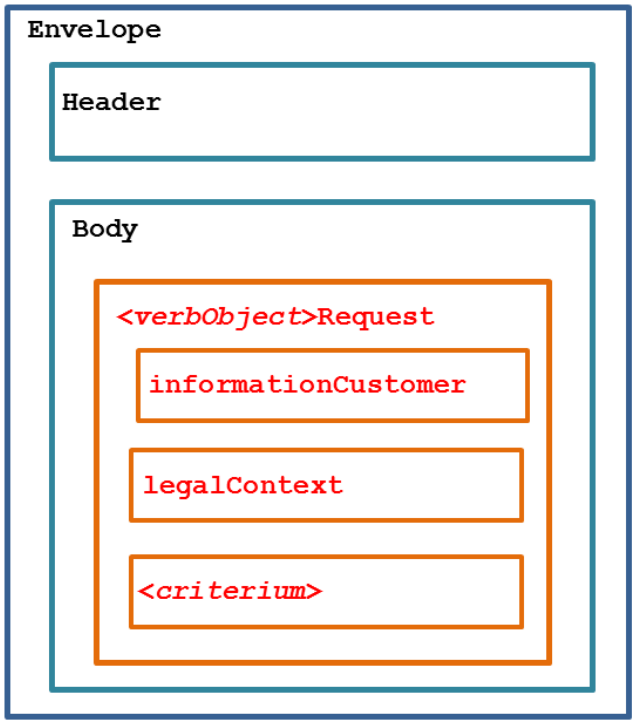
Nous évitons la technique « salami » qui consiste à employer l'option 'ref='.

Nous préconisons le premier des deux design patterns :
- Venitian Blind Design : usage de Complex et Simple Types globaux.
- Russian Doll Design : usage de déclarations locales.

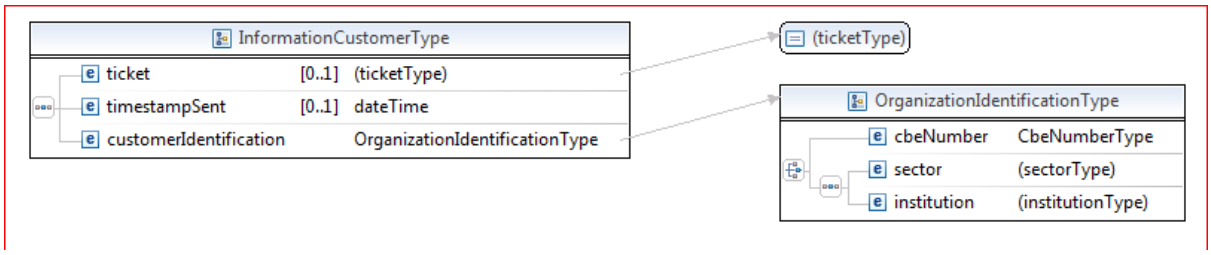
MESSAGE REQUÊTE

Chaque message métier est situé sous l'élément 'Body' d'un message SOAP.

On retrouvera donc, un élément portant le nom « verbObject »Request avec des éléments « enfants »

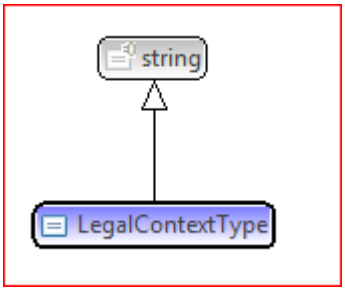


L'élément **informationCustomer** est fourni par le client en vue de s'identifier au niveau métier en fournissant son identification soit au niveau du réseau de la sécurité sociale, soit au niveau entreprise. Il peut contenir des références temporelle et métier.



L'élément **legalContext** détermine le cadre légal de l'utilisation du service.

