DIRECTION GÉNÉRALE DES IMPÔTS
DIRECTION GÉNÉRALE DE LA COMPTABILITÉ PUBLIQUE

Autorité de Validation

Spécifications détaillées du SVC de niveau 2

040339

Version 1.01

THALES
INFORMATION SYSTEMS

Circuit de validation

	Nom	Organisation	Date	Visa
Rédigé par :	Nathalie Jolly	Rédacteur	15/09/2004	
Vérifié par :	Phong Tran Antoinette Franzese	Responsable T-IS	28/06/2004	
Approuvé par :	Véronique Sage Patrick Murzeau	Responsable Domaine Directeur de projet DGI	16/09/2004	

Historique des évolutions

Ver	Date	Auteur	Justificatif
0.10	9/03/2004	N. Jolly	Création du document
0.20	17/03/2004	N. Jolly	Compléments suite à la réunion du 15/03/2004. Ajout de l'algorithme d'exécution d'une PV. Description des rôles d'administration. Recensement des contrôles à effectuer par VERCER et VALREV. Complément à l'algorithme de précalcul des chemins de certification pour le respect intégral de la RFC 3280.
0.25	22/03/2004	N. Jolly	Compléments suite à la réunion du 22/03/2004.
0.30	23/03/2004	N. Jolly	Ajout du modèle de données. Relecture SI3. Diffusion interne SI3.
0.40	26/03/2004	N. Jolly	Ajout de l'architecture technique et de l'architecture logicielle. Ajout des entrées/sorties des composants. Ajout des contrôles complémentaires à effectuer par CRL2DB. Mise en conformité par rapport aux spécifications fonctionnelles. Suppression de l'authentification par certificats. Diffusion interne SI3.
0.50	13/04/2004	N. Jolly	Ajout des messages d'erreur fonctionnels. Modification gestion onlyContainsCACerts et onlyContainsUserCerts suite à discussion avec Yannick Quenec'hdu.
0.60	3/05/2004	N. Jolly	SIGREP : appel à l'API C d'OpenSSL au lieu de la carte cryptographique hardware Trustway. Supervision : codage des transformations SNMP / Nagios dans les messages SNMP. Administration : format XML du fichier de PV signée.

Ver	Date	Auteur	Justificatif
0.70	3/06/2004	N. Jolly	Modification de l'architecture Oracle Ajout de la gestion des versions de format XML
0.80	23/06/2004	N. Jolly	Modification de la spécification du Webservice
0.90	27/08/2004	N. Jolly	Modification des schémas d'architecture
0.95	14/09/2004	N. Jolly	Relecture et modifications afin de rendre ce document indépendant des spécifications de Niveau 1
1.00	16/09/2004	V. Sage	Validation SI3
1.01	01/03/2005	N. Jolly	Ajout des traps SNMP liés à la sauvegarde-restauration. Modifications mineures sur la syntaxe du trap SNMP.

Références documentaires

N°	Titre	Référence	Auteur	Version
[1]	Spécifications fonctionnelles générales du SVC de niveau 2	040136	Yannick Quenec'hdu	1.00
[2]	Guide interface d'administration	040346	Youcef Semsoum	1.00
[3]	Document d'exploitation	040347	Phong Tran	1.00
[4]	Document d'installation	040348	Simon Lebettre	1.00
[5]	Documentation organique du module de récupération des LCR	040351	Phong Tran Sébastien Janaud	1.1

Sommaire

1 Introduction

L'Autorité de Validation de niveau 2 a pour objectif la validation des certificats qui lui sont présentés. L'implémentation technique associée, le **Service de Validation de Certificats (SVC),** est un système permettant :

- le paramétrage de domaines de confiance et de politiques de validation (PV) en fonction des processus définis par l'Autorité de Validation ;
- la validation d'un certificat en fonction d'un domaine de confiance et d'une politique de validation, cette validation pouvant comporter, de façon paramétrable dans la PV, une vérification de contenu, une vérification de non-révocation et une vérification de la chaîne de certification ;
- la mise à disposition de la liste des Autorités de Certification et d'autres informations utiles aux applications via un WebService ;
- un processus de mise à jour régulier du référentiel des certificats révoqués du SVC en fonction des LCR mises à disposition par les AC référencées.

La description fonctionnelle détaillée de ces éléments se situe dans le document [1].

1.1 Présentation du document

Le présent document constitue le document de conception technique détaillée pour l'implémentation technique de l'Autorité de Validation du projet Copernic. Il est le document de référence pour les chefs de projet et les développeurs.

Il contient la documentation technique complète du SVC de niveau 2, au sens où la spécification du SVC de niveau 1 n'est pas un prérequis à la lecture.

1.2 Front-office et back-office

Les blocs fonctionnels décrits dans la spécification fonctionnelle du SVC se décomposent le plus souvent dans deux systèmes bien distincts :

- l'un en charge de la réponse interactive à une requête d'un client (désigné dans la suite de ce document comme le « front-office » ou comme « Répondeur SVC »),
- l'autre en charge de la mise à jour et de l'administration du système (désigné dans la suite de ce document comme le « back-office »).

Le présent document présente l'architecture logicielle du système et les choix techniques d'implémentation selon ces deux axes, ainsi que le modèle de données du SVC.

1.3 Niveau 1 et niveau 2

Le SVC de niveau 1 fait référence à un premier niveau de fonctionnement (validation de la non-révocation, accessible en OCSP et via un client Java/XML). Il correspond à un premier palier d'implémentation mis en oeuvre dans le cadre du CFS-Pro.

Le SVC de niveau 2 fait référence à un niveau de fonctionnement nettement enrichi. Il apporte à l'Autorité de Validation un ensemble de fonctions supplémentaires (vérification de certificats et

validation du chemin de certification), ces fonctions indépendantes étant appelées ou non par l'Autorité de Validation en fonction de la politique de validation (PV) applicable pour la validation en cours.

2 Glossaire des acronymes

La définition des concepts associés est disponible dans le document [1].

Acronyme	Signification
AC - CA	Autorité de Certification.
AV	Autorité de Validation.
CAC	bloc fonctionnel dont le périmètre est la vérification des Chaînes d'Autorités de Certification.
CER	bloc fonctionnel dont le périmètre est la vérification du CERtificat.
CRL2DB	injection des CRL dans la Base de Données.
LCR - CRL	Liste des Certificats Révoqués.
PRECAL	PRECALcul des chemins de certification.
PV	Politique de Validation.
REV	bloc fonctionnel dont le périmètre est la vérification de non-REVocation
SIGREP	SIGnature de la REPonse du répondeur SVC.
SVC	Service de Validation de Certificats.
TFERR	service de Traitement Fonctionnel des ERReurs.
VALREV	VALidation vérification de non-révocation (REV). Il permet la vérification de la révocation d'un certificat par consultation d'une base de données de référence contenant la liste des certificats révoqués.
VERCAC	Composant front-office du service de vérification des chemins de certification (CAL). Il permet la vérification de la validité d'un chemin par consultation d'une base de données de référence contenant l'ensemble des chemins précalculés et le statut des AC.
VERCER	Composant front-office du service de vérification de certificats (CER). Il comporte les vérifications sur le contenu d'un certificat, indépendamment de son contexte. Note : ce composant n'a pas de pendant back-office.

3 Front-office

3.1 Interfaces d'appel du SVC

Le répondeur SVC de niveau 2 est accessible par ses clients depuis un point d'entrée unique (une seule URL d'accès). Sur ce point d'entrée, il reçoit :

- soit des requêtes respectant le standard OCSP et n'implémentant pas l'extension `requestorName`,
- soit des requêtes respectant le standard OCSP et implémentant l'extension `requestorName`,
- soit des requêtes respectant un format plus riche (mais non standard) permettant la transmission du certificat entier au SVC et donc une vérification approfondie de ce certificat.

3.2 Format d'échange

3.2.1 Cas d'une requête OCSP

Le service rendu est l'appel du composant VALREV qui offre le service de validation de non-révocation en implémentant le protocole OCSP (Online Certification Statut Protocol).

La requête et la réponse sont conformes au protocole OCSP tel que défini dans la RFC 2560. L'implémentation du protocole OCSP s'appuie sur le CODEC du composant logiciel SEMOA (logiciel libre sous certaines conditions, sous un contrat de licence spécifique, fourni par la société allemande Fraunhofer-Gesellschaft), qui fournit en particulier les classes OCSPRequest, OCSPResponse et BasicOCSPResponse.

La requête OCSP peut être signée. Dans ce cas, le répondeur SVC n'effectue pas de vérification de signature.

La réponse OCSP contient en particulier :

- le certificat de l'instance de répondeur SVC,
- le certificat de son AC terminale dans le champ OCSP `certs`,
- le `nonce` s'il a été envoyé,
- la date de réponse `producedAt`.

3.2.2 Cas d'une requête OCSP étendue

Le service rendu est l'appel des composants spécifiés dans la politique de validation (PV) par défaut du domaine de confiance de l'appelant. Cette PV doit concerner au moins le composant VALREV et ne peut pas concerner le composant VERCER. De façon paramétrable, elle peut concerner ou non les composants VERCAC et SIGREP.

L'OID du domaine de confiance de l'appelant est passé dans l'attribut optionnel `requestorName` de la requête OCSP (champ de type `registeredID`).

La requête OCSP étendue peut être signée. Dans ce cas, le répondeur SVC n'effectue pas de vérification de signature.

La réponse OCSP utilise le champ de réponse `CertStatus ([good|revoked|unknown])` pour transmettre la réponse du SVC. La valeur `good` signifie que l'ensemble des contrôles de la PV par défaut a été effectué avec succès. La valeur `revoked` signifie que l'un des contrôles de la PV par défaut a échoué. La valeur `unknown` n'est jamais rendue, elle est transformée en `good` ou en `revoked` par le traitement fonctionnel des erreurs (voir 3.8 - Traitement fonctionnel des erreurs).

La réponse OCSP contient en particulier :

- le certificat de l'instance de répondeur SVC,
- le certificat de son AC terminale dans le champ OCSP `certs`,
- le `nonce` s'il a été envoyé,
- la date de réponse `producedAt`.

3.2.3 Cas d'une requête enrichie

Le service rendu est l'appel des composants présents dans la politique de validation transmise dans la requête.

3.2.3.1 Description des formats XML

Le flux XML envoyé au service SVC est le suivant :

```
<?xml version="1.0"?>
<certvalidation>
     <svc version="2.00"/>
     <trustdom>OID du domaine de confiance de l'appelant</trustdom>
     <validationpolicy>OID (identifiant.version) de la politique de
validation à utiliser</validationpolicy>
     <certificate>certificat au format PEM</certificate>
     <nonce>identifiant optionnel à transmettre à VALREV</nonce>
</certvalidation>
```

La requête HTTP contient le `content-type` suivant : « application/svc-request ».

Le serveur retourne la réponse sous la forme du flux XML suivant :

```
<?xml version="1.0"?>
<certvalidation>
<svcresponse>
      <svc version="2.00"/>
      <certid>numéro de série du certificat</certid>
      <validationpolicy>OID (identifiant.version) de la politique de
validation utilisée</validationpolicy>
      <responsestatus>validated|rejected</responsestatus>
      <error type="S|F|T" code="xxx">message d'erreur</error>
      <producedat>date système de la réponse</producedat>
      <nonce>valeur du nonce (ou champ laissé vide en absence de
nonce)</nonce>
      <revocation>
            <version>version du protocole OCSP</version>
            <responderid>identifiant du répondeur OCSP</responderid>
            <certstatus>statut du certificat</certstatus>
            <thisupdate>date thisupdate présente dans la CRL</thisupdate>
            <nextupdate>date nextupdate présente dans la CRL</nextupdate>
            <revocationtime>date de révocation</revocationtime>
            <downloadtime>date de récupération de la CRL</downloadtime>
            <nocheck></nocheck>
      </revocation>
      <certpath>
            <cert order="1">certificat de l'AC racine (PEM)</cert>
            ...
            <cert order="n">certificat de l'AC terminale (PEM)</cert>

      </certpath>
</svcresponse>
<signature>signature (optionnelle) de l'ensemble de la réponse par
l'instance de SVC ayant effectué la validation</signature>
<certs>
      <certificate>certificat de l'instance de SVC</certificate>
      <cacertificate>certificat de son AC terminale</cacertificate>
</certs>
</certvalidation>
```

La requête HTTP de réponse contient le `content-type` suivant : « application/svc-response ».

Le statut de la réponse (bloc <responsestatus></responsestatus>) est « `validated` » si le certificat est validé. Il est « `rejected` » dans le cas d'une erreur fonctionnelle ou d'une erreur technique.

Le bloc XML <error></error> est toujours présent. En l'absence d'erreur, il contient les données (type = S, code = 0, message vide). En cas d'erreur technique, le type d'erreur est T. En cas d'erreur fonctionnelle, le type d'erreur est F. Dans les deux cas, le message d'erreur est libellé en français.

Le bloc XML <revocation></revocation> n'est présent que si un appel à VALREV a été effectué. S'il est présent, tous les champs du niveau inférieur sont présents.

Le bloc XML <certpath></certpath> n'est présent que si un appel à VERCAC a été effectué. S'il est présent, tous les champs du niveau inférieur sont présents. Les certificats renvoyés sont systématiquement ordonnés dans l'ordre de la chaîne de certification, de l'AC racine à l'AC terminale.

Le bloc XML de signature n'est présent que si un appel à SIGREP a été effectué. Il consiste en l'ajout de la signature de l'empreinte SHA-1 de la réponse <svcresponse></svcresponse>.

Techniquement, l'implémentation est effectuée par l'envoi des flux XML ci-dessus dans les requêtes HTTP, sans passer par une enveloppe SOAP, par souci de performance.

3.2.3.2 Gestion des versions de documents

Afin d'anticiper les éventuelles évolutions de format XML, l'ensemble des formats d'entrée acceptables par le service SVC est configuré dans un fichier de configuration. Si la version contenue dans le message XML reçu par le SVC appartient à cette liste de versions, le service est rendu. Dans le cas contraire, un message d'erreur est renvoyé à l'appelant.

