Spécifications fonctionnelles générales du SVC de niveau 2

040186

Version 1.00

Circuit de validation

	Nom	Organisation	Date	Visa
Rédigé par :	Y.Quenec'hdu	Rédacteur	10/04/2004	
Vérifié par :	V.Sage R.PIRIM	Responsable Domaine		
Approuvé par :	P. Murzeau V. Sage	Directeur de projet DGI		

Historique des évolutions

Ver	Date	Auteur	Justificatif
0.2	15/03/2004	Y.Quenec'hdu	
0.3	8/04/2004	Y.Quenec'hdu	Ajout de la partie technique des spécifications
0.4	13/04/2004	Y.Quenec'hdu	Ajout de la partie signature et supervision
0.5	14/04/2004	Y.Quenec'hdu	Ajout de la partie gestion des erreurs
0.6	05/05/2004	Y.Quenec'hdu	- Modification du point 6 de la section 7.2.3 - ajout de traitement sur les DeltaCRL (section 7.2.3) - ajout de la révocation manuelle d'AC depuis l'interface d'administration - ajout du niveau de confiance d'un AC dans depuis l'interface d'administration pour le calcul des chemins de certification
0.7	10/05/2004	Y.Quenec'hdu	Insertion de la partie sauvegarde de base de données
0.8	10/05/2004	Y.Quenec'hdu	Insertion de la supervision
0.9	14/05/2004	Y.Quenec'hdu	Relecture et passage en version 1
0.9.1	14/05/2004	Y.Quenec'hdu	Adaptation du document au déplacement du référentiel de la zone de confiance vers la DMZ Ajout de la section WebService
0.9.2	04/08/2004	Y.Quenec'hdu	Prise en compte des remarques de V.Sage
0.9.3	09/08/2004	Y.Quenec'hdu	Relecture et correction
1.00	1/09/2004	V.Sage	Relecture et passage en version 1.00

Sommaire

1 Glossaire

Autorité de certification (AC – CA)

L'AC est responsable des Certificats signés en son nom et de l'ensemble de l'infrastructure à clé publique qu'elle a mise en place. En particulier, l'AC assure les fonctions suivantes :

- Mise en application de la Politique de Certification,
- Émission des Certificats,
- Gestion des Certificats,
- Publication de la Liste des Certificats Révoqués (LCR),
- Journalisation et archivage des événements et informations relatives au fonctionnement de l'ICP.

L'AC assure ces fonctions directement ou en les sous-traitant, tout ou partie. Dans tous les cas, l'AC en garde la responsabilité.

Back Office

Partie du SVC en charge de la mise à jour du système (administration et récupération automatique des CRL).

CAC

Bloc fonctionnel dont le périmètre est la vérification des chaînes de certification.

CER

Bloc fonctionnel dont le périmètre est la vérification de certificat.

Liste de Certificats Révoqués (CRL - LCR)

Liste comprenant les numéros de série des Certificats ayant fait l'objet d'une Révocation, signée par l'AC émettrice.

Front Office

Partie du SVC en charge de la réponse interactive aux requêtes émises par les clients demandant la validation d'un certificat.

OID

Identificateur numérique unique enregistré conformément à la norme d'enregistrement ISO pour désigner un objet ou une classe d'objets spécifiques.

OCSP (*Online Certificate Statut Protocol*)

OCSP est défini dans le RFC2560. Le but d'OCSP est de surmonter les limitations imposées par les CRL de base et de fournir une réponse immédiate et à jour aux questions sur le statut d'un certificat. Une information spécifique de révocation pour un certificat est retournée plutôt qu'une grande liste linéaire de recherche sous forme de CRL.

DeltaCRL ou Delta de LCR

Un delta de LCR liste les certificats dont le statut de révocation a changé depuis l'établissement d'une LCR complète référencée. Elle permet d'indiquer les certificats révoqués avant la publication de la prochaine LCR. Le delta de LCR est utilisé dans des infrastructures à haut niveau de sécurité.

Domaine de confiance

Un domaine de confiance regroupe les AC reconnues de confiance par les applications. .

.

CRL indirecte

CRL qui n'a pas été signée par la même AC que celle qui a généré le certificat. L'autorité de certification qui signe les certificats délègue la signature des CRL à une autre autorité de certification.

