



DIRECTION GÉNÉRALE DES IMPÔTS
DIRECTION GÉNÉRALE DE LA COMPTABILITÉ PUBLIQUE

Utilisation du client WebService du SVC de niveau 2.5

060834

Version 1.00

Circuit de validation

	Nom	Organisation	Date	Visa
Rédigé par :	Jérôme Lubrez	CP SVC	24/11/2006	
Vérifié par :	Sébastien Levesque	Responsable développement SVC	24/11/2006	
Approuvé par :				

Historique des évolutions

Ver	Date	Auteur	Justificatif
0.01	24/11/2006	J. Lubrez	Création du document par dérivation du document équivalent du SVC 2 (040379)
1.00	24/11/2006	S. Levesque	Remplacement de « Collection » par « ensemble », pour éviter la confusion avec une collection au sens Java.

Sommaire

Sommaire	3
1 Objectif du document	4
1.1 Normalisation du Web Service SVC selon WS Tool 1.01	4
1.2 Support des Entités émettrices de CRL.....	4
2 Description du Webservice du SVC de niveau 2.5	5
2.1 Moteur Axis	5
2.2 SOAP, WSDL et WS Tool 1.0.1	6
3 Implémentation de l'interface SOAP	7
3.1 Méthodes accessibles	7
3.2 Exceptions	9
4 Client JAVA du webservice	10
4.1 Ligne de Commande.....	10
4.2 Données retournées.....	11
5 Appendice	12

1 Objectif du document.

Ce document présente les interfaces d'appel au Web Service du SVC de niveau 2.5

Il décrit la mise en oeuvre d'un client (écrit en java) construit sur la base du descripteur de déploiement wsdl du webservice de niveau 2.5, ainsi que son utilisation pour accéder au service web déployé sur le serveur.

Les modifications par rapport au Web Service de niveau 2 consiste en

- la normalisation du Web Service selon WS Tool 1.0.1 ;
- la présence des entités émettrices de LCR dans des requêtes et des réponses.

Ces 2 points sont expliqués ci-dessous.

1.1 Normalisation du Web Service SVC selon WS Tool 1.01

La version 2.5 du SVC présente désormais son Web Service selon l'outil WS Tool 1.01. Cet outil mis à disposition des projets par le bureau SI3 rend intéropérable, à travers un standard commun, la description de Services Web et de flux SOAP. Son impact sur la réalisation des Services et Clients Web est très léger mais pas nul : Lors du passage de la version 2.0 à la version 2.5 du SVC, l'application appelante doit modifier le typage des listes d'AC et de LCR (remplacement des java.util.Vector par des tableaux conforme aux standards).

1.2 Support des Entités émettrices de CRL

La version 2.5 du SVC supporte désormais les Autorités de Certification (AC) signant les certificats et les Listes de Révocation de Certificats (LCR) avec des clés privées distinctes. Le SVC doit vérifier la signature des LCR émises par une telle AC avec son certificat de signature de LCR (en tant qu'entité émettrice de LCR), distinct de son certificat de signature de certificat. C'est le cas l'AC GIP-CPS (Groupement d'Intérêt Public de la Carte de Professionnel de Santé) qui a été référencée en 2005. Cette AC utilise 2 certificats différents avec le même DN (Distinguished Name) pour signer les certificats et les LCR.

Bien entendu, le SVC 2.5 continue de supporter les AC signant avec une clé unique les certificats et les LCR, en utilisant le même certificat pour vérifier la signature des certificats et celle des LCR.

Aussi les méthodes du Web Service du SVC 2.5 sont impactées par cette fonction si (et uniquement si) le SVC incorpore l'AC du GIP-CPS ou une autre AC utilisant 2 clés distinctes pour signer certificats et LCR.