3.2.4 Client Java

3.2.4.1 Client Java de niveau 2

Le SVC de niveau 2 fournit une classe `ClientSVC` qui implémente 4 méthodes :

```
public static String sendXmlRequest(X509Certificate certificat_x509, String
OID_domaine_de_confiance, String OID_politique_de_validation, String nonce) throws
IOException


public static String sendXmlRequest(String urlServer, X509Certificate
certificat_x509, String OID_domaine_de_confiance, String
OID_politique_de_validation, String nonce) throws IOException


public static SvcSimpleResponse sendAndParseXmlRequest(X509Certificate
certificat_x509, String OID_domaine_de_confiance, String
OID_politique_de_validation, String nonce) throws SAXException, IOException


public static SvcSimpleResponse sendSVCRequest(String urlServer, X509Certificate
certificat_x509, String OID_domaine_de_confiance, String
OID_politique_de_validation, String nonce) throws SAXException, IOException
```

L'URL du répondeur SVC est soit présente dans un fichier de configuration du client Java, de façon à être facilement modifiable, soit passée directement en paramètre par le client final.

La classe `String` de réponse correspond à l'envoi direct du flux XML décrit ci-dessus, qui doit être analysé par le client.

La classe `SvcSimpleResponse` permet au client de manipuler directement les différents attributs de la réponse enrichie par l'appel à des accesseurs sans avoir à analyser le flux XML.

Note : Le SVC ne reçoit que trois types de requête : les requêtes OCSP, les requêtes OCSP étendues et les requêtes enrichies. Le client Java transforme les appels aux méthodes de sa classe en requêtes enrichies XML et les réponses XML qu'il reçoit du SVC en `String` ou en `SimpleSVCResponse`.

3.2.4.2 Compatibilité ascendante avec le niveau 1

Pour mémoire, le client Java de niveau 1 propose une classe `ClientValrevcer` implémentée par les applications désirant faire effectuer par le SVC la vérification de certains champs du certificat. Cette classe implémente deux méthodes selon le type de la réponse fournie à l'appelant :

```
String sendRequest(String urlServerValrev, X509Certificate x509)


SimpleResponse sendRequest2(String urlServerValrev, X509Certificate x509)
```

Cette interface, qui ne permet pas la transmission d'une requête enrichie (au sens du SVC de niveau 2) n'est pas maintenue dans le SVC de niveau 2.

3.3 Service « Pilote »

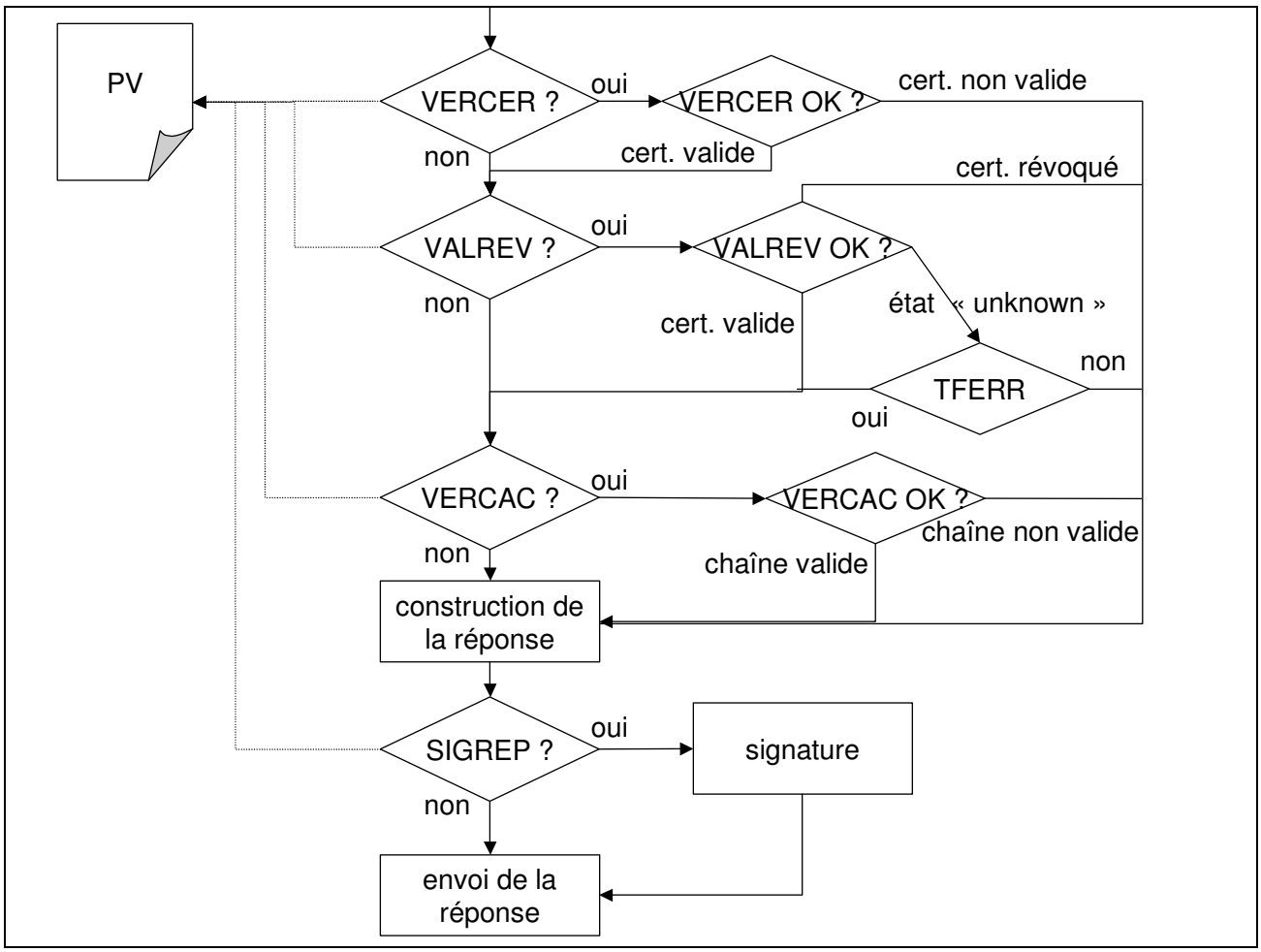
Le Pilote effectue les actions suivantes :

- si l'appel est un appel OCSP, exécution de la PV « VALREV – SIGREP » et renvoi de la réponse au format OCSP,
- si l'appel est un appel OCSP étendu, exécution de la PV par défaut du domaine de confiance présenté dans le champ `requestorName` et renvoi de la réponse au format OCSP,
- si l'appel est une requête enrichie :
  - lecture en base de la PV associée,
  - vérification que le domaine de confiance de l'application appelante est un des domaines de confiance reconnus par la PV,
  - vérification que l'AC qui a émis le certificat appartient au domaine de confiance de l'application appelante,
  - appel des services de validation présents dans la PV,
  - renvoi de la réponse au format XML.

Si la PV fait appel au composant VALREV, il est inutile que VALREV signe sa réponse. En retour de VALREV, le pilote applique le traitement fonctionnel des erreurs TFERR.

Algorithme utilisé par le pilote pour l'exécution d'une PV :

Les composants VERCER, VALREV et VERCAC sont appelés dans cet ordre s'ils sont présents dans la PV exécutée. Dès qu'un composant répond négativement, les composants suivants ne sont pas appelés et le pilote passe à la fonction de signature.



3.4 Description technique d'une politique de validation

3.4.1 Identification d'une PV

Chaque PV est identifiée par un identifiant unique (OID), qui se décompose en un préfixe, un identifiant de PV (champ `idvp` en base de données) et une version.

La syntaxe d'un OID de PV est la suivante :

ISO.member-body.France.type-org.minefi.autorité de validation.politique de validation.`id.version`

Les premiers champs (ISO.member-body.France.type-org.minefi.autorité de validation.politique de validation) sont fixes (1.2.250.1.131.1.5.4.6.1)

Les identifiants `id` et `version` sont des entiers, à la discrétion du SVC.

3.4.2 Règles de gestion

Une même PV peut être associée à aucun, un ou plusieurs domaines de confiance.

Parmi les versions de PV de même identifiant `id`, une seule version de la PV est active à un instant donné.

Parmi les PV actives, l'une d'elles est la PV par défaut, appliquée dans le cas des requêtes OCSP étendues. Celle-ci ne peut pas contenir l'appel à VERCER et doit contenir l'appel à VALREV.

En cas d'appel par une requête enrichie, le client transmet au SVC un OID que le pilote décompose en un `idvp` et un numéro de version (`idvp` = ISO.member-body.France.type-org.minefi.autorité de validation.politique de validation.`id`).

Le pilote applique systématiquement la PV active correspondant à cet `idvp` et transmet en retour au client l'OID de la PV réellement appliquée (même `idvp` mais numéro de version éventuellement différent).

Techniquement, les politiques de validation sont stockées dans la base de données. On trouve donc en base :

- une table des PV (`idvp`, `version`, état active ou non, et autres champs de description),
- une table des associations entre `idvp` et domaines de confiance, qui permet également de stocker pour chaque domaine de confiance quelle est l'`idvp` de la PV par défaut.

Plusieurs PV de même `idvp` peuvent être présentes en base avec des versions différentes, et la clef primaire d'une PV est le couple (`idvp`, `version`). Pour un `idvp` donné, une seule version de PV est active à un moment donné.

3.4.3 Paramétrage d'une politique de validation

Une PV est définie par les paramètres suivants :

Nom	Description
name	Nom de la PV
idvp	Identifiant de la PV
version	Version de la PV
isValid	Champ d'état définissant si la PV est à valider (« A »), validée (« V ») ou refusée (« R »)
isActive	Champ d'état définissant si la PV est active ou inactive (PV active => isValid = « V »)
isDefault	Champ définissant si la PV est la PV par défaut (PV par défaut => PV active)
vercer	Champ définissant si le contrôle VERCER est à effectuer
extendedkeyUsage	Si VERCER, liste des valeurs autorisées pour le champ d'extension extendedKeyUsage
cA	Champ booléen définissant si le certificat présenté est un certificat d'AC ou un certificat d'utilisateur
valrev	Champ définissant si le contrôle VALREV est à effectuer
unknown	Si VALREV, booléen définissant si VALREV répond `valide` ou `révoqué` en cas de réponse `unknown` du service OCSP
vercac	Champ définissant si le contrôle VERCAC est à effectuer
pathLengthConstraint	Si VERCAC, profondeur maximale de l'arbre acceptée par l'AC racine. Ce paramètre est utilisé lors de la construction des chemins, par la méthode `PKIXBuilderParameter.setMaxPathLength()`.
sigrep	Champ définissant si le composant de signature SIGREP est à appeler

Ces paramètres sont implémentés dans des tables du modèle de données (Pour la structure des tables et le type des champs, cf. paragraphe 5 - Modèle de données).

3.5 Description technique d'un domaine de confiance

3.5.1 Identification d'un domaine de confiance

Chaque domaine de confiance est identifié par un identifiant unique (OID), qui se décompose en un préfixe et un identifiant de domaine de confiance.

La syntaxe d'un OID de domaine de confiance est la suivante :

ISO.member-body.France.type-org.minefi.autorité de validation.domaine de confiance.`id`

Les premiers champs (ISO.member-body.France.type-org.minefi.autorité de validation.domaine de confiance) sont fixes (1.2.250.1.131.1.5.4.7.1)

L'identifiant `id` est un entier, à la discrétion du SVC.

3.5.2 Règles de gestion

Un domaine de confiance peut être associé à aucune, une ou plusieurs AC.

Une AC peut appartenir à aucun, un ou plusieurs domaines de confiance.

Un domaine de confiance peut être associé à aucune, une ou plusieurs PV (plusieurs OID de PV et non uniquement plusieurs versions d'une même PV).

3.6 Service « VERCER »

3.6.1 Présentation du service

Le service VERCER vérifie la validité du certificat en appliquant les contrôles définis dans le tableau suivant. Dès qu'un contrôle échoue, VERCER répond négativement au pilote sans exécuter les contrôles suivants. Si tous les contrôles sont effectués avec succès, VERCER répond positivement.

Champ	Contrôle
`version`	Champ obligatoire, valeur 2 (ce qui correspond à un certificat V3).
`serialNumber`	Champ obligatoire.
`signatureAlgorithm`	Champ obligatoire.
`issuer`	Champ obligatoire.
`subjectPublicKeyInfo`	Champ obligatoire.
`subject`	Champ obligatoire.
`validity`	Champ obligatoire. La date de validation doit être supérieure ou égale à `notBefore` et inférieure ou égale à `notAfter`.
`signatureValue`	Champ obligatoire, pour lequel il faut vérifier la signature avec la clef publique de l'autorité qui l'a émis.
`extendedKeyUsage`	La PV peut contenir un ensemble d'OID d'ExtendedKeyUsage. Dans ce cas, le champ `extendedKeyUsage` est obligatoire et chacun des OID qu'il contient doit être égal à une des valeurs de cette liste. Dans le cas contraire, tous les OID éventuellement présents dans ce champ sont acceptés.
`privateKeyUsagePeriod`	Si le `keyUsage` du certificat est « `digitalSignature` » et si ce champ

Champ	Contrôle
	est présent, alors il doit contenir les deux dates (`notBefore` et `notAfter`) et la date de validation doit être supérieure ou égale à `notBefore` et inférieure ou égale à `notAfter`.
`basicConstraints : attribut cA`	Champ obligatoire dont la valeur doit être égale à la valeur présente dans les paramètres de la PV.

Si d'autres extensions sont présentes dans le certificat, elles ne sont pas vérifiées, qu'elles soient critiques ou non.

3.6.2 Implémentation

En ce qui concerne le champ ExtendedKeyUsage, Java offre dans la classe de manipulation des certificats la méthode `getExtendedKeyUsage()` qui extrait du certificat la liste des OID des extendedKeyUsage.