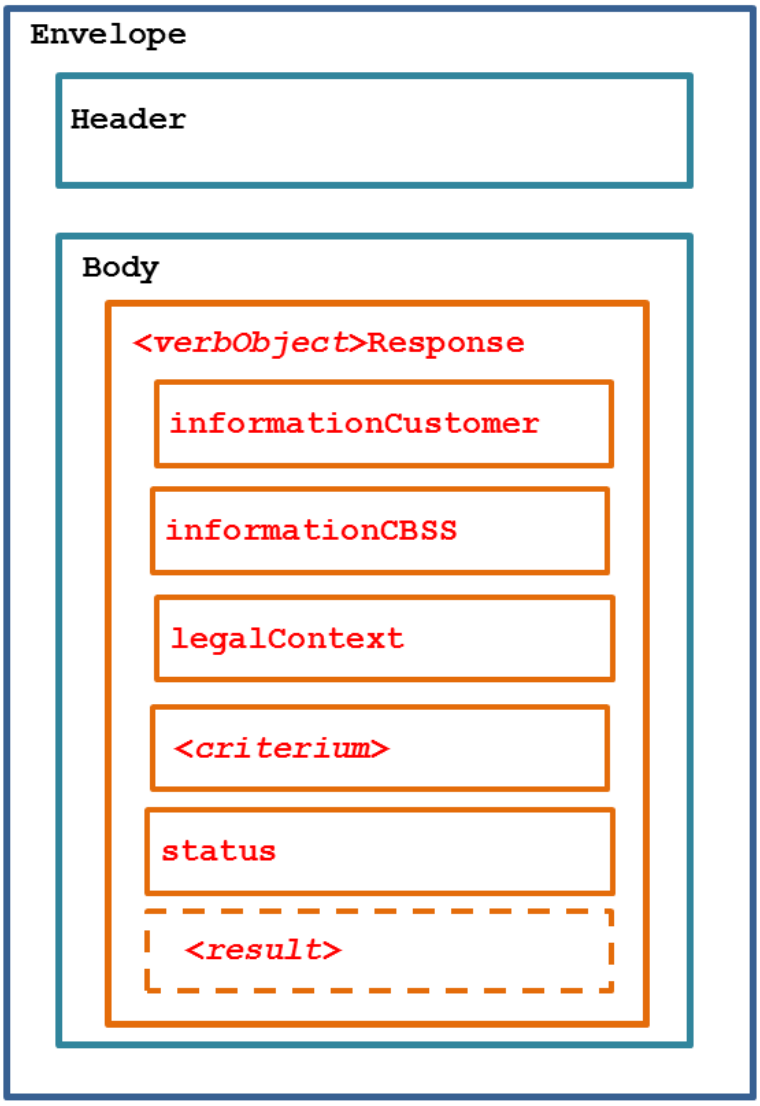
Cet information peut servir pour filtrer ou étendre le champ du service suivant les autorisations du comité sectoriel.



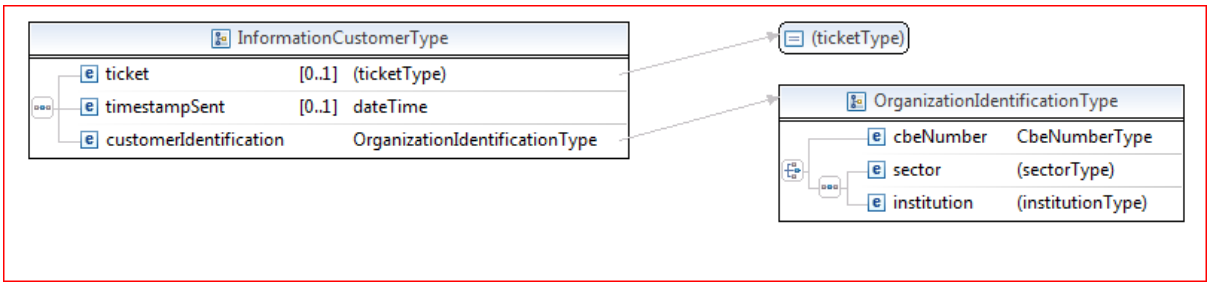
MESSAGE RÉPONSE

Chaque message métier est situé sous l'élément 'Body' d'un message SOAP.