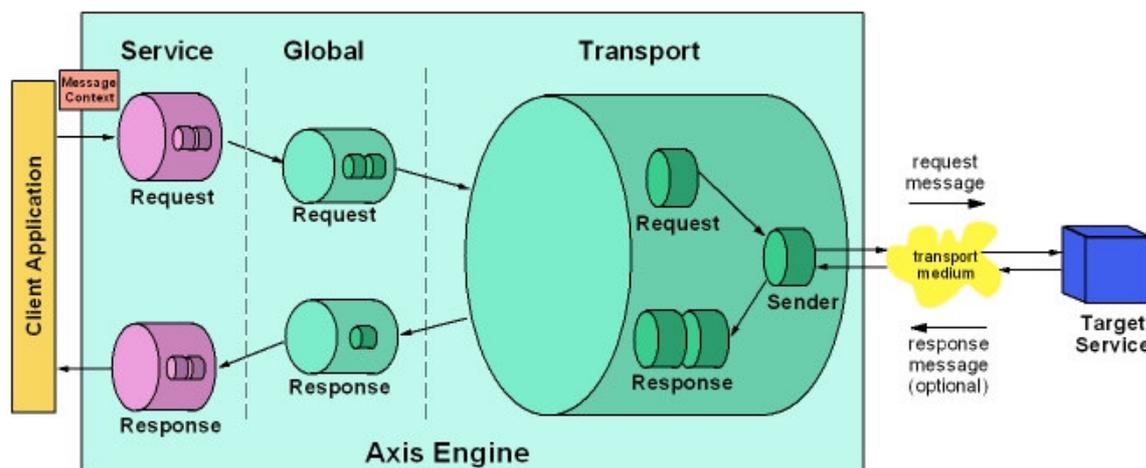
2 Description du Webservice du SVC de niveau 2.5

Le service web de publication, ou Webservice, permet l'accès des applications tierces aux données dont le SVC est le référentiel maître.

Il est accessible par un client web, via le protocole SOAP.

2.1 Moteur Axis

L'implémentation du Webservice repose sur la technologie Axis (Apache eXtensible Interaction System) qui est la dernière implémentation SOAP pour java. Il est la suite du populaire Apache SOAP. Axis est principalement un processeur SOAP. Son architecture Axis est la suivante :



L'application cliente invoque de manière transparente (vis-à-vis du protocole SOAP) le service web via un client java qui mappe les méthodes fournies par le service. L'appel est acheminé selon une chaîne de requêtes déterminées dans le moteur Axis, permettant l'utilisation de « handler » (petits cylindres internes sur le schéma) sur cette chaîne (par exemple des traitements de log ou de traçage des requêtes)

Le service web est appelé pour retourner la réponse sur le flux de sortie pouvant également contenir une chaîne de handlers.

Un Service web n'est autre qu'une transaction accessible par l'échange de documents XML entre deux URL. Le principal avantage est l'instauration d'un dialogue direct entre applications.

2.2 SOAP, WSDL et WS Tool 1.0.1

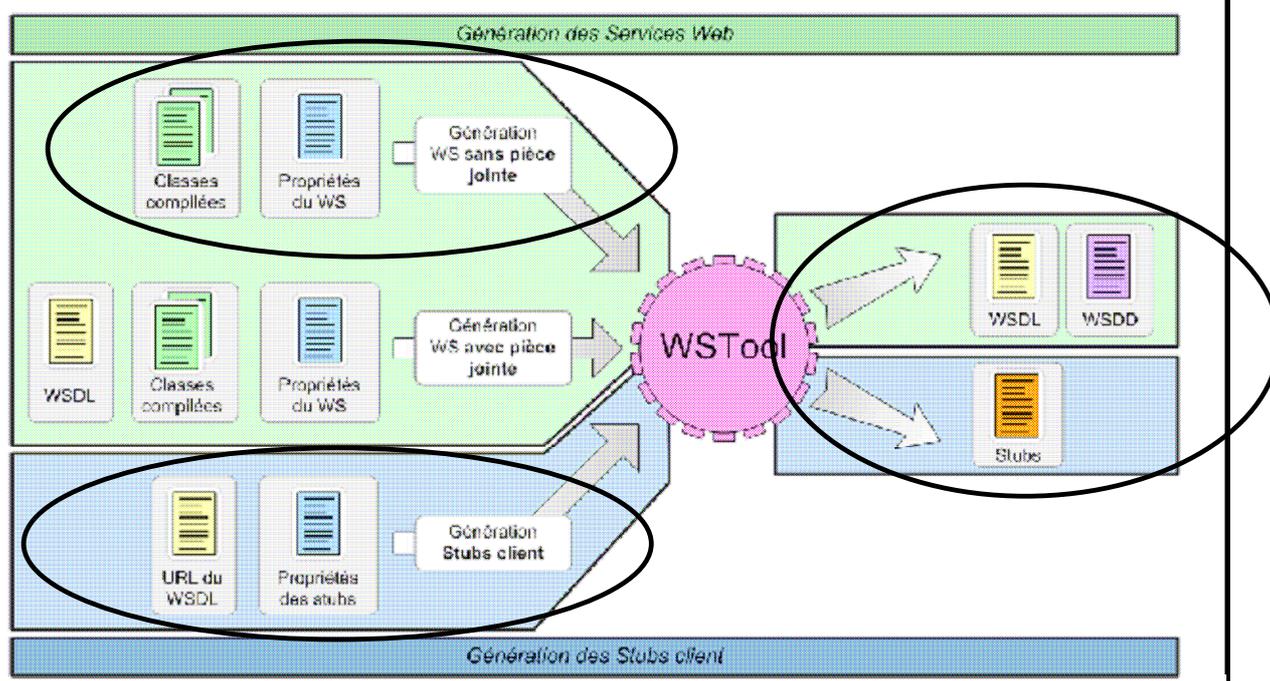
Les Services Web reposent sur le protocole SOAP qui définit la structure du message XML utilisé par les services Web pour dialoguer entre eux, et automatise ce dialogue.