La PV contient les extendedKeyUsage sous forme d'OID et VERCER effectue un contrôle d'égalité entre OID, sans descendre au niveau des chaînes de bits correspondantes.

3.6.3 Entrées / Sorties

Le service VERCER prend en entrée le certificat à vérifier sous la forme d'un objet `Certificate` et retourne :

- soit un booléen, un code et un message d'erreur fonctionnel,
- soit un booléen, un code et un message d'erreur technique.

3.7 Service « VALREV »

3.7.1 Présentation du service

Le composant VALREV offre le service de validation de non-révocation en implémentant le protocole OCSP (Online Certification Statut Protocol).

La réponse à cet appel est systématiquement non signée (la signature de l'ensemble de la réponse, et non uniquement de VALREV, étant effectuée par le composant SIGREP, module séparé, le cas échéant).

Le répondeur OCSP implémente l'ensemble des champs obligatoires de la RFC 2560 et les champs optionnels suivants :

Champ	Implémentation associée
`id-pkix-ocsp-nonce`	Si VALREV reçoit une valeur `nonce` dans le champ optionnel `id-pkix-ocsp-nonce` (`requestExtention`), il renvoie cette même valeur `nonce` dans le champ optionnel `id-pkix-ocsp-nonce` (`responseExtension`).
`id-pkix-ocsp-nocheck`	Lors de la génération de la réponse, VALREV positionne ce champ.

Les autres champs présents dans la RFC 2560 ne sont pas implémentés.

Note : le `nonce` est géré par le pilote dans le cas d'une requête enrichie.

En cas d'erreur technique de VALREV (`malformedRequest , internalError, tryLater`), le certificat est considéré comme « révoqué ».

3.7.2 Implémentation

Le composant VALREV établit la révocation ou non du certificat en consultant la table `revoline` des certificats révoqués.

Il est composé de deux modules :

- un module « OCSP » accessible selon le protocole OCSP et effectuant la vérification de non-révocation (l'implémentation de la vérification de non-révocation s'effectue par contrôle de la présence ou non du certificat dans la table `revoline`),
- un module de conversion entre les formats XML et OCSP pour l'appel du module OCSP à partir d'une requête enrichie.

En cas d'appel OCSP ou OCSP étendu du SVC, le fil d'exécution ne traverse que le module OCSP.

En cas d'appel « requête enrichie » du SVC, une conversion est effectuée par le module de conversion en entrée et en sortie afin de réaliser un appel OCSP au module effectuant la vérification de révocation.

3.7.3 Entrées / Sorties

Le service VALREV a deux modes d'appel :

- soit il prend en entrée un objet OCSPRequest tel que défini dans le codec OCSP et retourne un objet OCSPSimpleResponse tel que défini dans le codec OCSP,
- soit il prend en entrée le certificat à vérifier sous la forme d'un objet `Certificate` et retourne un objet xmlModuleResponse comportant la réponse XML, un booléen, un code et un message d'erreur fonctionnel ou un booléen, un code et un message d'erreur technique.

Dans ce dernier cas, la réponse XML respecte le format suivant :

```
<revocation>
      <version>version du protocole OCSP</version>
      <responderid>identifiant du répondeur OCSP</responderid>
      <certstatus>statut du certificat validé</certstatus>
      <thisupdate>date thisupdate présente dans la CRL</thisupdate>
      <nextupdate>date nextupdate présente dans la CRL</nextupdate>
      <revocationtime>date de révocation</revocationtime>
      <downloadtime>date de récupération de la CRL</downloadtime>
      <nocheck></nocheck>
</revocation>
```

3.8 Traitement fonctionnel des erreurs

VALREV peut fournir trois types de réponses :

- certificat valide,
- certificat révoqué,
- certificat pour lequel le statut est inconnu (« unknown ») dans le cas d'une requête OCSP.

Dans le cas d'une requête XML enrichie, si le statut est inconnu, le traitement fonctionnel des erreurs tranche en fonction de la valeur du champ VALREV-unknown de la PV et le certificat est considéré comme valide ou comme révoqué.

Le traitement fonctionnel des erreurs ne révoque pour autant pas le certificat dans la base puisque, en fonction d'une autre PV, le traitement fonctionnel des erreurs pourrait considérer le certificat comme valide.

3.9 Service « VERCAC »

3.9.1 Présentation du service

Ce service consiste à rechercher un chemin valide pour le certificat en cours de validation d'après les racines de confiance présentes dans la base, et ceci pour une politique de validation et un domaine de confiance donnés.

Dans le cas de l'appel OCSP étendu, comme le certificat n'est pas transmis par le client appelant le SVC, on se base directement sur le hash du DN de l'AC, sur le hash de la clé publique de cette AC et sur l'identifiant de l'algorithme de hash, passés dans la requête OCSP, pour retrouver le DN de l'AC correspondante (à partir de ces données, on retrouve l'AC via la table `hashca` du modèle de données) et chercher un chemin valide à partir de cette AC.

3.9.2 Implémentation

VERCAC effectue l'algorithme suivant :

1. Rechercher dans la table `validpath` l'occurence de l'AC émettrice du certificat qui correspond aux valeurs de l'identifiant de PV (`idvp`), de la version (`version`) et de l'identifiant du domaine de confiance (`idtrustdom`),
2. En déduire l'identifiant de l'AC racine correspondante (`idroot`),
3. Rechercher dans la table `validpath` l'ensemble (trié par `pathorder` croissant) des AC de la chaîne correspondant aux valeurs de l'identifiant de PV(`idvp`), de la version (`version`) ,de l'identifiant du domaine de confiance (`idtrustdom`) et de l'identifiant de l'AC racine (`idroot`),
4. Vérifier que la chaîne entre l'AC émettrice et la racine est valide (liste des `pathorder` sans doublon et sans trou),
5. Dans le cas de la vérification d'un certificat utilisateur, vérifier que l'AC présente dans le certificat est bien l'AC terminale de la chaîne (l'AC présente de certificat doit être la dernière de la liste des AC),
6. Vérifier que l'ensemble des AC de la chaîne entre l'AC émettrice et la racine est valide par vérification de la non-révocation (champ `larstatus`) et de la non-péremption (champ `enddate`) de la table `ca`.

Au cours de cette exécution, dès qu'un contrôle échoue, VERCAC répond négativement au pilote sans exécuter les contrôles suivants.

Si l'AC ayant délivré le certificat n'est pas terminale, le chemin n'est pas valide.

Si l'une des AC du chemin est révoquée, le chemin n'est pas valide.

Si l'une des AC du chemin est périmée, le chemin n'est pas valide.

Sinon, le chemin de certification est valide.

Si tous les contrôles sont effectués avec succès, VERCAC répond positivement.

3.9.3 Entrées / Sorties

Le service VERCAC reçoit comme paramètres d'entrée :

- l'identifiant de la PV applicable (`pvid`),
- la version de la PV applicable (`version`),
- le domaine de confiance de l'appelant,
- le DN de l'AC émettrice du certificat.

Il retourne :

- le flux XML contenant la liste des certificats des AC au format PEM, ordonnés dans l'ordre de la chaîne de certification,
- un booléen, un code et un message d'erreur fonctionnel ou un booléen, un code et un message d'erreur technique.

La réponse XML respecte le format suivant :

```
<certpath>
      <cert order="1">certificat de l'AC racine (PEM)</cert>
      ...
      <cert order="n">certificat de l'AC terminale (PEM)</cert>

</certpath>
```

3.10 Service « SIGREP »

Ce service signe la réponse en la cryptant avec la clef privée de l'instance de serveur SVC. Cette clef privée est paramétrable, afin que chaque machine physique installée « répondeur SVC » puisse avoir une clef privée propre.

L'implémentation de ce service est réalisée sous la forme d'une interface unifiée masquant deux implémentations :

- l'appel aux fonctions de signature Java (BouncyCastle, Sun, ...),
- l'appel aux fonctions C de signature de la librairie `openssl` (mise en oeuvre d'une interface JNI entre la partie Java de SIGREP et le langage C).

3.11 Implémentation technique

L'implémentation retenue pour ces services est un ensemble d'EJB qui s'exécutent dans un serveur d'applications. Le serveur retenu est un serveur JBoss sous Linux (master DGI, sur la base d'un Redhat 7.3).

L'architecture applicative correspondante est présentée au chapitre 7 - Architecture logicielle du front-office.

3.12 Mise à disposition des données

La liste des AC est mise à disposition des applications par l'intermédiaire d'un annuaire accessible par le protocole LDAP.

La liste des AC et des CRL est également accessible dans la base de données, par l'intermédiaire d'une interface SOAP (webservice).

Dans les réponses fournies par le webservice, une AC est appelée WS_AC et est décrite par :

- 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.

Une LCR est appelée WS_LCR et est décrite par :

- 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 sur le disque.

Le SVC de niveau 1 offre par l'interface SOAP les quatre méthodes suivantes :

- `getCA()` : obtention d'une WS_AC à partir d'un DN d'AC,
- `getCACertChain()` : obtention de la chaîne de certification sous la forme d'une collection de WS_AC à partir d'un DN d'AC,
- `getLastCRL()` : obtention de la dernière WS_LCR à partir d'un DN d'AC,
- `getListCRLFileFromCA()` : obtention de l'ensemble des WS_LCR entre deux dates données à partir d'un DN d'AC.

L'interface SOAP évolue dans le SVC de niveau 2 afin d'apporter les méthodes complémentaires suivantes :

- `getAllCA()` : obtention de l'ensemble des WS_AC déclarées dans le SVC,
- `getCAFromTrustDomain()` : obtention de l'ensemble des WS_AC d'un domaine de confiance à partir d'un OID de domaine de confiance,
- `getTermCAList()` : à partir d'un OID de domaine de confiance et d'un OID de PV, obtention d'un objet `CAListResponse` contenant :
  - 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.

Le Webservice est accompagné d'une classe cliente comportant les accesseurs permettant la manipulation des objets WS_AC et WS_LCR.

Note : la méthode `getCACertChain()` présente dans le SVC de niveau 1 est supprimée car le DN d'AC ne suffit plus à obtenir la chaîne de confiance, il faut également fournir un OID de domaine de confiance et un OID de PV.

4 Back-office

4.1 Composant CRL2DB (mise à jour des LCR)

Le composant CRL2DB permet la récupération des LCR mises à disposition sur Internet par les AC référencées. Il se décompose en trois modules :

- CRLProcess est l'ordonnanceur de récupération. Il prend en compte les paramétrages des rythmes de récupération dans la base de données et déclenche les demandes de récupération. A la réception d'une LCR, il effectue un ensemble de vérifications et met à jour la base de données.
- CRLFinder est le processus en charge de la récupération effective des LCR. Il reçoit les requêtes en provenance du CRLFinder, accède via HTTP aux sites internet sur lesquels les AC mettent les LCR à disposition et retransmet les LCR téléchargées au CRLProcess après avoir effctué un premier ensemble de vérifications.
- CRLVerify est le processus en charge de la surveillance que les AC ont des LCR à jour. Si la LCR est obsolète, ce processus passe l'AC concernée de l'état valide (nommé Terminé) à l'état Erreur.

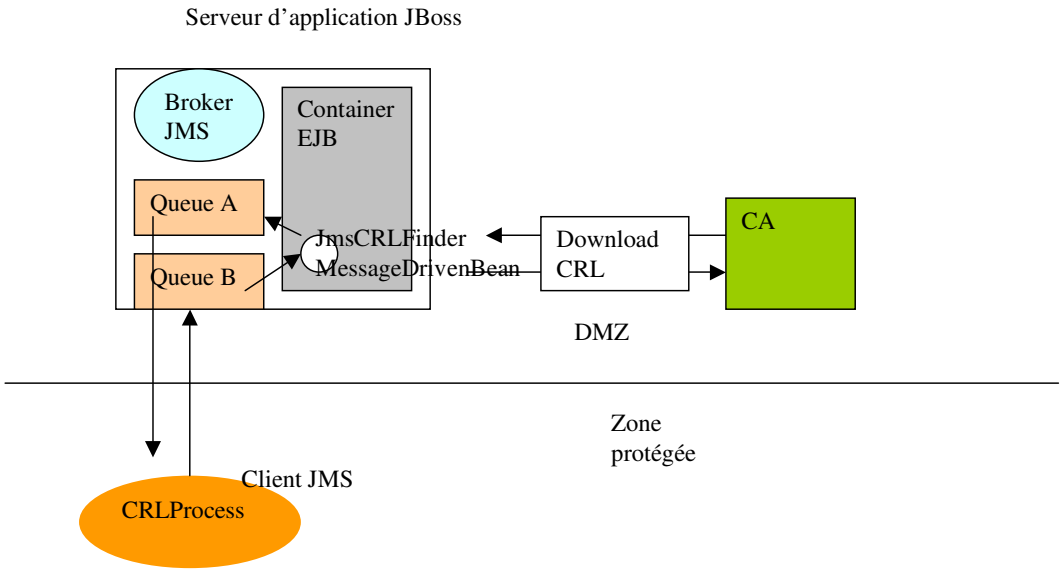
4.1.1 Architecture technique

Les composants CRLFinder, CRLProcess et CRLVerify sont implémentés sous la forme d'EJB, dans une architecture J2EE (socle Linux DGI, serveur d'application JBoss).

CRLProcess et CRLFinder sont déployés sur deux serveurs physiquement distincts et communiquent via le mécanisme JMS. Ceci augmente la sécurité de l'ensemble grâce à la rupture de protocole entre HTTP et JMS. Deux queues JMS sont initialisées. Au démarrage de JBoss, CRLFinder et CRLProcess s'abonnent en réception chacun sur une des deux queues JMS.

CRLFinder est localisé dans une zone disposant d'un accès à Internet.

CRLProcess dispose d'un accès à la base de données référentiel du SVC.