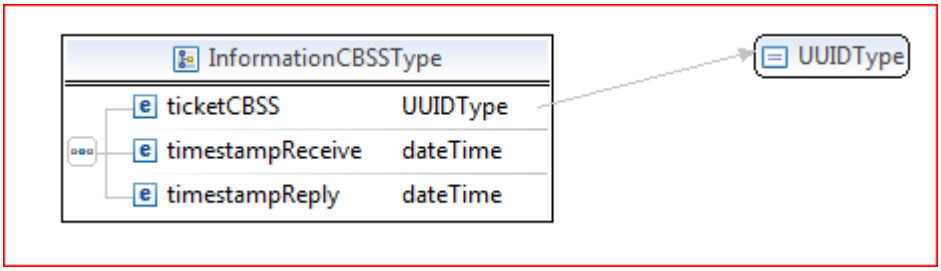
On retrouvera donc, un élément portant le nom « verbObject »Response



L'élément **informationCustomer** original est repris tel quel. Il permet au client de restituer le lien avec sa requête.



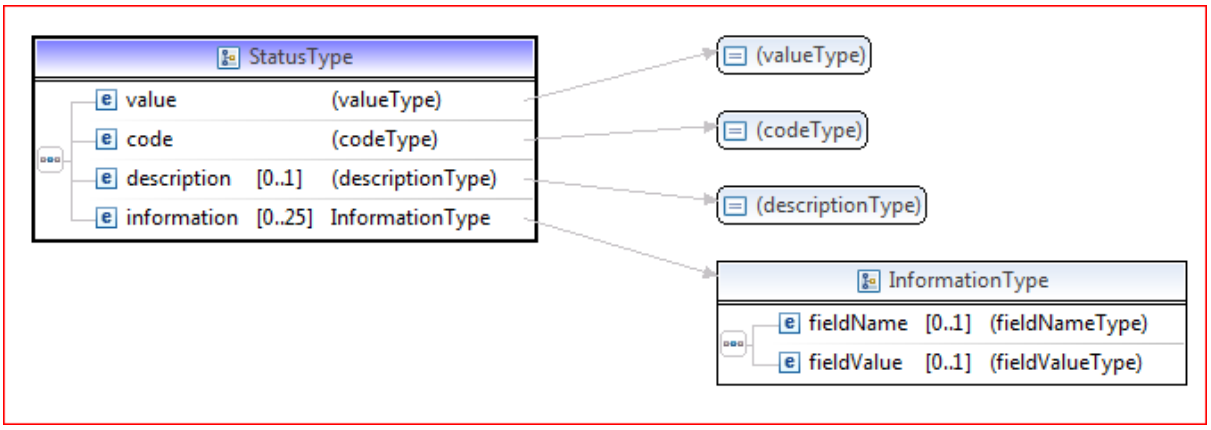
L'élément **informationCBSS** est produit par la BCSS en vue des recherches ultérieures dans les logging.



L'élément **legalContext** original est repris tel quel.

L'élément **<critérium>** original est repris tel quel.

L'élément **status** permet de qualifier la réponse métier : positif ou négatif et, éventuellement, des explications complémentaires.



En cas de réponse métier positive, un élément <**result**> donne le résultat demandé.

SESSION HTTP VS HTTPS

Nous préconisons l'usage d'une session sécurisée SSL/TLS[1] pour les environnements de développement, d'acceptation et de production pour se prémunir d'appel de serveurs non prévus ou d'une inversion d'environnement.

Au niveau de la BCSS nous vérifions si le certificat reçu du partenaire appartient à la liste de confiance (environnement approprié et autorité d'enregistrement de confiance) ainsi que l'autorisation d'accès au service appelé par l'organisation cliente.