WSDL est un format de description des composants (c'est-à-dire des services eux-mêmes) invocables par le biais de messages XML au format SOAP. Il s'agit d'un langage de contrat, qui permet de reconnaître les schémas XML utilisés et d'établir une connexion entre consommateur et fournisseur de service.

L'objectif est de permettre une invocation dynamique entre services Web, et de faciliter ainsi les échanges entre ces derniers.

WSTool a été utilisé en remplacement des outils de développement natifs fournis dans AXIS 1.1 pour

- la Génération du document WSDL et du fichier de déploiement WSDD à partir de l'interface et de la classe d'implémentation du Service Web SVC (sans pièce jointe, pas d'attachement).
- La Génération des stubs clients d'appel d'un Service Web à partir du document WSDL.



3 Implémentation de l'interface SOAP

3.1 Méthodes accessibles

Récupération d'un objet «WS_AC».

```
public svc.webService.beans.WS_AC getCa(java.lang.String dn) throws
java.rmi.RemoteException;
```

→ La méthode `getCa` permet via le DN d'une AC de récupérer un objet `WS_AC`, contenant :

- son identifiant dans le SVC,
- son DN,
- son certificat,
- le numéro de série de son certificat,
- la date de début de validité,
- la date de fin de validité

La réponse n'est pas signée par le SVC.

La classe `WS_AC` ainsi que ses accesseurs est décrite dans la suite de ce document.

Notons que dans le cas particulier d'une AC signant ses certificats et ses LCR avec une clé différente (cas du GIP-CPS), seul l'objet `WS_AC` du signataire des certificats est retourné.

Récupération de l'ensemble des AC déclarées dans le SVC.

```
public svc.webService.beans.WS_AC[] getAllCA() throws
java.rmi.RemoteException;
```

→ La méthode `getAllCA` renvoie un tableau d'objets `WS_AC`.

La réponse n'est pas signée par le SVC.

Notons que pour le cas particulier d'une AC signant ses certificats et ses LCR avec une clé différente (cas du GIP-CPS), les deux objets `WS_AC` du signataire des certificats et du signataire des LCR sont inclus dans le tableau retourné.

Obtention de l'ensemble des AC d'un domaine de confiance.

```
public svc.webService.beans.WS_AC[]
getAllCAFromTrustDomain(java.lang.String trustDomOid) throws
java.rmi.RemoteException;
```

→ La méthode `getAllCAFromTrustDomain` permet d'obtenir l'ensemble des AC d'un domaine de confiance à partir de l'OID de celui-ci `trustDomOid`. La réponse est un tableau d'objet `WS_CA`.

La réponse n'est pas signée par le SVC.

Notons que pour le cas particulier d'une AC signant ses certificats et ses LCR avec une clé différente (cas du GIP-CPS), les deux objets WS_AC du signataire des certificats et du signataire des LCR sont inclus dans le tableau retourné.

Obtention de l'ensemble des AC terminales de chemins de certification.

```
public svc.webService.beans.CAListResponse getTermCAList(java.lang.String trustDomOid, java.lang.String pvOid) throws java.rmi.RemoteException;
```

→ La méthode `getTermCAList` permet d'obtenir l'ensemble des AC terminales à partir d'un OID de domaine de confiance `trustDomOid` et d'un OID de PV `pvOid`. La réponse est un objet `CAListResponse` constitué par :

- la liste des WS_AC correspondant aux AC terminales de chemins de certification,
- la date de production des informations (date courante),
- la date de dernière mise à jour des chemins de validation (valeur la plus récente du champ `creationdate` de la table `validpath`)
- l'OID de PV réellement utilisé pour le calcul.

Si l'objet demandé n'existe pas, la méthode renvoie simplement la valeur `null`.

La réponse n'est pas signée par le SVC.

Notons que pour le cas particulier d'une AC signant ses certificats et ses LCR avec une clé différente (cas du GIP-CPS), seul l'objet WS_AC de l'AC terminale, signataire des certificats, est retourné.