REV

Bloc fonctionnel dont le périmètre est la vérification de statut du certificat.

SVC

Service de validation de certificats. Le SVC est l'implémentation technique du concept d'Autorité de Validation.

PV

Une politique de validation définit les conditions de validation d'un certificat utilisateur dans un contexte donné, cette restriction d'utilisation vient en complément des conditions générales décrites dans le domaine de confiance.

Pilote

Composant front-office de pilotage et d'orientation de l'appel aux différents services en fonction de la PV appliquée.

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.

VALREV

Composant front-office du service de 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 (CAC). 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.

TFERR

Service de traitement fonctionnel des erreurs. Il permet de trancher les cas d'indécision de VALREV en fonction de la politique de validation appliquée.

SIGREP

Service de signature de la réponse fournie par le répondeur SVC aux applications clientes.

PRECAL

Composant « back-office » du service de vérification des chemins de certification (CAC). Il permet le calcul des chemins de certification et la tenue à jour d'une base de données de référence contenant l'ensemble des chemins précalculés.

CRL2DB

Composant « back-office » du service de vérification de statut d'un certificat (REV). Il permet la tenue à jour d'une base de données de référence contenant l'ensemble des certificats révoqués.

CRLFinder

Au sein de CRL2DB, composant en charge de la récupération des CRL auprès des AC.

CRLProcess

Au sein de CRL2DB, composant en charge du pilotage de CRLFinder et de l'alimentation de la base de données.

2 Présentation du document

2.1 Objet du document

Le présent document constitue le document de conception de référence pour la réalisation d'un service de validation de certificat (SVC). Il a pour but de fournir un document à la fois générique pour les personnes souhaitant connaître le projet SVC et de proposer un document de référence pour les chefs de projet et les développeurs.

Une première présentation générale de l'ensemble du service de validation est réalisée pour permettre une introduction à la notion d'Autorité de Validation. Une présentation détaillée permet de se familiariser avec les différents éléments constituants le service de validation.

Une deuxième partie détaille les spécifications fonctionnelles qui décrivent le fonctionnement interne de chaque service qui s'ancre au service de validation.

La dernière partie définit les spécifications techniques du service de validation et les services associés. Elle permet aux développeurs d'obtenir l'ensemble des points constituant le service de Validation, pour réaliser le développement de la plate-forme.

Le document fait référence à 2 niveaux de fonctionnement :

- Le niveau 1 qui est un premier niveau de fonctionnalité. Il a été développé lors de la première phase du chantier SVC. Des référents au niveau 1 seront parfois utilisées dans ce document quand il sera fait référence au service de révocation (Spécifications fonctionnelles sur le niveau 1).
- Le niveau 2 reprend le concept d'Autorité de Validation avec l'apport des politiques de validation, l'administration et le référentiel centralisé. Il reprend un ensemble d'entités du niveau 1 et des nouvelles fonctionnalités apportées par le niveau 2.

2.2 Document connexe

Titre	Nom du document	Date
Spécifications fonctionnelles niveau 1	sf-validation-v1.1.b.doc	09/12/2003

3 Objectif de l'Autorité de Validation

3.1 Rôle

La validation de certificat est complexe. Si l'utilisation de certificat doit être largement déployée dans une variété d'applications et d'environnements, la quantité de traitement que doit effectuer une application avant que celle-ci n'accepte le certificat doit être réduite. Il y a une variété d'applications qui peuvent se servir des certificats, mais ces applications n'ont pas les capacités de pouvoir appliquer (de manière uniforme et exhaustive) tous les contrôles sur les certificats. Pour pouvoir réaliser les différents contrôles sur les certificats, les applications vont pouvoir faire appel à un service de validation de certificat.

Le SVC fournit un service de validation en temps réel pour les certificats numériques de manière sécurisée et transparente aux différents services utilisateurs. Il encadre cette validation d'une politique qui régit les différents processus qui valideront la requête du demandeur.

3.2 Objectif

Dans le cadre de la sécurisation des échanges, certains services utilisateurs doivent mettre en œuvre des procédures de contrôle sur les certificats des usagers qui accèdent à leurs applications.