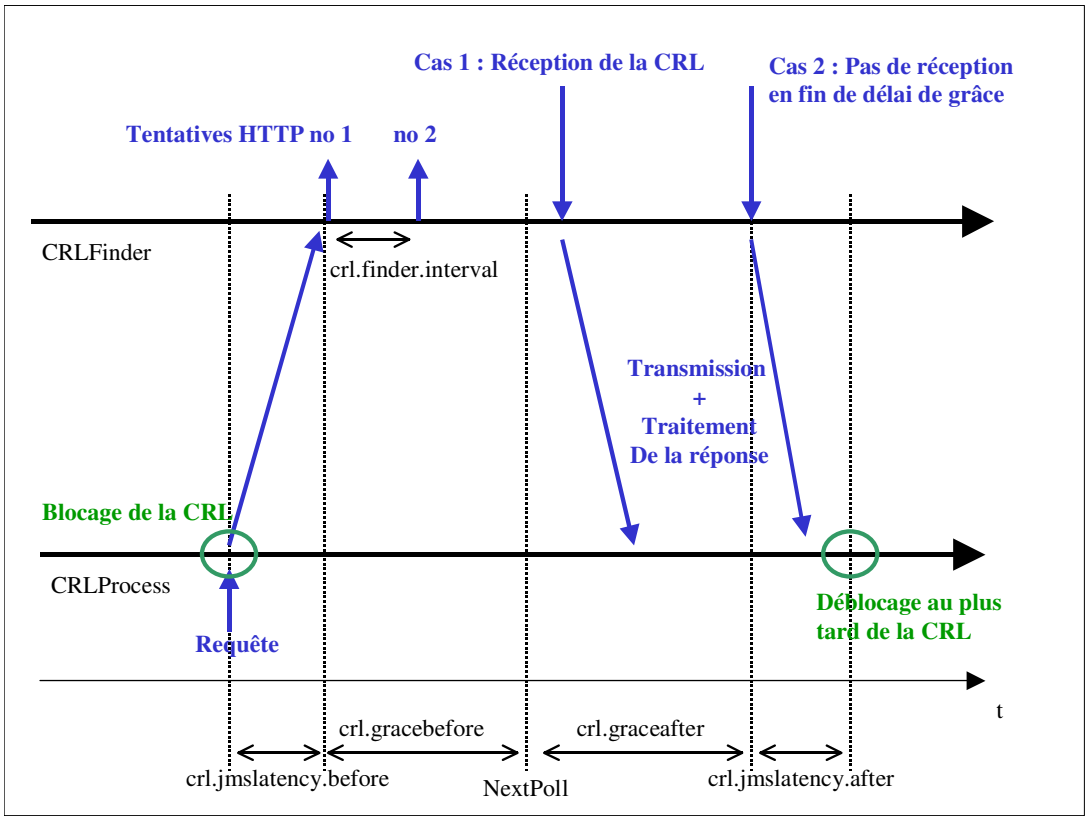
4.1.2 Description du mécanisme de mise à jour des LCR

CRLProcess vérifie périodiquement dans la base de données la date de synchronisation applicable à chaque LCR. Lorsqu'une synchronisation est requise, il émet une demande de récupération de la LCR au module CRLFinder via le système JMS.

CRLFinder est un Javabean piloté par message (MessageDrivenBean). A l'arrivée d'une requête, il effectue une connexion HTTP ou LDAP sur l'URI transmise par le CRLProcess afin de déclencher la récupération de la LCR. Il effectue un nombre paramétrable de tentatives si la connexion ne s'établit pas immédiatement. Soit le téléchargement se termine avec succès, auquel cas il dépose la LCR dans la queue JMS à destination de CRLProcess, soit CRLFinder atteint une date butoir à laquelle il transmet au CRLProcess une réponse négative, également via la queue JMS.

CRLProcess récupère la réponse (LCR ou message d'erreur) sur la queue sur laquelle il est abonné en réception et traite la réponse. S'il reçoit une LCR, il effectue un ensemble de contrôles et de traitements, puis l'intègre le cas échéant au sein de la base de données.

Le diagramme de séquence montrant la synchronisation entre CRLFinder et CRLProcess est représenté sur le schéma ci-dessous.



La description technique détaillée des paramètres représentés est disponible dans la documentation organique du SVC [5].

4.1.3 Contrôles effectués par CRLFinder

Le tableau suivant présente les vérifications effectuées par ce module dans le niveau 1 et les vérifications supplémentaires effectuées par ce module dans le niveau 2.

Niveau 1	Niveau 2
• Présence du champ `thisupdate`, sa valeur doit être supérieure ou égale à la valeur en base.	• Présence du champ `signatureAlgorithm`. • Vérification de la signature de la CRL (1). • Heure de récupération comprise entre les valeurs `thisUpdate` et `nextUpdate` de la CRL.

(1) Afin de limiter les transferts de données sur le réseau, CRLFinder vérifie l'intégrité de la CRL avant transmission à CRLProcess (le transfert comporte pour autant toujours la CRL signée et CRLProcess effectue une seconde vérification d'intégrité).

Important : ceci implique de faire évoluer la requête JMS afin de transmettre au CRLFinder le certificat de l'AC interrogée.

4.1.4 Contrôles effectués par CRLProcess

Le tableau suivant présente les vérifications effectuées par ce module dans le niveau 1 et les vérifications supplémentaires effectuées par ce module dans le niveau 2.

Niveau 1	Niveau 2
• Présence du champ `signatureAlgorithm`. • Vérification de la signature de la CRL. • Présence du champ `thisupdate`. • Présence du champ `nextupdate` si Pkix. • Heure de récupération entre les valeurs de `thisupdate` et de `nextupdate`. • Présence du champ `revokedCertificates`. • Si CRL de version 2, vérification que le champ `CRLNumber` a une valeur strictement supérieure à la valeur de celui de la dernière CRL récupérée.	• Présence du champ `issuer`, sa valeur doit être en rapport avec le certificat de l'AC qui a émis la CRL. • Validité (non-révocation et non-peremption) de l'ensemble du chemin de certification. (3) • Si CRL de version 2, vérification que l'émetteur de la CRL est l'émetteur du certificat ou qu'il correspond à la valeur indiquée par le champ additionnel `cRLDistributionPoints`. • Vérifier si la CRL ne contient pas une AC révoquée. Si existence du numéro de série du certificat de l'AC dans la CRL, émission d'une alerte par mail à l'administrateur, révocation de l'AC (champ `larstatus` de la table `ca`). (2) • Si CRL indirecte, présence des champs `indirectCRL` et `certificateIssuer`.

Pour tous les contrôles sauf (2), si l'un des contrôles à effectuer échoue, CRLProcess interrompt le traitement et ne met pas à jour la base de données avec le contenu de la CRL.

Le contrôle (2) a un traitement particulier :

- il est effectué juste avant insertion dans la base, sur la liste des nouvelles entrées de CRL (différence calculée entre les entrées de la table `revoline` et la CRL à intégrer),
- si la CRL contient une AC révoquée, la mise à jour de la base de données s'effectue seulement dans certains cas :

  - si le champ `onlyContainsUserCerts` est FALSE et le champ `onlyContainsCACerts` est TRUE, CRLProcess met à jour la base de données.
  - si les champs `onlyContainsUserCerts` et `onlyContainsCACerts` ont tous les deux la valeur FALSE, CRLProcess ne met pas à jour la base de données.
  - si le champ `onlyContainsUserCerts` est TRUE et le champ `onlyContainsCACerts` est FALSE, CRLProcess ne met pas à jour la base de données. (flag `onlyContainsUserCerts` faux par rapport au contenu de la CRL).
  - si les champs `onlyContainsUserCerts` et `onlyContainsCACerts` ont tous les deux la valeur TRUE, CRLProcess ne met pas à jour la base de données.
- s'il ne met pas à jour la base de données, CRLProcess conserve dans un répertoire la CRL afin de permettre une injection manuelle ultérieure par l'administrateur.

En complément de ces vérifications,

- CRLProcess tient à jour les statuts des AC dans la table `ca`. Après chaque mise à jour de la base après récupération des LAR, il passe le statut des AC révoquées à « `revoqué` » (état « R »).
- CRLProcess met à jour après chaque insertion de CRL en base un champ `downloadtime` dans la table `crl`, permettant de fournir aux clients appelants la date d'obtention de l'information de révocation.
- La prise en compte de la valeur `pollinterval` de la table `policy` est modifiée : l'unité est la minute et non l'heure.

(3) La validité du chemin impose de s'appuyer sur une PV et un domaine de confiance. Il est donc nécessaire de définir un domaine de confiance « SVC » pour les besoins propres du SVC, et une PV « SVC_VERCAC » qui contiendra un appel à VERCAC et pas d'appel à VELREV ni à VERCER. Les chemins correspondants sont calculés par PRECAL (voir §4.1.5 - Implémentation de CRLVerify

Le processus CRLVerify est paramétré par :

- un intervalle de vérification : c'est l'intervalle entre deux réveils de CRLVerify. A chaque réveil, CRLVerify parcourt la liste des LCR référencées dans le SVC et inspecte leur date de fin de validité (champ NextUpdate de la base).
- un délai d'attente avant alerte : si NextUpdate est périmé depuis plus que ce délai d'attente, CRLVerify positionne la LCR en invalide et l'AC en erreur.

Composant de précalcul des chemins de certification).

4.1.5 Implémentation de CRLVerify

Le processus CRLVerify est paramétré par :

- un intervalle de vérification : c'est l'intervalle entre deux réveils de CRLVerify. A chaque réveil, CRLVerify parcourt la liste des LCR référencées dans le SVC et inspecte leur date de fin de validité (champ NextUpdate de la base).
- un délai d'attente avant alerte : si NextUpdate est périmé depuis plus que ce délai d'attente, CRLVerify positionne la LCR en invalide et l'AC en erreur.

4.2 Composant de précalcul des chemins de certification

Ce composant est déclenché par le composant d'administration.

Il construit un chemin de certification pour une politique de validation et un domaine de confiance pré-définis. Le principe est le suivant : à partir de la politique et du domaine de confiance définis, il suffit d'introduire un certificat utilisateur et d'appliquer l'algorithme de construction.

Il utilise les paramètres de construction présents dans la PV et la liste des AC dignes de confiance (champ `trustable` de la table `ca`) valides, obtenues à partir du domaine de confiance.

La liste des AC racines de confiance est utilisée lors de la construction des chemins, par la méthode `addTrustAnchor`().

L'implémentation retenue est basée sur l'implémentation effectuée par SUN dans le JDK1.4.(package `certpath validation`). Cette implémentation garantit le respect des normes de la RFC 3280.

La validation d'un chemin précalculé insère le chemin dans la table validpath :

- pour le domaine de confiance et la PV considérée,
- pour le domaine de confiance SVC et la PV « SVC_VERCAC » pour les composantes de ce chemin non encore renseignées pour le domaine SVC et cette PV « SVC_VERCAC ».

Cette dernière insertion est effectuée à l'intention de CRLProcess.

4.3 Composant d'administration

4.3.1 Présentation générale

L'interface d'administration permet de maintenir à jour la base de données sur laquelle s'appuie le SVC. Les fonctions rendues sont décrites dans le document [1].

4.3.2 Choix techniques

L'interface d'administration est accessible dans un navigateur Web Internet Explorer version 6.0 ou supérieur. Le code applicatif s'exécute sur un serveur d'application JBoss.

4.3.3 Mécanisme d'authentification

L'authentification du client se réalise par vérification du couple login/mot de passe dans la table `admuser`.

4.3.4 Mécanisme de contrôle d'accès

Une fois l'authentification réussie, le login de l'administrateur permet la récupération du rôle dans la table `admuser`.

4.3.5 Ergonomie de présentation

Les principes de navigation dans l'interface d'administration sont les suivants :

- application mono-fenêtrée (pas de FRAMES HTML),
- adaptation de l'interface aux processus utilisateurs les plus fréquents.

L'ergonomie des pages est décrite dans le document [2].

4.3.6 Archivage

Tous les mouvements dans la base de données donnent lieu à une écriture dans la table `log`.

La validation d'une PV donne lieu à la génération d'un fichier texte de PV, signé par l'application d'administration et stocké sur le serveur d'administration. Ce fichier est destiné à être archivé via les fonctions d'archivage de l'ADP.

Le format de ce fichier est le suivant :

```
<?xml version="1.0"?>
<signedvalidationpolicy>
<svc version="2.00"/>
<vp>
     <idvp>identifiant de la PV</idvp>
     <version>version de la PV</version>
     <name>nom de la PV</name>
     <isvalid>[A|V|R]</isvalid>
     <isactive>[true|false]</isactive>
     <activationdate>si appel lors de la validation, vide. si appel lors
de l'activation, isactive = true et activationdate est la date de
l'activation de la PV. si appel lors de la désactivation, isactive = false
et activationdate est la date de désactivation de la PV.</activationdate>
     <vercer>
          <vercer-active>[true|false]</vercer-active>
          <vercer-ac>[true|false]</vercer-ac>
          <vercer-extendedKeyUsage>
               <eku name="nom" value="OID"/>
               ...
               <eku name="nom" value="OID"/>
          </vercer-extendedKeyUsage>
     </vercer>
     <valrev>
          <valrev-active>[true|false]</valrev-active>
          <valrev-unknown>[valid|revoked]</valrev-unknown>
     </valrev>
     <vercac>
          <vercac-active>[true|false]</vercac-active>
          <vercac-pathLengthConstraint>n</vercac-pathLengthConstraint>
     </vercac>
     <sigrep>
          <sigrep-active>[true|false]</sigrep-active>
     </sigrep>
     <creation>
          <creation-userid>identifiant créateur</creation-userid>
          <creation-date>date création</creation-date>
```

```
            <creation-comment>commentaire création</creation-comment>
      </creation>
      <validation>
            <validation-userid>identifiant validateur</validation-userid>
            <validation-date>date validation</validation-date>
            <validation-comment>commentaire validation</validation-comment>
      </validation>
</vp>
<signature>signature de l'ensemble de la réponse par le serveur
d'administration</signature>
<certs>
      <certificate>certificat du serveur d'administration</certificate>
      <cacertificate>certificat de son AC terminale</cacertificate>
</certs>
</signedvalidationpolicy>
```

4.4 Service d'archivage

L'archivage est effectué de la manière suivante :

- archivage des fichiers de logs par mise en oeuvre de `logrotate`,
- archivage de la table `log` de la base de données,
- archivage manuel des fichiers de PV signés, dans l'ADP, par l'administrateur de l'ADP.