Récupération d'un ensemble de listes de révocation.

```
public svc.webService.beans.WS_LCR[] getLastCRL(java.lang.String dn) throws java.rmi.RemoteException;
```

→ La méthode `getLastCRL` permet via le DN d'une AC de récupérer un tableau d'objets WS_LCR décrivant les listes de révocation avec :

- l'identifiant SVC de l'AC émettrice,
- le DN de l'AC émettrice,
- le contenu de la LCR,
- la date `thisupdate`,
- le nom du fichier de LCR archivé sur le serveur.

La réponse n'est pas signée par le SVC.

La classe WS_LCR ainsi que ses accesseurs est décrite dans la suite de ce document.

Récupération d'un ensemble de listes de révocation entre deux dates données.

```
public svc.webService.beans.WS_LCR[] getListCRLFileFromCA(java.lang.String dn, java.lang.String begin_date, java.lang.String end_date) throws java.rmi.RemoteException ;
```

→ La méthode `getListCRLFileFromCA` permet la récupération du sous-ensemble des listes de certificats révoqués publiées entre deux dates données. C'est un tableau d'objets WS_LCR.

La réponse n'est pas signée par le SVC.

Méthode de test.

```
public svc.webService.beans.WS_LCR getNewCRL() throws  
java.rmi.RemoteException;
```

→ La méthode `getNewCRL` renvoie une nouvelle instance de `WS_LCR` (objet vide).

La réponse n'est pas signée par le SVC.

3.2 Exceptions

En cas d'erreur technique, les méthodes appelées renvoient systématiquement une exception `RemoteException`.

4 Client JAVA du webservice

4.1 Ligne de Commande.

Après le déploiement d'un service web dans jboss, il est nécessaire d'avoir un client SOAP permettant de dialoguer avec celui-ci.

Le client `svc.client.SVCServiceClient` construit avec WSTool est déjà livré dans les clients de tests dans la ressource `svcwsclient.jar`.

Il est toutefois possible à un développeur ou une application de régénérer son propre client avec WSTool à partir du flux `wSDL` décrivant le service déployé (voir appendice).

La syntaxe pour exécuter le `main` de cette classe est la suivante :

```
java svc.client.SVCServiceClient [url] [nom de méthode] [paramètres de la méthode]
```

[url]	url de déploiement du webservice par exemple : <i>http://172.20.5.44</i>
[nom de méthode]	nom de la méthode du webservice que l'on veut exécuter : <i>getAllCAFromTrustdomain</i>
[paramètres de la méthode]	le nombre de paramètres dépend de la méthode appelée. Si on utilise une String avec des blancs ne pas oublier de mettre des guillemets autour du paramètre. <i>1.2.250.1.131.1.5.4.7.1.2</i> ou <i>"cn=CATest, o=CATest, c=fr"</i>

Voici un tableau récapitulatif des six méthodes que l'on peut appeler avec ce client et leurs paramètres d'appel :

Nom de méthode	Paramètres de la méthode	Données retournées
GetCa	Dn	WS_AC
GetAllCa	-	tableau de WS_AC
GetAllCAFromTrustdomain	TrustDomOid	tableau de WS_AC
GetTermCAList	trustDomOid pvOid	CAListResponse
GetListCrlFileFromCA	dn begin_date end_date	tableau de WS_LCR
GetLastCrl	dn	tableau de WS_LCR

4.2 Données retournées

Les méthodes du webservice renvoient des objets WS_AC, WS_LCR et CAListResponse ainsi que des tableaux de ces objets. Tous ces objets appartiennent au package svc.webService.beans, et possèdent des accesseurs pour manipuler leurs données.