Le service de validation regroupe l'ensemble des contrôles que l'on doit appliquer sur les certificats et porte la responsabilité des réponses apportées par le service de validation. Le service de validation fourni doit être générique pour s'intégrer aux différents systèmes informatiques existants et porter sur des contrôles reconnus et validés par l'ensemble des services utilisateurs.

Les services assurés par le SVC sont regroupés autour de plusieurs aspects fonctionnels et techniques, les grands principes sont les suivants :

- Application des politiques de validation selon le domaine de confiance
- Vérification du format des certificats et de leur contenu
- Contrôle du statut du certificat
- Validation de la chaîne de certification
- Encadrement du format des requêtes et réponses en relation avec les standards

Le service de Validation intègre la notion d'autorité. À ce titre, il est responsable de l'ensemble du processus de validation et de la validité des réponses émises. Un service de validation doit définir des **politiques de validation** qui vont établir l'ensemble des règles de vérification, de validation et de confidentialité des données appartenant à un service de validation.

L'Autorité de Validation peut définir plusieurs politiques de validation en fonction du domaine de confiance et de l'usage du certificat.

Lors d'un appel client le service de validation identifie la requête par la politique de validation et le domaine de confiance. Par la suite, il applique des contrôles sur le certificat en fonction de ceux défini dans la politique de validation.

Le service de validation en complément des opérations sur les certificats clients, récupère périodiquement les LCR nécessaires à l'exécution des processus de contrôle et en particulier les prises de décision dans les réponses.

Pour les prestations techniques, l'Autorité de validation s'appuie sur un service de validation, dont elle approuve et audite les moyens et procédures.

Le service de validation peut être vu sous deux angles différents, d'un côté il offre des services à des utilisateurs en respectant des politiques de validation, de l'autre il effectue un travail de récupération, de traitement et de référencement d'AC et de LCR.

Le service de validation a été conçu selon un concept générique et modulaire pour fournir l'ensemble de ses services et pouvoir les étendre selon les besoins.

3 notions ont été définies dans le projet du SVC, qui faciliteront la lecture et la compréhension du document :

- **Service de validation**

Il s'agit de la solution logicielle de l'Autorité de Validation. Il met en œuvre les besoins identifiés auprès des différents services qui feront appel au SVC. Il permet aussi de fournir un cadre d'intégration pour les modules qui s'ancreront au service de validation : la solution applicative est une plate-forme autonome (soumise à des contraintes fortes de sécurité). Dans son cadre d'utilisation, l'Autorité de Validation en tant que service a un engagement à rendre aux utilisateurs.

Le service de validation est un système conçu pour recevoir des composants additionnels qui étend son champ de fonctionnalités.

- **Bloc fonctionnel**

Le bloc fonctionnel regroupe un ensemble de composants qui fournit un service précis au sein du service de validation. Le bloc fonctionnel décrit de manière synthétique et pertinente un service spécifique rendu par le service de validation.

Le service de validation de niveau 1 n'était constitué que du bloc REV. Le service de validation de niveau 2 étend ses fonctionnalités par l'apport de nouveaux blocs fonctionnels :

- CER : vérification de validité des formats de certificat
- CAC : vérification et calcul du chemin de certification

En outre, il fournit une mutualisation de l'accès au service de validation et offre une modularisation de certains composants qui était anciennement monolithique.

Le service de validation offrira un cadre commun d'intégration pour permettre à d'autres composants de s'ancrer sur ce service. Les composants seront décrits précisément dans le chapitre « description de solution fonctionnelle ».

- **Composants**

La plupart des blocs fonctionnels fournissent deux niveaux d'accès, le front Office et le back Office.
Le composant est un constituant élémentaire qui remplit une fonction précise dans un bloc fonctionnel. Il fournit une vue modulaire et technique du bloc fonctionnel. La plupart des composants ont une approche front Office ou back Office.

3.3 Client du système

Le service de validation est destiné aux applications Copernic devant mettre en œuvre la validation des certificats sur leur plate-forme. Les clients aujourd'hui identifiés sont ALP, OPALE, ADP. SSO (ICHAIN)

3.4 Périmètre

Cette section précise les aspects non couverts par le service de validation. Ces éléments sont les suivants :

Authentification

Le service de Validation n'a pas pour fonction de fournir un service d'authentification. Le service de validation établissant son dialogue avec le client aux travers d'un protocole d'échange OCSP ou propriétaire, le processus d'authentification ne peut être réalisé. L'Autorité de Validation de par sa structure de conception et de développement ne doit pas orienter ce service vers ces fonctionnalités.