4.5 Service d'horodatage

L'horodatage ne fait pas partie du périmètre du SVC de niveau 2.

5 Modèle de données

5.1.1 Introduction

Le modèle de données du SVC de niveau 2 complète le modèle de données du SVC de niveau 1, et s'ajoute dans la même base de données.

La technologie de base de données retenue est Oracle 8iV2 sous Linux (master DGI).

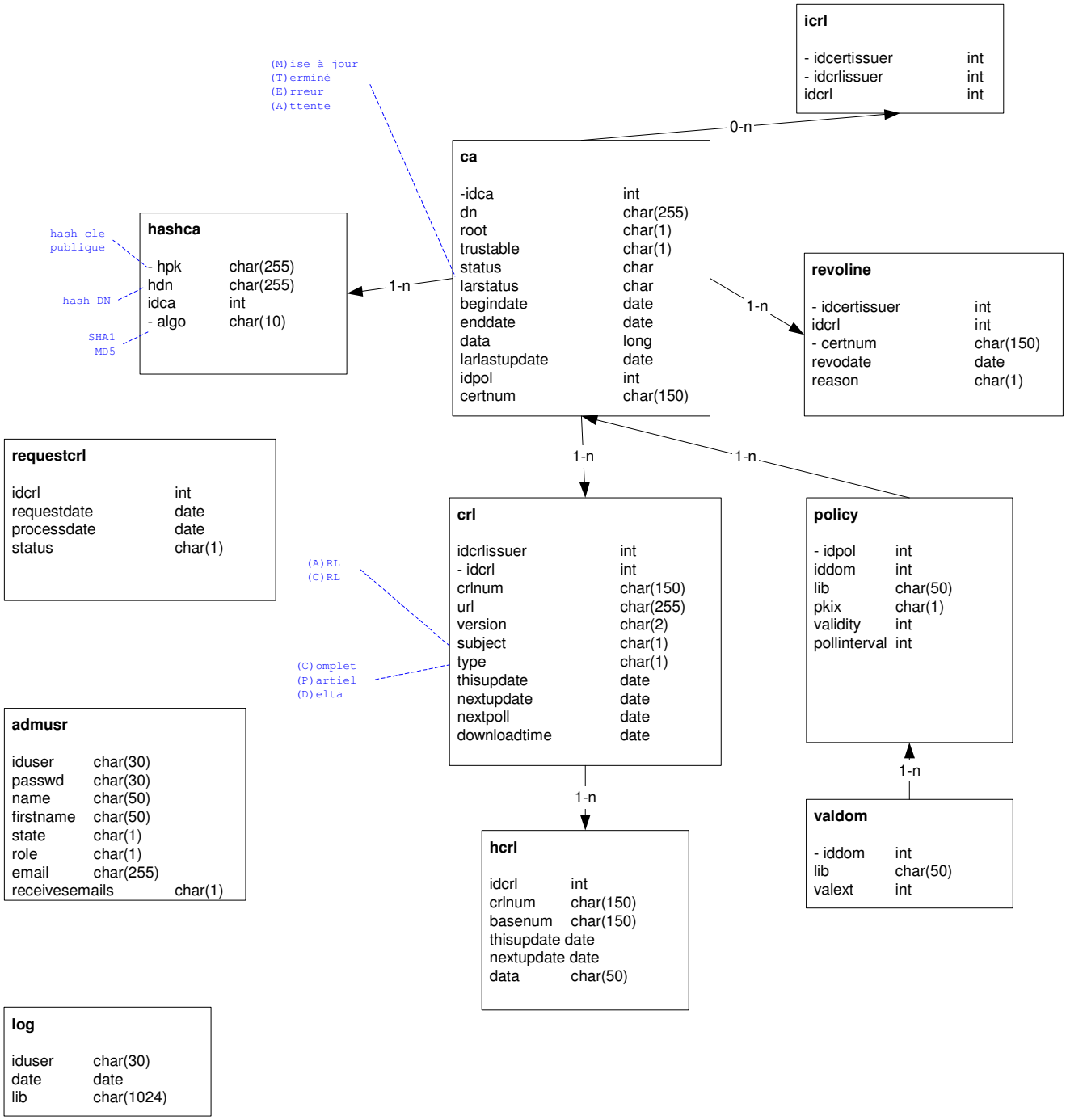
5.1.2 Compatibilité ascendante des données

Le modèle de données du SVC de niveau 1 est repris intégralement, à l'exception de la table `capath` qui est remplacée par la table plus riche `validpath`.

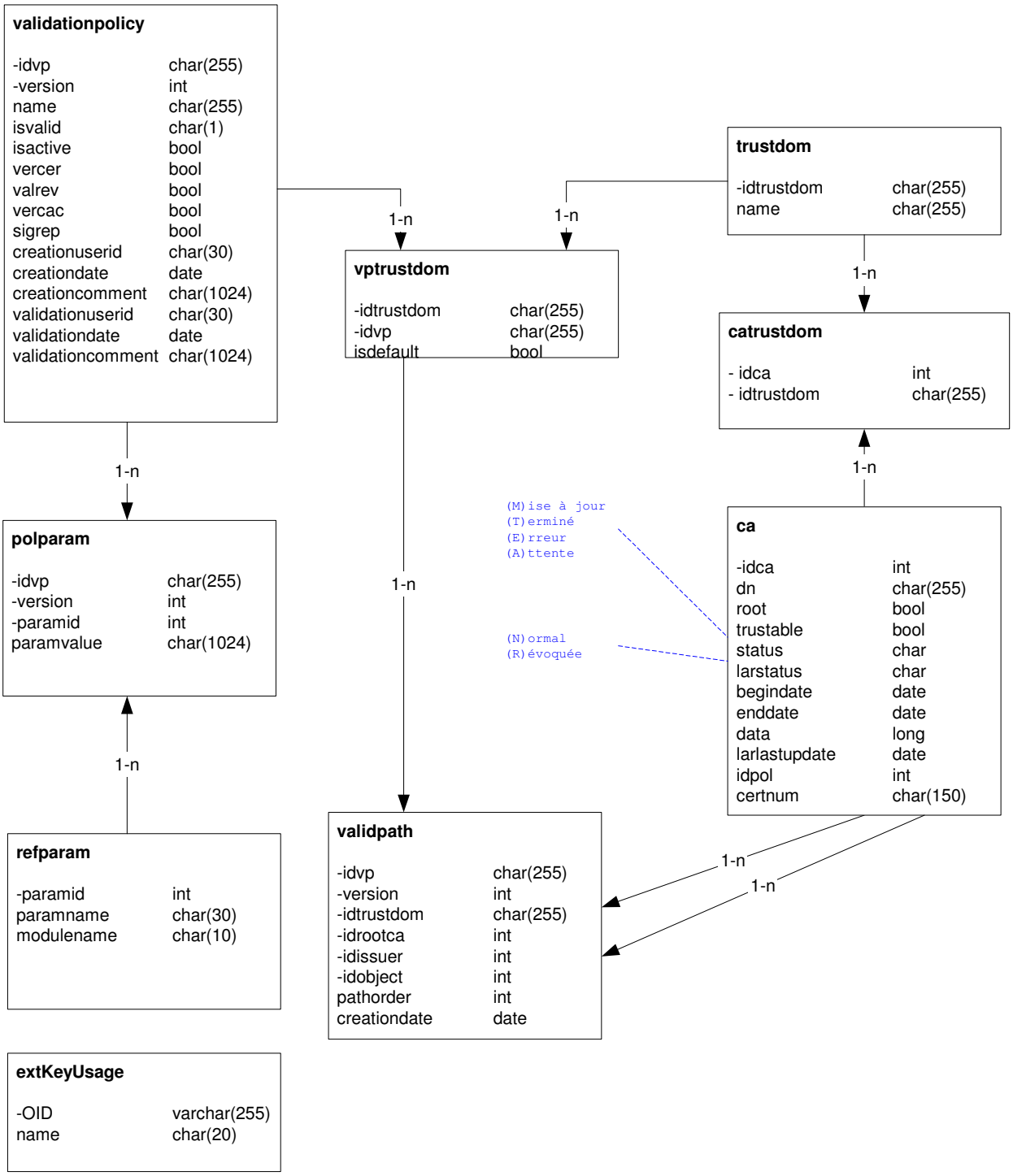
Ceci ne pose pas de problème de compatibilité ascendante, la table `capath` du niveau 1 n'étant utilisée que par l'interface d'administration. Le modèle de données du SVC de niveau 2 n'est compatible qu'avec l'interface d'administration de niveau 2, mais les services de niveau 1 déployés (répondeur OCSP ou service `valrevcer`) fonctionnent sur la base du modèle de données de niveau 2.

5.2 Modèle de données du SVC de niveau 2

5.2.1 Enrichissement des tables existantes (niveau 1)



5.2.2 Nouvelles tables (niveau 2)



5.3 Description détaillée

5.3.1 Politiques de publication

valdom : table des domaines de validation

champ	description	type
`iddom`	identifiant SVC du domaine de validation	`int`
`lib`	libellé du domaine de validation	`char(50)`
`valext`	extension de validité (durée de validité d'une LCR au-delà de sa date de péremption).	`int`

policy : table des politiques de publication des LCR

champ	description	type
`idpol`	identifiant SVC de la politique de publication	`int`
`iddom`	identifiant du domaine de validation associé	`int`
`lib`	libellé de la politique de publication	`char(50)`
`pkix`	booléen indiquant si la politique de publication des LCR respecte ou non la norme PKIX, c'est-à-dire si le champ NextUpdate de la LCR est à prendre en compte ou non pour déterminer la fin de validité d'une LCR.	`char(1)`
`validity`	Période de validité d'une LCR (en heures). Non renseigné si PKIX est vrai. Ce champ permet de recalculer le champ NextUpdate de la LCR.	`int`
`pollinterval`	Intervalle entre deux récupérations des LCR (en minutes). L'intervalle de récupération peut être inférieur à la période de validité de la LCR. Non renseigné si PKIX est vrai.	`int`

Exemple de paramétrage de politiques de publication :

Table policy	NON PKIX	PKIX
`validity`	24 heures	Non renseigné. Par définition validity = nextUpdate - thisUpdate de la LCR
`pollinterval`	60 minutes	Non renseigné. Par définition pollinterval = nextUpdate – thisUpdate de la LCR

5.3.2 Autorité de certification

ca : table des AC

champ	description	type
`idca`	identifiant SVC de l'AC	`int`
`dn`	DN de l'AC, en minuscules	`char(255)`
`root`	AC racine ou non	`char(1)`
`trustable`	AC de confiance ou non	`char(1)`
`status`	statut de l'AC (en attente, terminé, terminé sans LCR, erreur, en cours de mise à jour)	`char`
`larstatus`	statut OK ou révoqué	`char(1)`
`begindate`	date de début de validité de l'AC (information présente dans le certificat, non modifiable)	`date`
`enddate`	date de fin de validité de l'AC (information présente dans le certificat, non modifiable)	`date`
`data`	contenu du certificat	`long raw`
`larlastupdate`	date de dernière modification du champ `larstatus`	`date`
`idpol`	identifiant de la politique de publication	`int`
`certnum`	numéro de série du certificat de l'AC	`char(150)`

hashca : table des hashs des AC

champ	description	type
`hpk`	hash de la clef publique de l'AC	`char(255)`
`hdn`	hash du DN de l'AC	`char(255)`
`idca`	identifiant SVC de l'AC	`int`
`algo`	algorithme de hash (SHA1 ou MD5)	`char(10)`

5.3.3 Listes de certificats révoqués

crl : table des LCR actives

champ	description	type
idcrlissuer	identifiant SVC de l'AC signataire de la LCR	int
idcrl	identifiant SVC de la LCR	int
crlnum	numéro de série de la LCR	char(150)
url	URL de récupération de la LCR	char(255)
version	version de la LCR	char(2)
subject	catégorie de LCR : LCR ou LAR	char(1)
type	type de LCR : complète, partielle ou delta	char(1)
thisupdate	date de publication de la LCR par l'AC	date
nextupdate	date de péremption de la LCR	date
nextpoll	date de la prochaine récupération de LCR	date
downloadtime	jour et heure de récupération de la CRL	date

En fonction de la politique de publication, après chaque opération d'import réussie par le CRLProcess, les champs `thisupdate`, `nextupdate` et `nextpoll` sont mis à jour de la façon suivante :

Table crl	NON PKIX	PKIX
thisupdate	prend la valeur du champ `thisUpdate` de la LCR lorsqu'une nouvelle liste est intégrée.	champ `thisUpdate` de la LCR
nextupdate	prend la valeur `thisUpdate` + `validity`	champ `nextUpdate` de la LCR
nextpoll	prend la valeur `thisUpdate` + `pollintervall`	champ `nextUpdate` de la LCR

hcrl : table historique des LCR

champ	description	type
idcrl	identifiant SVC de la LCR	int
crlnum	numéro de la LCR	char(150)
basenum	numéro de la LCR de base dans le cas des delta LCR	char(150)
thisupdate	date de publication de la LCR	date
nextupdate	date de fin de validité de la LCR	date
data	contenu de la LCR	char(50)

icrl : table des LCR indirectes

champ	description	type
idcertissuer	identifiant SVC de l'AC émettrice des certificats	int
idcrlissuer	identifiant SVC de l'AC signataire de la LCR	int
idcrl	identifiant SVC de la LCR	int

revoline : table des certificats révoqués

champ	description	type
idcertissuer	identifiant SVC de l'AC signataire de la LCR	int
idcrl	identifiant SVC de la LCR	int
certnum	numéro du certificat révoqué	char(150)
revodate	date de révocation	date
reason	cause de révocation	char(1)

requestcrl : table des demandes de récupération exceptionnelle

champ	description	type
idcrl	identifiant SVC de la LCR	int
requestdate	date de requête	date
processdate	date de traitement de la demande	date
status	statut de la demande	char(1)