AUTHENTIFICATION DE L'APPELANT (APPLICATION CLIENTE, …)

Nous préconisons l'adoption des normes WS-Security [WS-SEC v1.0/1.1] pour authentifier le client dans un SOAP 1.1. Pour les WS avec fichiers attachés, SOAP 1.2 est requis pour suivre le standard MTOM.

Suivant le cas de figure, nous opterons :

- Une signature du contenu <Body>, du certificat et d'un timestamp , avec la communication du certificat de l'application cliente. [X509v3] (cfr ci-après)
- Une assertion d'authentification [SAML v2]
    - soit le "end user » authentifié,
    - soit son appartenance à une organisation (ou group))(=anonyme),
    - soit le « end user » ainsi que ses attributs déclarés,
    - soit le « end user » ainsi que ses attributs certifiés.

Cela permet de disposer de l'organisation qui nous appelle et, dans le deuxième cas, également, l'utilisateur qui s'est connecté à l'application cliente. Ces informations sont nécessaires pour interroger le système UAM afin de respecter la police d'autorisation.

Ces informations sont donc glissées sous l'élément '<Envelope>**<Header>**' du message SOAP et la logique métier n'est pas impactée par cet ajout d'une couche de sécurité.

[1] L'option RENEGOCIATION du protocole TLS est refusée pour corriger la faille de sécurité du MITM (novembre 2009). Les normes SSL V1, V2 et V3 sont désactivées et le TLS 1.2 est préconisé.

NORMALISATION DES CHAMPS DES CERTIFICATS DE LA BCSS

Ces informations concernent les certificats de la BCSS que le partenaire recevra s'il désire nous authentifier. Ces certificats sont également publiés sur le site web de la BCSS.

Il y a deux types de certificats : côté serveur ou côté client. Dans le certificat, le champ 'usage' précise les limites de son utilisation.

1. Le certificat « serveur » d'identification de la BCSS pour les sessions en SSL (transport)

Champ X.509	Valeur	Commentaire
C	BE	
O	Kruispuntbank van de Sociale Zekerheid	*Nom de l'organisation*
ST	Brussel Hoofdstad	
L	Brussels	
OU	KSZ-BCSS	
OU	0244640631	*Numéro BCE-KBO de la BCSS*
OU	TEST ACPT	*Une « Organization Unit » supplémentaire pour les deux environnements de test et d'acceptation.*
CN	b2b-test.ksz-bcss.fgov.be b2b-acpt.ksz-bcss.fgov.be b2b.ksz-bcss.fgov.be	*Les points d'entrées sont distincts par environnement* ***Hostname des serveurs***
TLS server authentication		

Remarques :

- Les informations contenues dans un certificat respectent la structure de l'Abstract Syntax Notation One [ASN.1] et le contenu des champs est IA5String à savoir pas de caractères accentués.
- Le numéro d'entreprise consiste à 10 chiffres consécutifs.

2. Le certificat « client » d'identification de la BCSS pour les sessions en SSL (transport)

Champ X.509	Valeur	Commentaire
C	BE	
ST	Brussel Hoofdstad	
L	Brussels	
O	Kruispuntbank van de Sociale Zekerheid	*Nom de l'organisation*
OU	KSZ-BCSS	
OU	0244640631	*Numéro BCE-KBO de la BCSS*
OU	TEST ACPT	*Une « Organization Unit » supplémentaire pour les deux environnements de test et d'acceptation.*
CN	c-b2b-test.ksz-bcss.fgov.be c-b2b-acpt.ksz-bcss.fgov.be c-b2b.ksz-bcss.fgov.be	*Les sources sont distinctes par environnement (hostname fictif)*
TLS client authentication		

3. Le certificat d'application identifiant la BCSS utilisé pour signer les messages (WS-security x.509 certificate Token Profile 1.0/1.1) (niveau message)