En théorie, il est possible, grâce aux champs extensions du certificat X.509 v3, d'intégrer dans les certificats des informations concernant le profil du porteur du certificat (fonction dans l'entreprise, vis-à-vis de telle ou telle application, pouvoir en terme de passage de commande ou de transaction…). En interprétant ces informations, les applications ont la possibilité d'accorder ou non le droit d'accéder aux données et aux transactions associées.

En pratique, il est pourtant peu recommandé d'utiliser ces méthodes pour gérer les habilitations applicatives. En effet, le profil applicatif d'une personne est appelé à évoluer fréquemment (changement de fonction, accès à une nouvelle application…), ce qui impliquerait :

- soit révoquer et recréer fréquemment des certificats pour chaque agent et de les mettre à disposition sur un serveur de gestion des habilitations centralisé ;
- soit diffuser à chaque personne, en plus de son certificat d'identité, des certificats attributs pour chaque application ou pour chaque groupe d'agent.

Dans les deux cas, les opérations d'enregistrement seraient multipliées et la gestion des certificats deviendrait très contraignante pour les utilisateurs.

Les certificats sont donc aujourd'hui essentiellement utilisables pour assurer l'authentification des utilisateurs c'est-à-dire pour garantir l'identité d'une personne qui se connecte à une application. La gestion des droits applicatifs et des fonctions de contrôle d'accès reste du ressort des annuaires LDAP.

Archivage

L'archivage des accès sur le service de validation est initialement mis en place pour des contraintes spécifiques au service de validation. Il doit fournir une aide aux exploitants en cas de défaillance du système. Il n'a pas pour fonction de fournir une base d'archivage pour la consultation en ligne des demandes d'accès dans le cadre de besoin métier d'autres services.

Horodatage

Le service de validation peut faire appel à un service d'horodatage en cas de nécessité d'augmenter la sécurité des informations échangées. Il n'a pas été défini de besoin précis d'horodater des données du SVC. Il ne sera pas fait référence dans ce document à des éléments ayant trait à l'horodatage.

4 Description de la solution fonctionnelle

Cette section présente :

- Le service de validation et l'architecture générale ainsi que les blocs fonctionnels du système.
- La cinématique de fonctionnement du service de validation en relation avec les clients et les composants additionnels.
- Une description détaillée de la solution.

4.1 Autorité de Validation

Le service de validation est le cœur du système. Il est composé d'une unité centrale et d'un ensemble de blocs fonctionnels qui s'ancrent à cette unité. Un bloc fonctionnel doit respecter un certain nombre de contraintes pour pouvoir être incorporé au service de Validation. Les contraintes sont organisationnelles, techniques et procédurales.

Le service de validation est la partie centrale qui valide les domaines de confiance en relation avec la ou les politiques de validation.

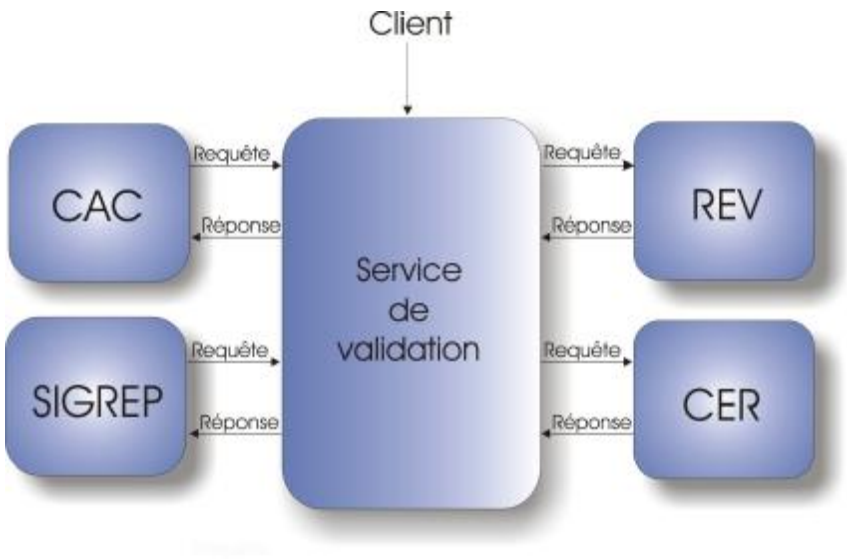


Schéma de principe du service de validation

4.2 Service de validation

Le service de validation reçoit les requêtes provenant des clients, les demandes sont étudiées et relayées auprès des composants pouvant répondre au besoin.