5.3.4 Domaines de confiance

trustdom : table des domaines de confiance

champ	description	type
idtrustdom	OID du domaine de confiance	char(255)
name	nom convivial du domaine de confiance	char(255)

catrustdom : table d'association entre AC et domaines de confiance

champ	description	type
idtrustdom	OID du domaine de confiance	char(255)
idca	identifiant de l'AC	int

5.3.5 Politiques de validation

vp : table des politiques de validation

champ	description	type
idvp	identifiant de la PV	char(255)
version	version de la PV	int
name	nom convivial de la PV	char(255)
isvalid	état de la PV (cycle de vie)	char
isactive	état de la PV (active ou non)	bool
vercer	activation ou non de VERCER	bool
valrev	activation ou non de VALREV	bool
vercac	activation ou non de VERCAC	bool
sigrep	activation ou non de SIGREP	bool
creationuserid	identifiant de l'utilisateur ayant créé la PV	char(30)
creationdate	date de création de la PV	date
creationcomment	commentaire de création de la PV	char(1024)
validationuserid	identifiant de l'utilisateur ayant validé la PV	char(30)
validationdate	date de validation de la PV	date
validationcomment	commentaire de validation de la PV	char(1024)

polparam : table des paramètres des politiques de validation

champ	description	type
idvp	identifiant de la PV	char(255)
version	version de la PV	int
paramid	identifiant du paramètre	int
paramvalue	valeur du paramètre	char(1024)

refparam : table de référence des paramètres des politiques de validation

champ	description	type
paramid	identifiant du paramètre	int
paramname	nom du paramètre	char(10)
modulename	nom du module	char(10)

vptrustdom : table d'association entre PV et domaines de confiance

champ	description	type
idtrustdom	OID du domaine de confiance	char(255)
idvp	identifiant de la PV	char(255)
isdefault	état de la PV (par défaut ou non)	bool

5.3.6 Chemins de certification

validpath: table des chemins de certification

champ	description	type
idvp	identifiant de la PV	char(255)
version	version de la PV	int
idtrustdom	OID du domaine de confiance	char(255)
idrootca	identifiant de l'AC racine	int
idissuer	identifiant de l'AC mère	int
idobject	identifiant de l'AC fille	int
pathorder	rang de l'AC de dn idsub dans la chaîne	int
creationdate	date à laquelle le chemin a été calculé	date

5.3.7 Extentions d'usage du certificat

extKeyUsage: table des valeurs de champ extendedKeyUsage

champ	description	type
OID	OID du champ extendedKeyUsage	char(255)
name	nom du champ extendedKeyUsage	char(20)

5.3.8 Administration de l'application

admuser : table des administrateurs de l'application

champ	description	type
`iduser`	login de l'administrateur	`char(30)`
`passwd`	mot de passe de l'administrateur	`char(30)`
`name`	nom de l'administrateur	`char(50)`
`firstname`	prénom de l'administrateur	`char(50)`
`state`	état de l'administrateur : (A)utorisé ou (S)uspendu	`char(1)`
`role`	rôle de l'administrateur : (E)xploitant ou (R)esponsable	`char(1)`
`email`	adresse email de l'administrateur	`char(256)`
`receivesemails`	indicateur de réception ou non des emails de supervision	`char(1)`

log : table des actions d'administration

champ	description	type
`iduser`	login de l'administrateur	`char(30)`
`date`	date de l'action	`date`
`lib`	libellé de l'action	`char(1024)`

5.4 Initialisation de la base de données

Les enregistrements suivants sont insérés dans le SVC à l'initialisation du système :

- politique de publication 'MINEFI', `valdom` = 24.
- valeurs de extendedKeyUsage (RFC 3280) :

extendedKeyUsage	OID
`serverAuth`	1.3.6.1.5.5.7.3.1
`clientAuth`	1.3.6.1.5.5.7.3.2
`codeSigning`	1.3.6.1.5.5.7.3.3
`emailProtection`	1.3.6.1.5.5.7.3.4
`timeStamping`	1.3.6.1.5.5.7.3.8
`OCSPSigning`	1.3.6.1.5.5.7.3.9

- domaine de confiance SVC : OID 1.2.250.1.131.1.5.4.7.1.0
- politique de validation « SVC_ADMIN » : OID 1.2.250.1.131.1.5.4.6.1.0, version 1, valide, active, VERCER, cA = « true », extendedKeyUsage vide.
- politique de validation « SVC_VERCAC » : OID 1.2.250.1.131.1.5.4.6.1.1, version 1, valide, active, VERCAC, pathLengthConstraint non renseigné.
- association entre SVC et SVC_ADMIN.

- association entre SVC et SVC_VERCAC.
- champs à insérer dans la table refparam :

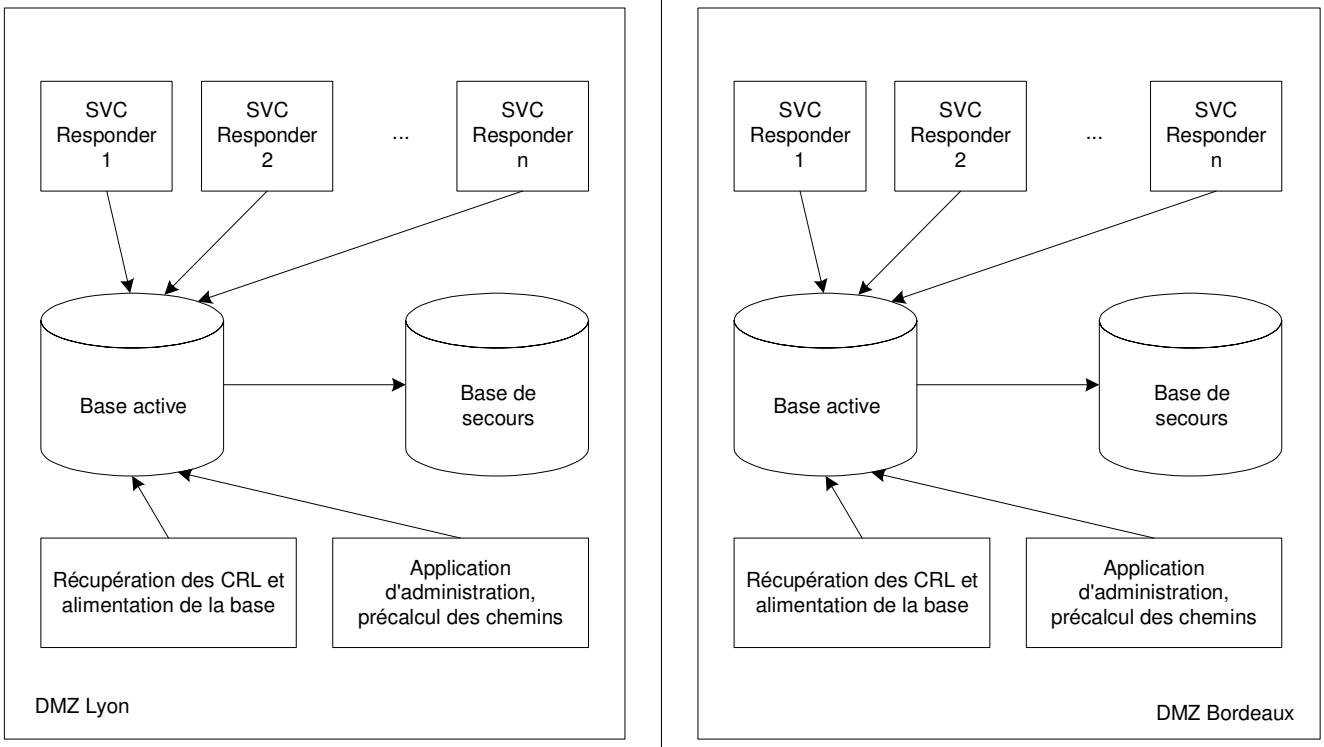
composant utilisant ce paramètre	nom	description	paramid	contenu
valrev	`unknown`	réponse de VALREV (`valide` ou `révoqué)` en cas de réponse `unknown` du service OCSP.	0	« R » ou « V »
vercer	`extendedkeyUsage`	valeur autorisée pour le champ d'extension extendedKeyUsage (OID).	1	Liste d'OID
vercer	`aC`	champ indiquant si le certificat présenté est un certificat d'AC ou un certificat d'utilisateur.	2	`true` ou `false`
vercac	`pathLengthConstraint`	profondeur maximale de l'arbre acceptée par l'AC racine.	3	nombre entier

- un administrateur responsable à définir (`admin, password` pour les tests).

6 Architecture physique du SVC

6.1 Architecture physique générale

L'architecture technique du SVC est représentée sur le schéma ci-dessous.



Le déploiement du SVC se fait dans les DMZ, à proximité des applications clientes.
Le déploiement du back-office se fait également dans les DMZ.

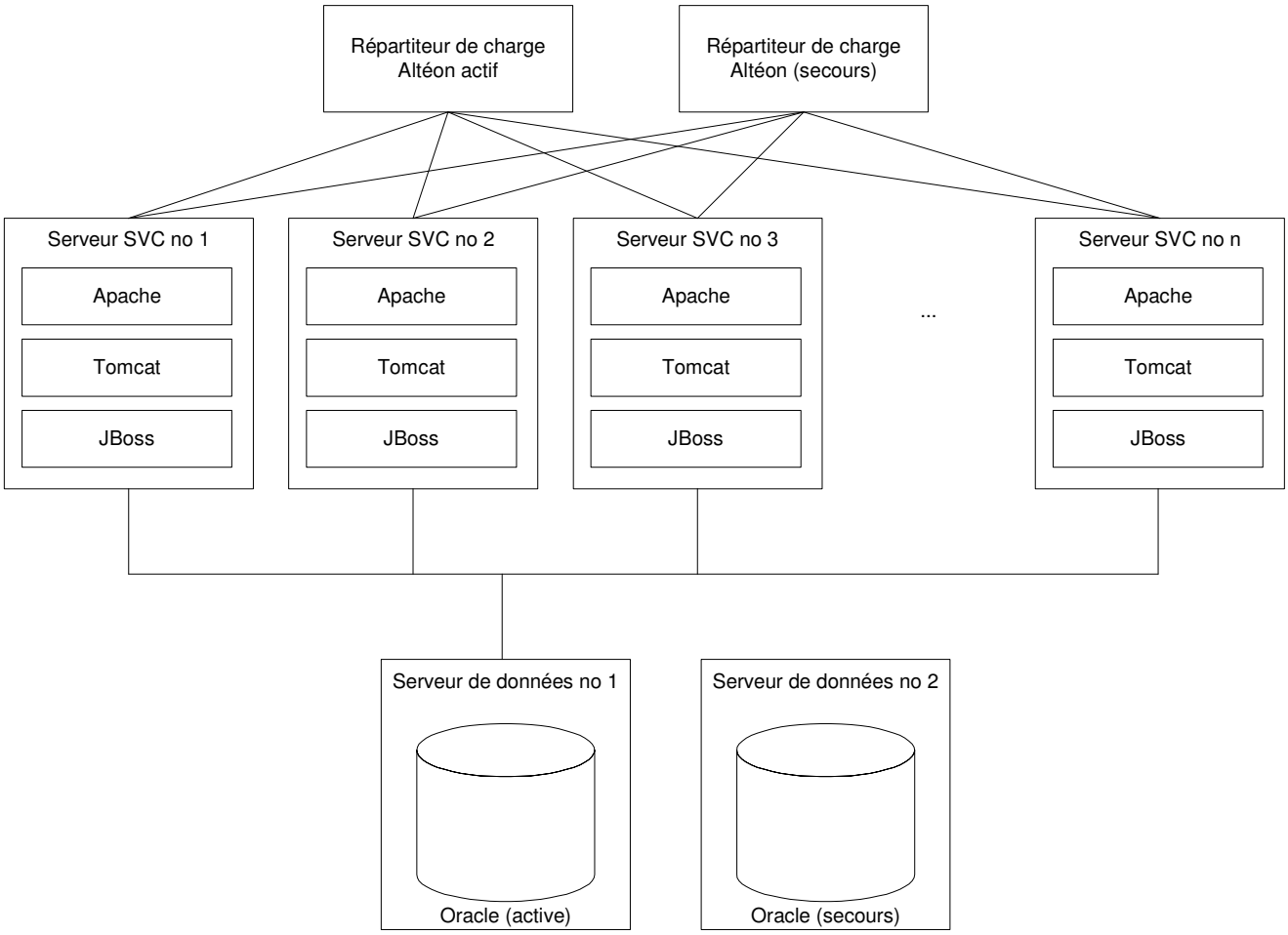
L'autorité de validation est une autorité logique mais les bases de données sont physiquement distribuées entre les DMZ hébergeant le service.

La présence de la base de données dans les DMZ à proximité des répondeurs permet de garantir des temps de réponse optimaux au niveau du répondeur SVC.

En cas de panne, l'ensemble des applications doit être reconfigurée afin de basculer sur la base Oracle de secours.

6.2 Architecture physique du front-office

Afin d'absorber la charge de sollicitation prévue pour le SVC de niveau 2, plusieurs instances de SVC peuvent être déployées derrière un répartiteur de charge matériel (Altéon). L'architecture de déploiement correspondant à ce cas est présentée sur le schéma ci-dessous.



L' architecture représentée ci-dessus offre de multiples avantages :

- simplicité de l'exploitation : tous les serveurs SVC ont la même architecture et suivent les mêmes procédures d'exploitation,
- limitation des mécanismes de répartition de charge : la répartition de charge est faite au niveau de l'Altéon en fonction des charges des serveurs SVC, et aucune perte de temps n'est due à des répartitions de charge complémentaires, ce qui serait le cas s'il était nécessaire de séparer physiquement les serveurs Web des serveurs d'application (par mise en oeuvre de mod_jk).