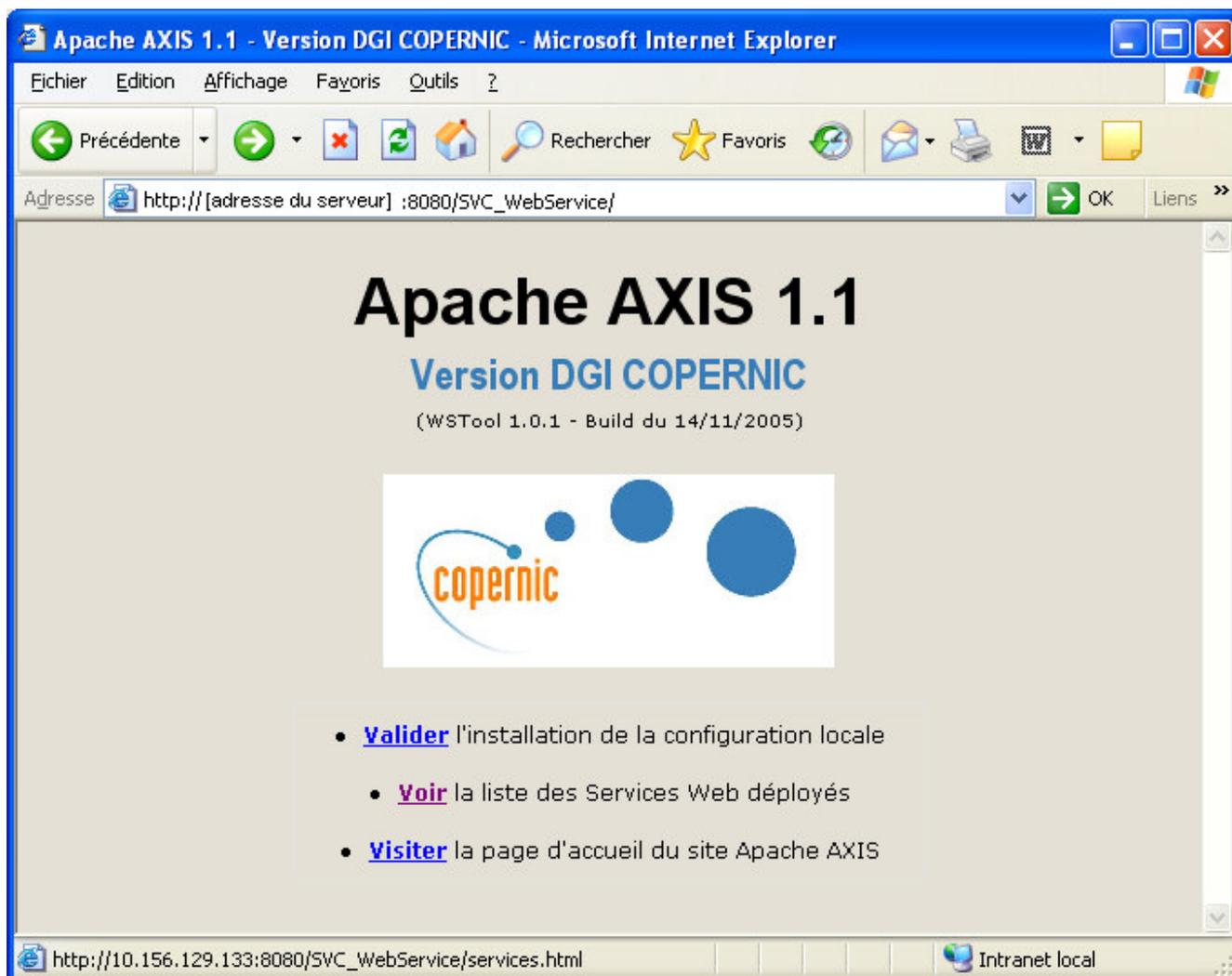
<i>svc.webService.beans.WS_AC</i>		
Accesseur	Type	Signification
getSvcId()	String	Identifiant de l'AC dans le SVC
getSubjectDn()	String	DN de l'AC
getCertContent()	bytes[]	Certificat de l'ac.
getSerialNumber()	String	Numéro de série de son certificat
getNotBefore()	java.util.Calendar	Date de début de validité
getNotAfter()	java.util.Calendar	Date de fin de validité