Champ X.509	Valeur	Commentaire
C	BE	
ST	Brussel Hoofdstad	
L	Brussels	
O	Kruispuntbank van de Sociale Zekerheid	*Nom de l'organisation*
OU	KSZ-BCSS	
OU	urn:be:fgov:kbo-bce:organization:cbe-number: 0244640631	*Numéro BCE-KBO de la BCSS*
OU	TEST ACPT	*Une « Organization Unit » supplémentaire pour les deux environnements de test et d'acceptation.*
CN	esb-test.ksz-bcss.fgov.be esb-acpt.ksz-bcss.fgov.be esb.ksz-bcss.fgov.be	*La plateforme qui émet le message* *(Hostname fictif)*
TLS client authentication		

DESCRIPTION DU SUJET DU CERTIFICAT X.509 V3

Le sujet du certificat X.509V3 se décompose en des champs [Relative Distinguish Name] .

Certains obligatoires :

- CN (Common Name),
- C (Country) ,
- O (Organization) ,
- OU (Organization Unit)
- ST (State or Province)
- et L (Locality).

En respect du RFC 2253, nous préconisons de ne pas utiliser les caractères spéciaux (dont la virgule, le signe égal) auxquels nous ajoutons les barres obliques.

USAGE: SESSIONS SSL (SECURED SOCKETS LAYER)

Ces certificats sont utilisés pour établir des sessions sécurisées avec authentification mutuelle.

Ils permettent d'avoir l'encryption des données au niveau du transport et ce, dès l'établissement de la session sécurisée après l'échange des certificats, d'un accord d'un algorithme et des clefs temporaires.

Chacun des partenaires peut s'assurer qu'il dialogue avec celui qu'il croit : le certificat du partenaire a été communiqué préalablement. Cela permet à chacun de dresser la liste des certificats des sites ou des clients en qui il a confiance.

Subject		Certificat pour session SSL

Common Name	CN	CN[2]= b2b[-*environnement ''***test**'' ou ''**acpt**''][3].*nom de domaine*
		CN[4]= c-b2b[-*environnement ''***test**'' ou ''**acpt**''].*nom de domaine*
Country	C	C= ''BE''
Organization	O	O = 'nom de l'organisation ''
Organization Unit	OU	OU1 = *acronyme de l'organisation* OU2 = numéro d'entreprise (10 chiffres consécutifs) [OU3= *environnement ''***TEST**'' ou ''**ACPT**'']

Certificat de type CLIENT	Certificat de type SERVEUR
Key Usage • Digital Signature • Key Encipherment Extended Key Usage • TLS web client authentication	Key Usage • Digital Signature • Key Encipherment Extended Key Usage • TLS web server authentication
Taille des clefs RSA	2048
Signature hash algorithm	sha256

USAGE : AUTHENTIFICATION ET INTÉGRITÉ

Ces certificats servent à authentifier l'application cliente. Les messages SOAP sont signés conformément aux standards OASIS WS-Security. L'élément 'Header' contient une signature portant sur un timestamp, le 'Body' et le BST (Binary Secure Token = le certificat). On peut également s'assurer de l'intégrité des données durant leur transfert et de leur actualité. (Plage de 5 minutes, par exemple)

Subject		Certificat de signature pour s'authentifier
Common Name	CN	CN= *application*[-*environnement ''***test**'' *ou '***acpt***''*].*nom de domaine* *Ou autre contenu pour identifier une application ou un rôle vis-à-vis d'une organisation*
Country	C	C= ''BE''
Organization	O	O = 'nom de l'organisation ''
Organization Unit	OU	OU1 = *acronyme de l'organisation* OU2 = **urn:be:fgov:kbo-bce:organization:cbe-number:** *numéro d'entreprise* [OU3= *environnement ''*TEST'' ou ''ACPT'']

Certificat applicatif	
Key Usage • Digital Signature, • Non-Repudiation, • Key Encipherment	Extended Key Usage • TLS web client authentication
Taille des clefs RSA	2048
Signature hash algorithm	sha256

[2] Certificat placé sur le serveur.
[3] Ne pas préciser l'environnement pour la production
[4] Certificat utilisé par une application cliente.