Le service de validation peut selon la réponse du ou des composants influer dans le processus de réponse selon sa politique de validation et selon certaines contraintes de fonctionnement inhérentes au système.

Si un composant ne peut satisfaire à une demande ou ne répondre que de manière partielle à une requête client, le service de validation doit décider de la réponse à adresser au client final, en relation avec ses politiques de validation.

Le service de validation permet via une interface d'administration d'appliquer des décisions sur le mode de fonctionnement. Les administrateurs sont prévenus des problèmes survenus sur le service via une interface de supervision, qui renseigne par des alarmes les anomalies constatées sur le service de Validation.

Voici quelques exemples de fonctions gérées par le service de Validation :

- La responsabilité de la tenue d'une base de LCR à jour et de la fourniture d'une réponse fiable malgré les mécanismes de cache, de duplication et de réplication qui pourront se mettre en place
- L'archivage sécurisé des PV (technique et juridique avec un simple estampillage)
- La déclaration des points de confiance et des terminaisons dans les hiérarchies de certificat.
- La Définition des chaînes de confiance
- Une exploitation avec un cahier de consignes strictes pour pouvoir réagir aux différents cas de dysfonctionnement des composants
- Un chemin de prise de décision parfaitement défini

4.3 Bloc fonctionnel

Pour pouvoir fournir un panel de service, le service de validation est conçu de manière à pouvoir recevoir des composants lui permettant d'élargir son champ de fonctionnalités.

Le service étend ses fonctions de validation au-delà du contrôle du statut d'un certificat, en fournissant d'autres points de contrôle sur les certificats. Pour cela, il fournit un cadre technique et organisationnel pour accepter l'intégration de composants en son sein.

Pour ce faire, le service de validation a les capacités de relayer des demandes clients vers des composants permettant de fournir un cadre élargi de réponse. Il peut aussi prendre une décision en cas de défaillance d'un des composants du service de validation.

Chaque fonction au sens large du terme est définie comme un « Bloc fonctionnel », assurant une fonction précise. Le bloc est constitué généralement d'un partie frontale (notée dans le document front-office) et d'un partie arrière (notée back-office dans le document)

4.3.1 Le Pilote

Ce composant permet de mutualiser les demandes selon le type de client sur un seul point d'accès unique et d'orienter le processus de validation selon les politiques de validation.

4.3.2 Service de vérification des certificats (CER)

CER permet de vérifier le format des certificats. Il vérifie les champs obligatoires selon la norme X.509v3. Il effectue cette vérification en relation avec la politique de validation et son contenu. Les vérifications de base sont les suivantes :

- Vérification de la date de validité du certificat,

- Vérification de l'intégrité du certificat,
- Vérification du format du certificat (format X509v3),
- Vérification de la signature (en relation avec les ACs référencées dans le SVC),
- Vérification des extensions sur l'utilisation des clefs (KEyUsage et ExtKeyUSage).

4.3.3 Service de validation de révocation (REV)

Ce bloc réalise les fonctions suivantes :

- Récupération périodique des Listes des Certificats Révoqués (LCR) auprès des Autorités de Certification et vérification de ces listes,
- Sauvegarde locale des LCR rapatriées dans un dépôt temporaire,
- Traitement des listes de certificats révoqués pour les référencer dans une base de données,
- Traitement des demandes de statut de certificats provenant de ICHAIN, OPALE, ADP et autre et retourner au demandeur le statut du certificat.