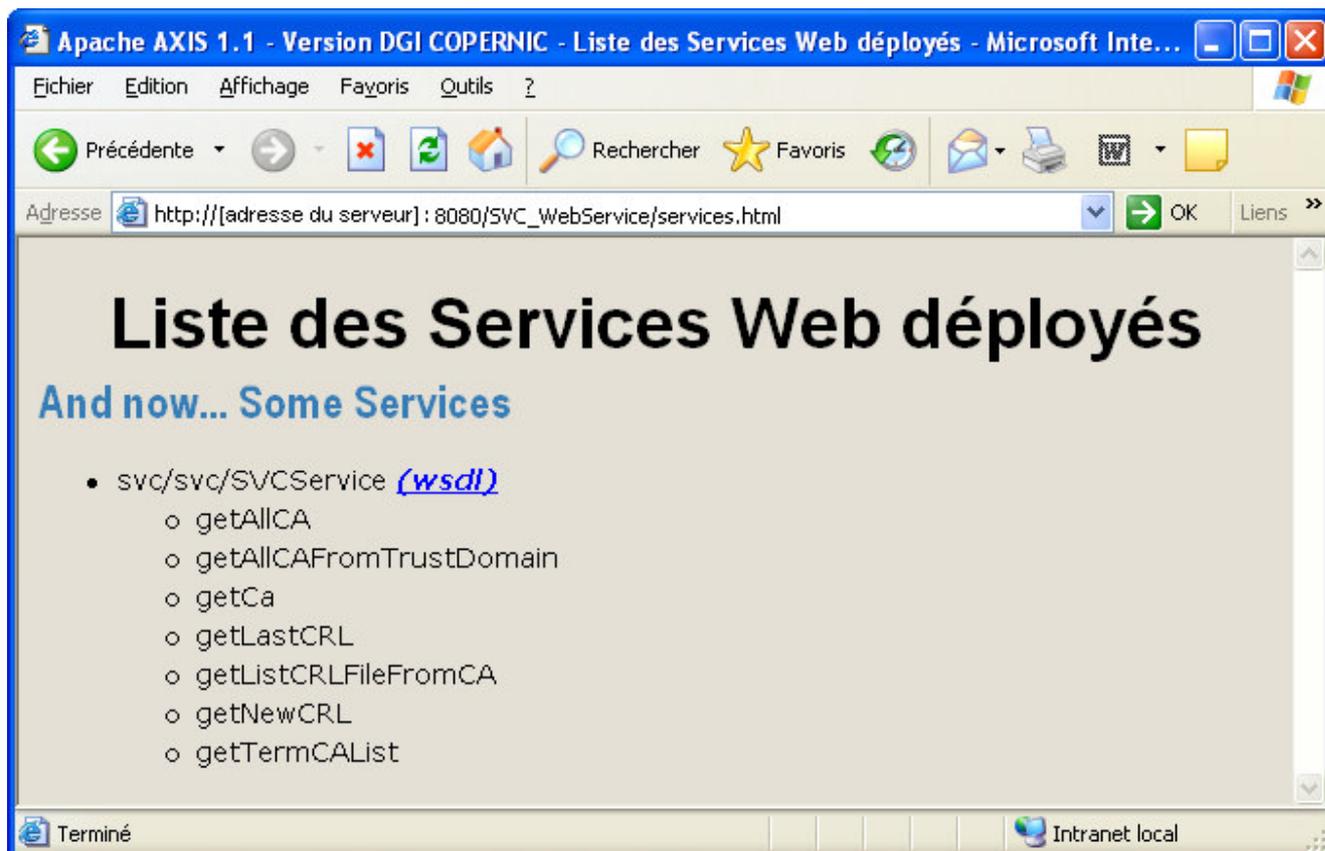
<i>svc.webService.beans.WS_LCR</i>		
Accesseur	Type	Signification
getSvcAcId()	String	Identifiant de l'AC dans le SVC
getAcDn()	String	Dn de l'AC
getLcrContent()	bytes[]	Contenu de la LCR
getFileName()	String	Nom du fichier archivé sur le serveur
getThisupdate()	java.util.Calendar	Date Thisupdate de la crl

<i>svc.webService.beans.CAListResponse</i>		
Accesseur	Type	Signification
getProductionDate()	java.util.Calendar	Date courante
getLastUpdateDate()	java.util.Calendar	Date de dernière mise à jour des chemins de validation
getPvOid()	String	OID de PV réellement utilisé pour le calcul
getACs()	tableau WS_AC[]	Ensemble des AC terminales du chemin de certification

5 Appendice

Voici le déploiement du service SVC, visualisé en se connectant au répondeur Web Service du SVC :
(cf. page suivante).