INFORMATIONS SUR LA SIGNATURE DU BODY, DU BINARY SECURE TOKEN ET DU TIMESTAMP

WS-Security	Normal (actuel)	Alternative (future)
WS-Security version	1.1	1.1
Include SOAP mustUnderstand	on	on
WS-Sec ID Reference Type	wsu:Id	wsu:Id
Use asymetric key	on	on
Signing algorithm	rsa	rsa-sha256/384/512
Canonicalization Algorithm	Exclusive	Exclusive
Message Digest Algorithm	sha256	sha384/512
Token Reference Mecahanism	Direct Reference	Direct Reference
X.509 Token Type	X.509	X.509
X.509 Token Profile 1.0 : BinarySecurityToken	#X509v3	#X509v3
Include Timestamp	on	on
Timestamp Expiration Period	300 sec	300 sec

ADRESSES « INBOUND » ET « OUTBOUND » DES ENVIRONNEMENTS BCSS

Pour atteindre la BCSS, deux conditions doivent être remplies :

- Le trafic IP doit pouvoir traverser les firewalls de la Smals de l'extranet (cfr demande d'ouverture du trafic IP)
- Le certificat Client doit avoir été communiqué à la BCSS.
    - to : esb@ksz-bcss.fgov.be

Environnement BCSS	Address nat (Inbound) Port **45xx (https)**	Address nat (Outbound)
Développement	85.91.184.**96** **b2b-test.ksz-bcss.fgov.be**	85.91.184.**96**
Acceptation	85.91.184.**103** **b2b-acpt.ksz-bcss.fgov.be**	85.91.184.**91** 85.91.184.**92**
Production	85.91.184.**102** **b2b.ksz-bcss.fgov.be**	85.91.184.**93** 85.91.184.**94**

Remarques :

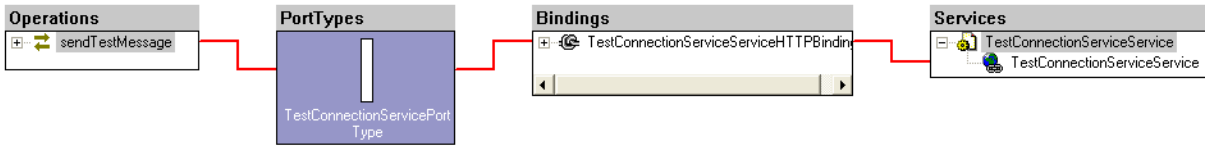
- Le port 4520 est destiné aux WS avec WSDL & XSD spécifiques.
- Le port 4522 est destiné aux WS avec WSDL & XSD spécifiques avec fichiers attachés.

  (MTOM)
- Le port 4530 accueille les requêtes SOAP avec élément de type string
    - <SendXml> [SSDN-Request]
    - XMLITE

    Se référer à la documentation sur le site web de la BCSS.

Lorsqu'un fournisseur change de certificat serveur, ils devraient nous être communiqués anticipativement à leur utilisation afin d'éviter une interruption de service pour les autres partenaires du réseau de la sécurité sociale.

La définition *TestConnectionService.wsdl* permet d'établir une session SSL avec l'environnement désiré. Si la session utilise le certificat client « accepté par la BCSS », le service répond avec le message envoyé ainsi que le « Distinguish Name » du certificat.