6.3 Architecture physique du back-office

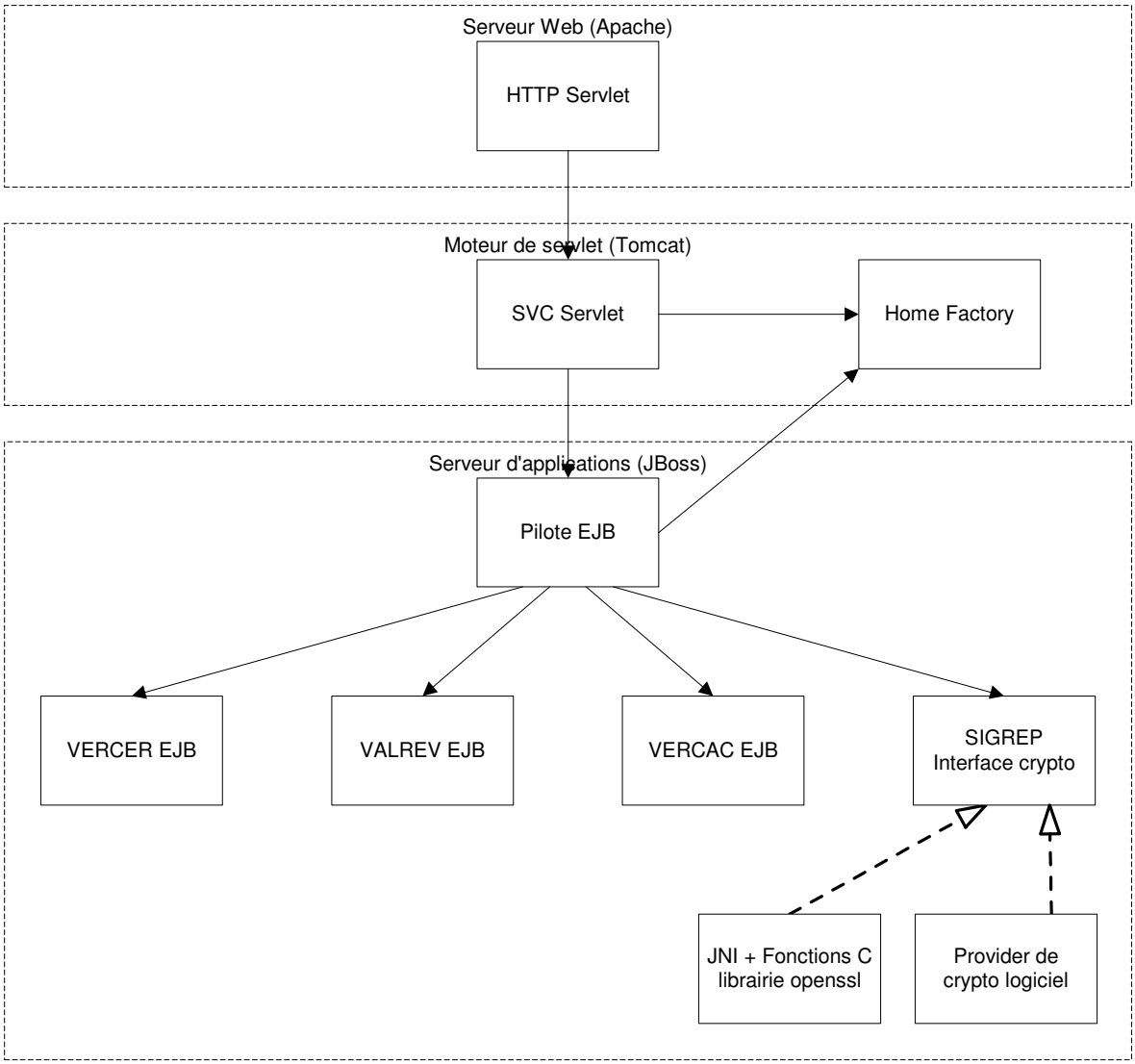
L'architecture physique du back-office est identique à celle du SVC de niveau 1.

7 Architecture logicielle du front-office

L'implémentation de l'ensemble des services de « front-office » est réalisée de façon à profiter au maximum des capacités de Tomcat et JBoss concernant l'absorption de montée en charge. Pour cela, on sépare dans des EJB différents le pilote et chacun des composants VERCER, VALREV, VERCAC et SIGREP. Selon les vitesses d'exécution de chacun des composants en réponse à une sollicitation, le serveur JBoss instancie pour chacun des Beans le nombre de threads nécessaires à la répartition de charge.

Lors du déploiement, l'ensemble des Beans est déployé sur chacune des instances de SVC, afin d'éviter les communications entre machines et donc la sérialisation : chaque pilote communique localement avec les différents composants qu'il appelle.

L'architecture logicielle de chaque instance de SVC est la suivante :



Le service SVC est appelé sur une URL d'entrée unique. Le SVC servlet identifie le type de requête reçue (OCSP, OCSP étendue ou requête enrichie). Il n'effectue pas d'appel en base pour récupérer

la PV correspondant à l'appel. Il transmet la requête au pilote EJB qui gère le processus de validation par appel aux différents EJB correspondants aux différents services.

8 Mécanismes d'exploitation du SVC

8.1 Mécanismes de sauvegarde/restauration

Ces mécanismes sont spécifiés dans le document [3].

8.2 Gestion des logs

Ces mécanismes sont spécifiés dans le document [3].

9 Formats d'échange

Les formats d'échange utilisés dans le SVC de niveau 2 sont normalisés :

- Les certificats manipulés sont au format PEM.
- Les requêtes et les réponses OCSP sont au format DER.
- Les formats d'échange de haut niveau entre applications sont au format XML.
- Les dates sont au format UTCTime.

10 Supervision

La supervision de l'application se décline en supervision d'infrastructure, supervision applicative et supervision fonctionnelle.

10.1 Supervision de l'infrastructure

L'infrastructure SVC (bases Oracle, serveurs Apache/Tomcat/JBoss) est supervisée sur la base des supervisions spécifiées par le bureau SI3 (équipe ESR de la DAI).

10.2 Supervision applicative

La supervision applicative est effectuée par l'envoi de traps SNMP V2 qui sont collectées par un collecteur Nagios et transmises via un tunnel de communication à un serveur Nagios pour présentation à l'administrateur de la supervision.

De plus, l'envoi de tout trap SNMP génère une écriture dans un fichier de log. Ce fichier n'est pas supervisé par Nagios.

Afin que les traps SNMP émises soient intégrées dans Nagios via le tunnel de communication établi par ESR, elles doivent respecter le format suivant :

```
/HOST=host/SERVICE=service/STATUS=statut/DATE=date/commentaire et infos de
contexte

* host = nom court de la machine sur laquelle le code s'exécute

* service = le nom du composant affecté

* statut = [0|1|2|3|OK|WARNING|CRITICAL|UNKNOWN]
OK ou 0 = tout va bien
WARNING ou 1 = le service est rendu, avec éventuellement une dégradation
CRITICAL ou 2 = le service n'est pas rendu, dysfonctionnement grave
UNKNOWN ou 3 = on ne peut pas conclure

* date = timestamp unix

* commentaire = le libellé de description de la MIB

* infos de contexte : la valeur des  variables concernées dans l'application
```

Exemple :

```
"11/2/2004:5:50:13 trap from 99.4.23.78 ccitt.1 = STRING: "/HOST=SVCSRV-
01/SERVICE=APPL_etat_svcResponder/STATUS=OK/DATE=1083659030/commentaire"
```

Liste des traps émis

Service	Notification	Commentaire	Statut
APPL_etat_BDfrontoffice	svcresponderbddconnect	SVC : Problème de connexion à la base de données	Critical
	svcresponderbddinterro	SVC : Problème d'interrogation de la base de données	Critical
	BDfrontofficeOK	SVC : Base de données accessible	OK (*)
APPL_etat_SVC Responder	chargementcertif	SVC : Problème de chargement du certificat du serveur	Critical
	SVCResponderOK	SVC : Certificat serveur chargé	OK (*)
APPL_etat_PILOTE	connexionpilote	SVC : Connexion à l'EJB Pilote impossible	Critical
	PILOTEOK	SVC : Connexion à l'EJB Pilote opérationnelle	OK (*)
APPL_etat_VERCER	connexionvercer	SVC : Connexion à l'EJB VERCER impossible	Critical
	VERCEROK	SVC : Connexion à l'EJB VERCER opérationnelle	OK (*)
APPL_etat_VALREV	connexionvalrev	SVC : Connexion à l'EJB VALREV impossible	Critical
	VALREVOK	SVC : Connexion à l'EJB VALREV opérationnelle	OK (*)
APPL_etat_VERCAC	connexionvercac	SVC : Connexion à l'EJB VERCAC impossible	Critical
	VERCACOK	SVC : Connexion à l'EJB VERCER opérationnelle	OK (*)
APPL_etat_SIGREP	connexionsigrep	SVC : Connexion à l'EJB SIGREP impossible	Critical
	SIGREPOK	SVC : Connexion à l'EJB VERCAC opérationnelle	OK (*)
APPL_etat_BDbackoffice	crlprocessbddconnect	SVC : Problème de connexion à la base de données	Critical
	BDbackofficeOK	SVC : Connexion à la base de données opérationnelle	OK
APPL_etat_CRL Process	processconnexioncrlfind	SVC : Problème de connexion au module CRL Finder	Critical
APPL_etat_CRL Process_idAC	processlcrnonconforme	SVC : Le fichier LCR n'est pas conforme pour l'AC :	Warning
	processrecuplcr	SVC : Problème de récupération du fichier LCR sur le serveur pour l'AC :	Warning
	processmajlcr	SVC : Problème de mise à jour de la LCR pour l'AC :	Warning

Service	Notification	Commentaire	Statut
	processcrlfindtimeout	SVC : L'URL ne répond pas dans les temps impartis pour l'AC :	Warning
	processnosignaturealgorithm	SVC : La LCR téléchargée n'est pas valide, absence du champ signaturealgorithm. AC concernée :	Warning
	processbadsignature	SVC : La LCR téléchargée n'est pas valide, signature non vérifiée. AC concernée :	Warning
	processbadthisupdate	SVC : La LCR téléchargée n'est pas valide, champ thisupdate absent. AC concernée :	Warning
	processnonextupdate	SVC : Absence du champ nextupdate pour cette LCR qui est censée respecter la norme Pkix. AC concernée :	Warning
	processnorevokedcertificates	SVC : La LCR téléchargée n'est pas valide, absence du champ revokedcertificates. AC concernée :	Warning
	processbadcrlnumber	SVC : La LCR téléchargée n'est pas valide, le champ CRLNumber n'est pas strictement croissant. AC concernée :	Warning
	processnoissuer	SVC : La LCR téléchargée n'est pas valide, absence du champ issuer. AC concernée :	Warning
	processbadchain	SVC : La LCR téléchargée n'est pas valide, chemin de certification invalide. AC concernée :	Warning
	processbadissuer	SVC : La LCR téléchargée n'est pas valide, son émetteur n'est ni l'émetteur du certificat ni celui indiqué dans CRLDistributionPoints. AC concernée :	Warning
	processbadindirect	SVC : La LCR téléchargée n'est pas valide, elle est indirecte mais les champs indirectCRL et certificateIssuer sont absents. AC concernée :	Warning
	processrevokedacincrl	SVC : L'AC émettrice de cette LCR est révoquée dans la LCR. La LCR n'est pas fiable et n'a pas été intégrée automatiquement dans la base de données. AC concernée :	Warning
	CRLProcessOK	SVC : La LCR téléchargée est valide et injectée dans la base de données. AC concernée :	OK (**)
APPL_etat_CRLFinder_idAC	finderbadthisupdate	SVC : La LCR téléchargée n'est pas valide, champ thisupdate absent. AC concernée :	Warning
	findernosignaturealgorithm	SVC : La LCR téléchargée n'est pas valide, absence du champ signaturealgorithm. AC concernée :	Warning
	finderbadsignature	SVC : La LCR téléchargée n'est pas valide, signature non vérifiée. AC concernée :	Warning

Service	Notification	Commentaire	Statut
	finderbaddownloadtime	SVC : La LCR téléchargée n'est pas valide, heure de récupération non comprise entre thisupdate et nextupdate. AC concernée :	Warning
	CRLFinderOK	SVC : La LCR téléchargée est valide et a été transmise à CRLProcess. AC concernée :	OK (**)
APPL_etat_CRLVerify_idAC	lcrperimee	SVC : La LCR est périmée pour l'AC :	Warning
	desactiverac	SVC : La LCR est périmée, désactivation automatique de l'AC :	Warning
	CRLVerifyOK	SVC : La LCR est valide pour l'AC :	OK (***)
APPL_etat_hotarch	HOTARCHOK	SVC : Le script hotarch.pl s'est exécuté sans erreur critique.	OK
	OPENLOG	SVC : Impossible de créer le fichier de LOG : <nom du fichier>	Critical
	SID	SVC : Impossible d'exporter la variable ORACLE_SID par la commande : <commande>	Critical
	RUNNING	SVC : Script déjà en cours d'exécution	Critical
	OPENLOCK	SVC : Impossible de créer le fichier LOCK	Critical
	CLOSELOCK	SVC : Impossible de fermer le fichier LOCK	Warning
	DBNAME	SVC : Nom de l'instance de base de données inattendu : <nom de l'instance>	Critical
	NOARCH	SVC : Mode ARCHIVELOG non positionné sur la base de données.	Critical
	NOAUTOARCH	SVC : Mode ARCHIVELOG desactivé sur la base de données.	Critical
	NOARCHDEST	SVC : Pas de destination pour les archives.	Critical
	STOPPED	SVC : Programme interrompu : <cause>	Critical
	COPY	SVC : Erreur technique de copie, fichier : <nom du fichier>	Critical
	DELLOCK	SVC : Impossible de supprimer le fichier LOCK	Warning
	OPENSQL	SVC : Impossible de créer le fichier SQL	Critical
	SQLERROR	SVC : Erreur Oracle : <libellé de l'erreur>	Critical
	ERRARCHLOG	SVC : Impossible de récupérer les informations liées au mode ARCHIVELOG.	Critical
	ALTERTBLSP	SVC : Impossible de modifier le tablespace : <nom>	Critical
	BADFILENAME	SVC : Nom de fichier incorrect : <nom>	Critical
	NODATAFILES	SVC : Aucun fichier de données disponible.	Critical
	ERRRM	SVC : Suppression des archives obsolètes impossible dans le répertoire : <nom>	Critical