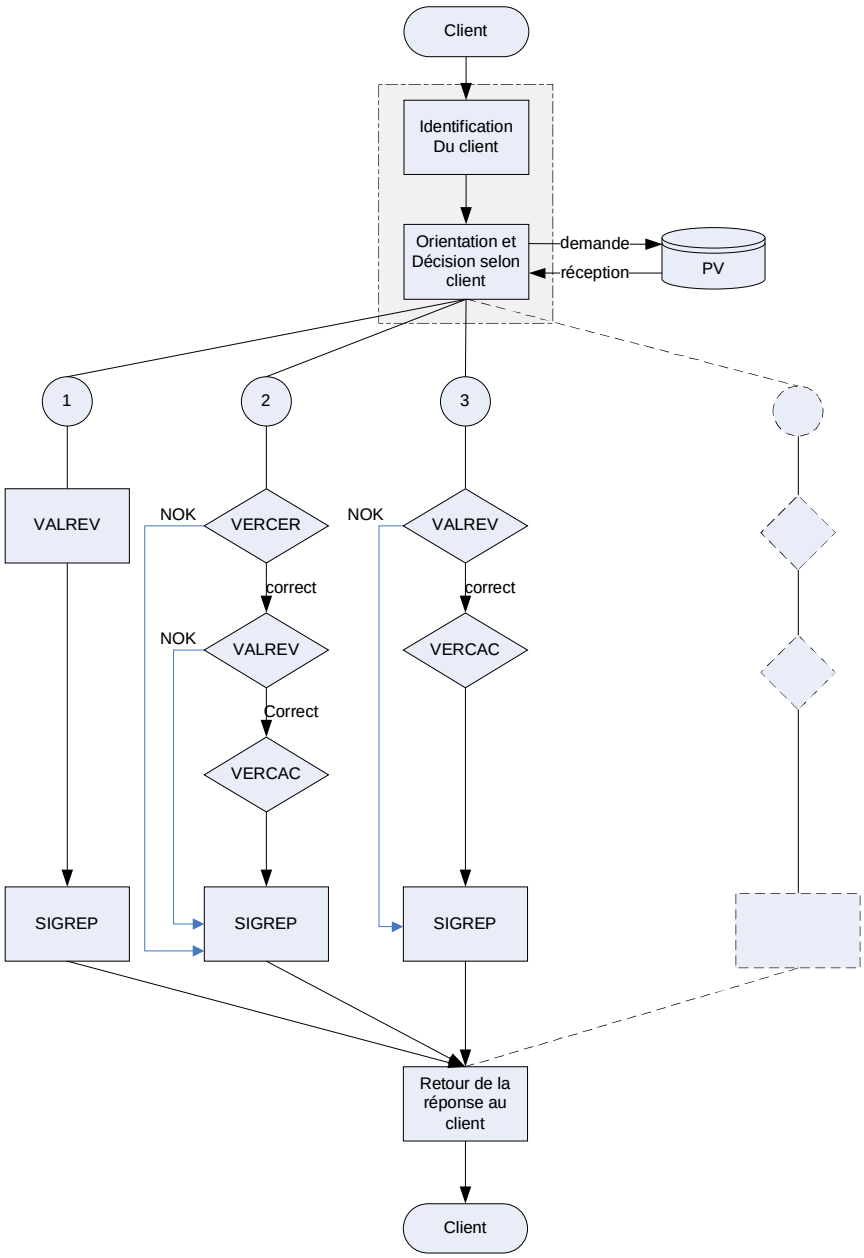
4.3.4 Service de vérification du chemin de certification des AC (CAC)

Le composant CAC offre les fonctionnalités suivantes:

- La validation du chemin de certification des certificats clients et des ACs,
- La construction du chemin de certification des différentes AC et la validation les différentes AC présentes dans le chemin de certification.

5 Cinématique

La cinématique ci-dessous présente un ensemble de scénarios qui résulte de l'utilisation du SVC de niveau 2. Les scénarios indiqués à la suite de cette section sont des trames d'exemples qui découlent de l'utilisation des clients ICHAIN et JAVA et des composants de validation intégrés au SVC de niveau 2. Il est envisageable de générer de nouvelles contextures selon les besoins de validation ou d'intégration de nouveaux types de clients.



Ce schéma représente des exemples de scénarios d'utilisation du SVC version 2 au niveau frontal. (Front Office)

L'orientation du processus de validation vers telle ou telle trame de validation est en relation avec le type de client et les identifiants qui sont positionnés dans la requête. Le Pilote est le composant en charge de l'identification et l'adressage de la requête vers les différents processus de validation. Le SVC met à disposition quatre composants front office indépendants concourant à la validation (VERCER, VALREV VERCAC et SIGREP) La politique de validation demandée par l'appelant précise l'ensemble des contrôles à effectuer en relation avec le domaine de confiance de l'appelant.