La requête : **https://.....:4520/TestConnectionServiceService/sendTestMessage**

```
<soapenv:Envelope
    xmlns:v1="http://kszbcss.fgov.be/intf/TestConnectionServiceService/v1"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
>
  <soapenv:Header />
  <soapenv:Body>
    <v1:sendTestMessageRequest>
      <!-- type: string -->
      <echo>hello cbss service</echo>
    </v1:sendTestMessageRequest>
  </soapenv:Body>
</soapenv:Envelope>
```

La réponse

```
<soapenv:Envelope
    xmlns:v1="http://kszbcss.fgov.be/intf/TestConnectionServiceService/v1"
    xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
>
  <soapenv:Header />
  <soapenv:Body>
    <v1:sendTestMessageResponse>
      <informationCBSS>
        <ticketCBSS>317066d4-6111-4bf4-a50b-fdf37b6ba313</ticketCBSS>
        <timestampReceive>2009-08-31T11:49:42.108Z</timestampReceive>
        <timestampReply>2009-08-31T11:49:42.299Z</timestampReply>
      </informationCBSS>
      <echo>hello cbss service</echo>
      <sslCertificate>CN=c-b2b-acpt.ksz-bcss.fgov.be,C=BE,OU=KSZ-BCSS,OU=0244640631,OU=ACPT,O=Federal
      Government</sslCertificate>
    </v1:sendTestMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Aide mémoire

Référence	La casse	signification	exemple
<name1>	Pascal case	Le nom exprimant la finalité du service	MandatPuHma
<operation>	camel case	Le <verbe-objet> identifiant le besoin demandé	checkMandatPuHma
<name2>	Pascal case	Le ***<operation>*** en Pascal case	CheckMandatPuHma

wsdl :definitions	namespace • s11 = 'http://schemas.xmlsoap.org/soap/envelope/' • wsdl = 'http://schemas.xmlsoap.org/wsdl/' • xsd = 'http://www.w3.org/2001/XMLSchema' • puo ='http://kszbcss.fgov.be/**types**/***<name1>***/v1' [Pascal case] • tns ='http://kszbcss.fgov.be/**intf**/***<name1>*****Service**/v1' targetnamespace → tns ='http://kszbcss.fgov.be/**intf**/***<name1>*****Service**/v1' name = ***<name1>*****Service** [Pascal case]
wsdl :types	xsd :schema • attributeFormDefault ='unqualified' • elementFormDefault ='unqualified' • xmlns :puo= …. • xmlns=tns= … • targetNamespace→ tns
	xsd :import • namespace → puo • schemaLocation= **<name1>V1.xsd**
	xsd :element • name = ***<operation>*****Request** [camel case] • type = puo :***<name2>*****RequestType** [Pascal case] *même démarche pour* **Response / Fault** , sauf un même '**type=**' générique pour les fault si plusieurs opérations
wsdl :message	name = ***<operation>*****RequestMsg**
	wsdl :part • element=**tns :*<operation>*Request** • name=***<operation>*****RequestParameters** *idem avec suffixes* **Response, Fault**
wsdl :portType	name =***<name1>*****PortType** [Pascal case]

	wsdl :operation name= ***<operation>*** [camel case] pour chaque opération
	wsdl:input ○ message = **tns :*<operation>*RequestMsg** ○ name= ***<operation>*Request** wsdl:output ○ message = **tns :*<operation>*ResponseMsg** ○ name= ***<operation>*Response** wsdl:fault ○ message = **tns :*<operation>*FaultMsg** ○ name= ***<operation>*Fault**
wsdl :binding	• name = ***<name1>*ServiceHTTPBinding** • type= **tns *:<name1>*PortType**
	soap:binding style="**document**" transport="http://schemas.xmlsoap.org/soap/http"/>
	wsdl :operation name= ***<operation>*** *idem pour chaque opération* soap :operation soapAction= 'http://kszbcss.fgov.be/***<name1>*Service/*<operation>*** wsdl:input name=***<operation>*Request** + soap:body use=**'literal'** wsdl:output name=***<operation>*Response** + soap:body use=**'literal'** wsdl:fault name=***<operation>*Fault** + soap:fault use=**'literal'** name=***<operation>*Fault**
wsdl :service	name = ***<name1>*Service**
	wsdl :port • binding = **tns :*<name1>*ServiceHTTPBinding** • name =**<name1>**
	soap :address location=**'https://b2b.kszbcss.fgov.be :4520/*<name1>*Service/*<name1> or <operation> si unique***