Le flux du descripteur wsdl du SVCService est le suivant :

```
<?xml version="1.0" encoding="UTF-8" ?>
= <definitions name="SVCService"
  targetNamespace="http://svc.webservice.dgi.finances.gouv.fr/svc/svc/SVCService"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns2="http://svc.webservice.dgi.finances.gouv.fr/types"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://svc.webservice.dgi.finances.gouv.fr/svc/svc/SVCService"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
= <types>
= <schema targetNamespace="http://svc.webservice.dgi.finances.gouv.fr/objetvaleur"
  xmlns="http://www.w3.org/2001/XMLSchema" xmlns:soap11-
  enc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://svc.webservice.dgi.finances.gouv.fr/objetvaleur"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <import namespace="http://svc.webservice.dgi.finances.gouv.fr/types" />
= <complexType name="WS_AC">
= <sequence>
  <element name="certContent" nillable="true" type="base64Binary" />
  <element name="notAfter" nillable="true" type="dateTime" />
  <element name="notBefore" nillable="true" type="dateTime" />
  <element name="serialNumber" nillable="true" type="string" />
  <element name="subjectDn" nillable="true" type="string" />
  <element name="svcId" nillable="true" type="string" />
</sequence>
```

```

</complexType>
<complexType name="WS_LCR">
  <sequence>
    <element name="acDn" nillable="true" type="string" />
    <element name="fileName" nillable="true" type="string" />
    <element name="lcrContent" nillable="true" type="base64Binary" />
    <element name="svcAcId" type="int" />
    <element name="thisupdate" nillable="true" type="dateTime" />
  </sequence>
</complexType>
<complexType name="CAListResponse">
  <sequence>
    <element maxOccurs="unbounded" minOccurs="0" name="ACs" nillable="true"
      type="tns:WS_AC" />
    <element name="lastUpdateDate" nillable="true" type="dateTime" />
    <element name="productionDate" nillable="true" type="dateTime" />
    <element name="pvOid" nillable="true" type="string" />
  </sequence>
</complexType>
</schema>
<schema targetNamespace="http://svc.webservice.dgi.finances.gouv.fr/types"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:ns2="http://svc.webservice.dgi.finances.gouv.fr/objetvaleur" xmlns:soap11-
  enc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:tns="http://svc.webservice.dgi.finances.gouv.fr/types"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <import namespace="http://svc.webservice.dgi.finances.gouv.fr/objetvaleur" />
  <complexType name="getAllICA">
    <sequence />
  </complexType>
  <complexType name="getAllICAResponse">
    <sequence>
      <element maxOccurs="unbounded" minOccurs="0" name="result"
        nillable="true" type="ns2:WS_AC" />
    </sequence>
  </complexType>
  <complexType name="getAllICAFromTrustDomain">
    <sequence>
      <element name="String_1" nillable="true" type="string" />
    </sequence>
  </complexType>
  <complexType name="getAllICAFromTrustDomainResponse">
    <sequence>
      <element maxOccurs="unbounded" minOccurs="0" name="result"
        nillable="true" type="ns2:WS_AC" />
    </sequence>
  </complexType>
  <complexType name="getCa">
    <sequence>

```

```

        <element name="String_1" nillable="true" type="string" />
    </sequence>
</complexType>
<complexType name="getCaResponse">
    <sequence>
        <element name="result" nillable="true" type="ns2:WS_AC" />
    </sequence>
</complexType>
<complexType name="getLastCRL">
    <sequence>
        <element name="String_1" nillable="true" type="string" />
    </sequence>
</complexType>
<complexType name="getLastCRLResponse">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="result"
            nillable="true" type="ns2:WS_LCR" />
    </sequence>
</complexType>
<complexType name="getListCRLFileFromCA">
    <sequence>
        <element name="String_1" nillable="true" type="string" />
        <element name="String_2" nillable="true" type="string" />
        <element name="String_3" nillable="true" type="string" />
    </sequence>
</complexType>
<complexType name="getListCRLFileFromCAResponse">
    <sequence>
        <element maxOccurs="unbounded" minOccurs="0" name="result"
            nillable="true" type="ns2:WS_LCR" />
    </sequence>
</complexType>
<complexType name="getNewCRL">
    <sequence />
</complexType>
<complexType name="getNewCRLResponse">
    <sequence>
        <element name="result" nillable="true" type="ns2:WS_LCR" />
    </sequence>
</complexType>
<complexType name="getTermCAList">
    <sequence>
        <element name="String_1" nillable="true" type="string" />
        <element name="String_2" nillable="true" type="string" />
    </sequence>
</complexType>
<complexType name="getTermCAListResponse">
    <sequence>