Service	Notification	Commentaire	Statut
	OPENHDR	SVC : Impossible de créer le fichier Header :	Critical
	CANNOTBACKUP	SVC : Impossible de sauvegarder le fichier de contrôle.	Critical
	CANNOTSWITCH	SVC : Impossible de basculer le fichier de log.	Critical
	NOFILE	SVC : Absence du fichier : <nom>	Critical
	NODIRECTORY	SVC : Impossible d'accéder au répertoire : <nom>	Critical
	NOTAR	SVC : Impossible de compresser par la commande : <commande>	Critical
	SUSPECT	SVC : Archive suspecte, taille des fichiers incorrecte pour le fichier : <nom du fichier>	Critical
	NOSCP	SVC : Copie impossible par la commande : <commande>	Critical
	NOSTART	SVC : Impossible de créer le fichier témoin :	Critical
APPL_etat_hotrest	HOTRESTOK	SVC : Le script hotrest.pl s'est exécuté sans erreur critique. Dérouler la procédure de recovery complémentaire avant de basculer cette base en base active.	OK
	OPENLOG	SVC : Impossible de créer le fichier de LOG : <nom du fichier>	Critical
	SID	SVC : Impossible d'exporter la variable ORACLE_SID par la commande : <commande>	Critical
	NOSTART	SVC : Absence du fichier témoin : <nom>	Critical
	NOCOPY	SVC : Impossible de copier le fichier : <nom du fichier>	Critical
	ERRPS	SVC : Impossible d'exécuter la commande ps.	Critical
	DBNAME	SVC : Nom de l'instance de base de données inattendu : <nom de l'instance>	Critical
	RUNNINGDB	SVC : Impossible de restaurer une base active.	Critical
	ERRRM	SVC : Suppression des archives obsolètes impossible dans le répertoire : <nom>	Critical
	OPENHDR	SVC : Impossible d'ouvrir le fichier Header : <nom>	Critical
	BADHDR	SVC : Impossible de traiter le fichier Header.	Critical
	NODIRECTORY	SVC : Impossible d'accéder au répertoire : <nom>	Critical

Service	Notification	Commentaire	Statut
	NOTAR	SVC : Impossible de compresser par la commande : <commande>	Critical
	NOUNTAR	SVC : Impossible de décompresser par la commande : <commande>	Critical

(*) Côté front-office, les traps OK ne sont émises que lors du démarrage réussi de l'application.

(**) Trap envoyé si l'ensemble des traitements s'est déroulé sans erreur, après modification de la base de données.

(***) Trap envoyé si la LCR est valide (aucun des autres traps n'a été envoyé)

11 Codes et messages d'erreur

11.1 Respect du standard OCSP (RFC 2560)

Dans le cas d'un appel OCSP, la codification des erreurs et des statuts de révocation respecte le standard OCSP, qui est résumé ci-dessous :

Champ	Valeur	Code	Signification
OCSPResponseStatus	successful	0	réponse valide
	malformedRequest	1	requête malformée
	internalError	2	erreur interne dans le répondeur OCSP
	tryLater	3	le serveur est momentanément indisponible pour répondre à cette requête
		4	ce code n'est pas utilisé
	sigRequired	5	la requête OCSP doit être signée
	unauthorized	6	la requête n'est pas autorisée
CertStatus	good	0	certificat valide
	revoked	1	certificat révoqué
	unknown	2	certificat en état inconnu

Si le OCSPResponseStatus est « successful », le « CertStatus » est renseigné et vaut 0, 1 ou 2.
Si le OCSPResponseStatus n'est pas « successful », le « CertStatus » n'est pas renseigné.

11.2 Compatibilité ascendante pour les messages XML

Afin de garantir la compatibilité ascendante avec le niveau 1, les codes d'erreur sont des entiers qui respectent les règles suivantes :

- codes d'erreur négatifs pour les erreurs techniques,
- codes d'erreur positifs pour les erreurs fonctionnelles.

Pour mémoire, dans le SVC de niveau 1, les codes d'erreur sont les suivants :

Code	Message	Signification
0	successful	réponse valide
-1	malformedRequest	requête malformée
-2	internalError	erreur interne dans le répondeur OCSP
-3	tryLater	le serveur est momentanément indisponible pour répondre à cette requête
-4		ce code n'est pas utilisé
-5	sigRequired	la requête OCSP doit être signée
-6	unauthorized	la requête n'est pas autorisée
-7	serverunavailable	le serveur OCSP ne répond pas

Les codes 0, -1, -2, -3, -5, -6 sont conformes au standard OCSP, au signe près.

Les statuts de révocation sont les suivants :

Code	Message	Signification
0	good	le certificat est valide
1	revoked	le certificat est révoqué
2	unknown	le statut est inconnu, par exemple le serveur ne gère pas cette AC
3	malformed	le format du certificat n'est pas conforme
4	expired	le certificat est périmé
5	notyetvalid	le certificat n'est pas encore valide
6	badsignature	la signature n'est pas intègre

Si une erreur se produit (Code différent de « successful »), le statut est égal au code d'erreur retourné.

Les codes 0, 1 et 2 sont conformes au standard OCSP.

11.3 Codification des erreurs pour le SVC de niveau 2

Dans le SVC de niveau 2, en cas de requête OCSP ou OCSP étendue, les codes d'erreur et statuts sont conformes au standard OCSP.

En cas de requête enrichie, les codes d'erreur et statuts peuvent être plus riches. Ils respectent le format suivant :

```
<responsestatus>validated|rejected</responsestatus>
<error type="S|F|T" code="xxx">message d'erreur</error>
```

11.4 Erreurs fonctionnelles

11.4.1 Codification

Les codes d'erreur fonctionnels sont groupés par tranches selon le composant qui génère l'erreur :
101 à 199 Pilote
201 à 299 VERCER
301 à 399 VALREV
401 à 499 VERCAC
501 à 599 SIGREP
601 à 699 CRLFinder
701 à 799 CRLPRocess
801 à 899 PRECAL
901 à 999 ADMIN

11.4.2 Pilote

Code	Message	Signification
101	Politique de validation ou domaine de confiance inconnus ou non associés, , ou aucune version active de cette politique de validation	PV non présente en base Ou domaine de confiance non présent en base Ou PV non associée au domaine de confiance dans la base Ou aucune version active de cette PV n'a été trouvée en base.
102	Politique de validation non correctement paramétrée	PV sans aucun service paramétré (sauf éventuellement SIGREP), ou erreur de paramétrage de la PV
103	Version de la requête XML non supportée	Le SVC n'a pas été appelé avec une version supportée du format XML.

11.4.3 VERCER

Code	Message	Signification
201	Champ version absent	Absence du champ obligatoire version
202	La version est incorrecte	La version du certificat n'est pas V3
203	Champ numéro de série absent	Absence du champ obligatoire serialnumber
204	Champ algorithme de signature absent	Absence du champ obligatoire signatureAlgorithm
205	Champ émetteur absent	Absence du champ obligatoire issuer
206	Champ clef publique absent	Absence du champ obligatoire subjectPublicKeyInfo
207	Champ sujet absent	Absence du champ obligatoire subject
208	Certificat périmé	Certificat périmé

209	Certificat pas encore valide	Certificat pas encore valide
210	Champ signature absent	Absence du champ obligatoire signature
211	Signature incorrecte	La signature n'est pas intègre
212	Champ d'extension de clef absent	Absence du champ extendedKeyUsage s'il est attendu
213	Utilisation de clef incorrecte	Valeur du champ extendedKeyUsage non conforme à la PV
214	Absence d'une date privatekeyusageperiod	notBefore absent OU notAfter absent
215	La clé privée est périmée	privatekeyusageperiod périmé
216	La clef privée n'est pas encore valide	privatekeyusageperiod pas encore valide
217	Champ ca absent	Absence du champ obligatoire ca
218	Champ ca incorrect	Valeur du champ cA non conforme à la PV
219	Champ pathlenconstraint absent	Absence du champ pathlenconstraint alors qu'il est attendu
220	Champ pathlenconstraint incorrect	Présence du champ pathlenconstraint alors qu'il est interdit
221	Champ privatekeyusageperiod invalide	notBefore n'est pas antérieur à la date de vérification OU notAfter n'est pas ultérieur à la date de vérification
222	Champ date absent	Absence du champ obligatoire validity
223	AC inconnue	AC émettrice du certificat non présente en base

11.4.4 VALREV

Code	Message	Signification
301	revoked	le certificat est révoqué

11.4.5 VERCAC

Code	Message	Signification
401	Absence de chemin de certification	Chemin de certification non présent en base
402	Chemin de certification incorrect	Chemin de certification non unique (doublon ou « trou »)
403	AC non terminale	L'AC dans le certificat n'est pas terminale
404	AC révoquée	Une des AC de la chaîne est révoquée
405	AC expirée	Une des AC de la chaîne a expiré
406	AC inconnue	AC émettrice du certificat non présente en base

11.4.6 SIGREP

Aucune erreur fonctionnelle n'est remontée par SIGREP.

11.4.7 PRECAL

Code	Message	Signification
801	Le calcul ne peut-être réalisé	Le domaine de confiance ne contient pas toutes les AC de la chaîne, ou la limite de PathLengthConstraint a été atteinte sans trouver l'AC racine
802	Le chemin calculé ne peut être validé	Le domaine de confiance ne contient pas l'AC racine
803	Erreur lors du précalcul : absence de politique de validation	La PV n'a pas été trouvée en base
804	Erreur lors du précalcul : aucune AC n'est associée au domaine de confiance	Aucune AC n'a été trouvée en base pour le domaine de confiance sélectionné.
805	Erreur lors du précalcul : la politique de validation ne concerne pas vercac	La PV n'a pas été paramétrée pour appel au service VERCAC.

11.5 Erreurs techniques

11.5.1 Codification

Les codes d'erreur techniques sont groupés par tranches selon le composant qui génère l'erreur :
-101 à -199 Pilote
-201 à -299 VERCER
-301 à -399 VALREV
-401 à -499 VERCAC
-501 à -599 SIGREP
-601 à -699 CRLFinder
-701 à -799 CRLPRocess
-801 à -899 PRECAL
-901 à -999 ADMIN

11.5.2 Pilote

Code	Message	Signification
-101	Analyse de la requête xml	La requête XML a un format incorrect.

	impossible	
-102	Analyse du certificat impossible	Le certificat a un format incorrect
-103	Requête xml incomplète	La requête XML n'a pas tous les champs obligatoires
-104	Base de données indisponible	La connexion à la base de données n'est pas disponible et les données ne sont pas accessibles.

11.5.3 VERCER

Code	Message	Signification
-201	Vérification de signature impossible : absence de Provider	Le fichier JAR contenant la librairie cryptographique Java n'est pas correctement installé.

11.5.4 VALREV

Code	Message	Signification
-301	Erreur interne du serveur VALREV (OCSP internalError)	Le répondeur OCSP interne au SVC n'est pas accessible ou n'a pas pu rendre de réponse.
-302	Erreur interne du serveur VALREV (OCSP tryLater)	Le répondeur OCSP interne au SVC est en cours de mise à jour.

11.5.5 VERCAC

Aucune erreur technique n'est remontée par VERCAC.

11.5.6 SIGREP

Aucune erreur technique n'est remontée par SIGREP lors d'une signature. Si la clef n'a pas été trouvée, le message d'erreur est émis dans les logs lors de l'initialisation du système.

11.5.7 PRECAL

Code	Message	Signification
-801	erreur de création de PKIXParameters	La liste des AC de confiance pour le domaine de confiance sélectionné est vide ou incorrecte.
-802	erreur de création de PKIXBuilderParameters	La liste des AC de confiance pour le domaine de confiance sélectionné est vide ou incorrecte, ou le paramètre PathLengthConstraint de la PV est invalide.
-803	erreur de création de certificat X509 à partir du fichier	Erreur technique SQL : connexion à la base de données impossible.
-804	erreur de recherche de la	Erreur technique SQL : connexion à la base de données

	liste des AC de confiance dans la base	impossible.
-805	erreur de recherche de la liste des AC non de confiance dans la base	Erreur technique SQL : connexion à la base de données impossible.
-806	erreur de construction du chemin de certification	Algorithme de précalcul non disponible ou paramètres invalides (absence possible du fichier JAR correspondant).
-807	erreur de validation du chemin de certification	Algorithme de validation non disponible ou paramètres invalides (absence possible du fichier JAR correspondant).
-808	erreur de recherche de la PV par défaut dans la base	Erreur technique SQL : connexion à la base de données impossible. Message non utilisé.
-809	erreur de recherche de la PV dans la base	Erreur technique SQL : connexion à la base de données impossible.
-810	erreur de recherche de chemin dans la table validpath	Erreur technique SQL : connexion à la base de données impossible.
-811	erreur d'insertion dans la table validpath	Erreur technique SQL : connexion à la base de données impossible.
-812	erreur d'archivage des chemins dans la table validpath	Erreur technique SQL : connexion à la base de données impossible. Message non utilisé.
-813	erreur d'archivage des chemins dans la table validpath avec XSQL	Erreur technique SQL : connexion à la base de données impossible.
-814	erreur de copie des chemins dans la table validpath	Erreur technique SQL : connexion à la base de données impossible. Message non utilisé.