A titre d'exemple, voici la cinématique pressentie pour trois applications identifiées aujourd'hui comme clientes du SVC.

Les différents scénarios sont définis à partir des fonctions de validation mise à disposition par le SVC et des besoins de validation demandés par les services faisant appel au SVC.

1. Le premier scénario réalise la validation du statut de révocation d'un certificat final ; c'est un appel OCSP défini conformément au RFC2560.

2. Le second scénario est destiné au client JAVA qui recourt à un protocole propriétaire pour la requête et la réponse. L'utilisation de ce protocole permet d'étendre les capacités de vérification à effectuer sur le certificat. Ces fonctionnalités sont envisageables par l'apport du certificat client dans la requête (fonctionnalité inexistante dans OCSP) et le contexte d'appel. Ce scénario permet ainsi d'approuver le contenu et l'aspect du certificat, d'appliquer une vérification du statut de révocation et de valider le chemin de certification qui fait autorité pour ce certificat. Les contrôles spécifiques à réaliser sont définis par la politique de validation. Dans la suite du document, ce service est désigné sous le nom de « service enrichi »

3. Le troisième scénario accomplit une validation du chemin de certification en supplément de la vérification du statut de révocation. Ce service est accessible à travers un appel OCSP standard. La norme OCSP n'ayant pas prévu d'adresser le certificat client au serveur, la validation du chemin de certification se basera sur le DN de l'AC positionné dans la requête. Pour réaliser la complétude d'identification du client, la requête contient le domaine de confiance de l'appelant. Le domaine de confiance sera positionné dans l'attribut « *RequestorName* » de la requête OCSP. Dans la suite du document, ce service est désigné sous le nom de « service OCSP étendu »

Exemple de processus de validation

1. Un client de type JAVA initie une requête auprès du service de validation
2. Le service de validation en utilisant le composant pilote identifie le format de la requête et récupère le contexte d'appel (domaine de confiance et identifiant de la politique de validation)

Remarque : Quels que soient le type de client et le contexte d'appel, le « pilote » oriente les paramètres de validation vers les différents composants du SVC.

3. Le pilote invoque la politique de validation correspondante à l'identifiant de PV transmis dans la requête.
   Remarque : Si la PV est absente de la requête, il applique la PV active pour ce domaine de confiance. Si la PV a été remplacée par une PV plus récente par un administrateur, c'est la PV active pour ce domaine qui est utilisée et non plus celle positionné en argument de la requête.

4. Le pilote lit la PV pour prendre connaissance des différents composants auxquels il va faire appel pour réaliser l'opération de validation.
5. Le pilote adresse au composant VERCER le certificat client et les informations spécifiques qui lui sont dédiés dans la PV. Les informations de la PV permettront d'appliquer des tests spécifiques à ce certificat.
6. Le composant VERCER adresse au pilote le résultat de la validation. Le pilote applique une décision selon le contenu transmis par VERCER :
    a. Si la validation est négative, le pilote génère le format de réponse et l'adresse au composant SIGREP pour signature si nécessaire. La réponse est ensuite adressée au client.
    b. Si la validation est positive, le pilote transmet le certificat au composant suivant.
7. Le composant VALREV vérifie le statut du certificat
8. Le composant VALREV adresse au pilote le résultat de la validation. Le pilote applique une décision selon le contenu transmis par VALREV
    a. Si la validation est négative, le pilote génère le format de réponse et l'adresse au composant SIGREP pour signature si nécessaire. La réponse est ensuite adressée au client.
    b. Si la validation est positive, le pilote transmet le certificat au composant suivant.
9. Le composant VERCAC vérifie le chemin de certification du certificat présenté par le pilote.
    a. Si la validation est négative, le pilote génère le format de réponse et l'adresse au composant SIGREP pour signature si nécessaire. La réponse est ensuite adressée au client.
    b. Si la validation est positive, le pilote s'adresse au composant SIGREP si nécessaire pour signature. La réponse est ensuite adressée au client