```

```

        <element name="result" nillable="true" type="ns2:CAListResponse" />
    </sequence>
</complexType>
<element name="getAllICA" type="tns:getAllICA" />
<element name="getAllICAResponse" type="tns:getAllICAResponse" />
<element name="getAllICAFromTrustDomain" type="tns:getAllICAFromTrustDomain" />
<element name="getAllICAFromTrustDomainResponse"
    type="tns:getAllICAFromTrustDomainResponse" />
<element name="getCa" type="tns:getCa" />
<element name="getCaResponse" type="tns:getCaResponse" />
<element name="getLastCRL" type="tns:getLastCRL" />
<element name="getLastCRLResponse" type="tns:getLastCRLResponse" />
<element name="getListCRLFileFromCA" type="tns:getListCRLFileFromCA" />
<element name="getListCRLFileFromCAResponse"
    type="tns:getListCRLFileFromCAResponse" />
<element name="getNewCRL" type="tns:getNewCRL" />
<element name="getNewCRLResponse" type="tns:getNewCRLResponse" />
<element name="getTermCAList" type="tns:getTermCAList" />
<element name="getTermCAListResponse" type="tns:getTermCAListResponse" />
</schema>
</types>
= <message name="SVCSservice_getAllICA">
    <part element="ns2:getAllICA" name="parameters" />
</message>
= <message name="SVCSservice_getAllICAResponse">
    <part element="ns2:getAllICAResponse" name="result" />
</message>
= <message name="SVCSservice_getAllICAFromTrustDomain">
    <part element="ns2:getAllICAFromTrustDomain" name="parameters" />
</message>
= <message name="SVCSservice_getAllICAFromTrustDomainResponse">
    <part element="ns2:getAllICAFromTrustDomainResponse" name="result" />
</message>
= <message name="SVCSservice_getCa">
    <part element="ns2:getCa" name="parameters" />
</message>
= <message name="SVCSservice_getCaResponse">
    <part element="ns2:getCaResponse" name="result" />
</message>
= <message name="SVCSservice_getLastCRL">
    <part element="ns2:getLastCRL" name="parameters" />
</message>
= <message name="SVCSservice_getLastCRLResponse">
    <part element="ns2:getLastCRLResponse" name="result" />
</message>
= <message name="SVCSservice_getListCRLFileFromCA">
    <part element="ns2:getListCRLFileFromCA" name="parameters" />
</message>

```

```

=> <message name="SVCSservice_getListCRLFileFromCAResponse">
    <part element="ns2:getListCRLFileFromCAResponse" name="result" />
</message>
=> <message name="SVCSservice_getNewCRL">
    <part element="ns2:getNewCRL" name="parameters" />
</message>
=> <message name="SVCSservice_getNewCRLResponse">
    <part element="ns2:getNewCRLResponse" name="result" />
</message>
=> <message name="SVCSservice_getTermCAList">
    <part element="ns2:getTermCAList" name="parameters" />
</message>
=> <message name="SVCSservice_getTermCAListResponse">
    <part element="ns2:getTermCAListResponse" name="result" />
</message>
=> <portType name="SVCSservice">
    => <operation name="getAllICA">
        <input message="tns:SVCSservice_getAllICA" />
        <output message="tns:SVCSservice_getAllICAResponse" />
    </operation>
    => <operation name="getAllICAFromTrustDomain">
        <input message="tns:SVCSservice_getAllICAFromTrustDomain" />
        <output message="tns:SVCSservice_getAllICAFromTrustDomainResponse" />
    </operation>
    => <operation name="getCa">
        <input message="tns:SVCSservice_getCa" />
        <output message="tns:SVCSservice_getCaResponse" />
    </operation>
    => <operation name="getLastCRL">
        <input message="tns:SVCSservice_getLastCRL" />
        <output message="tns:SVCSservice_getLastCRLResponse" />
    </operation>
    => <operation name="getListCRLFileFromCA">
        <input message="tns:SVCSservice_getListCRLFileFromCA" />
        <output message="tns:SVCSservice_getListCRLFileFromCAResponse" />
    </operation>
    => <operation name="getNewCRL">
        <input message="tns:SVCSservice_getNewCRL" />
        <output message="tns:SVCSservice_getNewCRLResponse" />
    </operation>
    => <operation name="getTermCAList">
        <input message="tns:SVCSservice_getTermCAList" />
        <output message="tns:SVCSservice_getTermCAListResponse" />
    </operation>
</portType>
=> <binding name="SVCSserviceBinding" type="tns:SVCSservice">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
    
```

```

=> <operation name="getAllCA">
    <soap:operation soapAction="" />
=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
=> <operation name="getAllCAFromTrustDomain">
    <soap:operation soapAction="" />
=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
=> <operation name="getCa">
    <soap:operation soapAction="" />
=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
=> <operation name="getLastCRL">
    <soap:operation soapAction="" />
=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
=> <operation name="getListCRLFileFromCA">
    <soap:operation soapAction="" />
=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
=> <operation name="getNewCRL">
    <soap:operation soapAction="" />

```

```

=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
=> <operation name="getTermCAList">
    <soap:operation soapAction="" />
=> <input>
    <soap:body use="literal" />
</input>
=> <output>
    <soap:body use="literal" />
</output>
</operation>
</binding>
=> <service name="SVCSservice">
=> <port binding="tns:SVCSserviceBinding" name="SVCSservicePort">
    <soap:address
        location="http://localhost:8080/SVC_WebService/services/svc/svc/SVCSservice"
    />
</port>
</service>
</definitions>

```