6 Description fonctionnelle détaillée

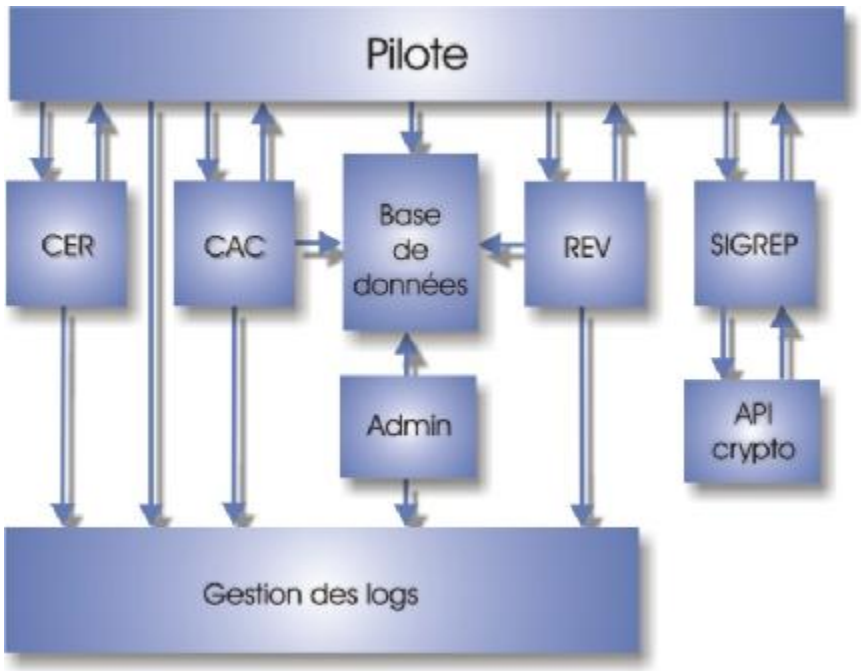


fig 1 : Les différents modules qui composent le service de validation

6.1 Les composants

Le fonctionnement du SVC repose sur deux notions qui sont le **domaine de confiance** et la **politique de validation** (cf. paragraphes suivants).

Le service de validation est composé des composants suivants :

- CER,
- REV,
- CAC,
- SIGREP.

Les différents composants sont étudiés dans les sections suivantes.

6.2 Domaine de confiance

Un domaine de confiance regroupe les AC reconnues de confiance par les applications. La notion de domaine de confiance permet, pour un même domaine, de regrouper plusieurs AC selon un modèle hiérarchique. Le domaine de confiance apport une variante du modèle hiérarchique à racine unique

par l'introduction de plusieurs racines de confiance dans une domaine, ce modèle garde la simplicité de la validation hiérarchique, mais rend équivalents les AC regroupés dans une même liste

Le domaine de confiance en relation avec la politique de validation définit les conditions générales d'appel du SVC pour un client appartenant à ce domaine.

En association avec la politique de validation, l'utilisation du domaine de confiance permet en outre de réduire la combinatoire d'appel du SVC.

Le domaine de confiance est une valeur qui est positionnée en argument dans la requête adressée par le client au SVC.

6.3 Politique de validation

Une politique de validation définit les conditions de validation d'un certificat utilisateur dans un contexte donné, cette restriction d'utilisation vient en complément des conditions générales décrites dans le domaine de confiance.

Une politique de validation est composée de la manière suivante :

- Identifiant de la PV, version, statut,
  - § L'identifiant de la PV est identifié sous la forme d'un OID référencé au sein de la DGI,
  - § Le statut correspond à l'état de la PV : valide, active ou refusée.
- La liste des composants qui doivent être appelés lors de la phase de validation,
- La liste des contrôles spécifiques à effectuer par composants.

La composition d'une politique de validation est réalisée par l'exploitant du domaine concerné en relation avec l'administrateur du SVC, ce dernier valide ou refuse la PV définie par 'exploitant.

La politique de validation est positionnée en argument dans la requête adressée par le client au SVC. Le service de validation peut utiliser une PV différente de celle indiquée dans la requête du client, dans le cas ou une nouvelle PV pour ce domaine de confiance est connu du SVC et ne le serait pas du client. La PV qui est utilisée pour le processus de validation est indiquée dans la réponse.

Le contenu de la politique de validation est détaillé dans les spécifications techniques détaillées.

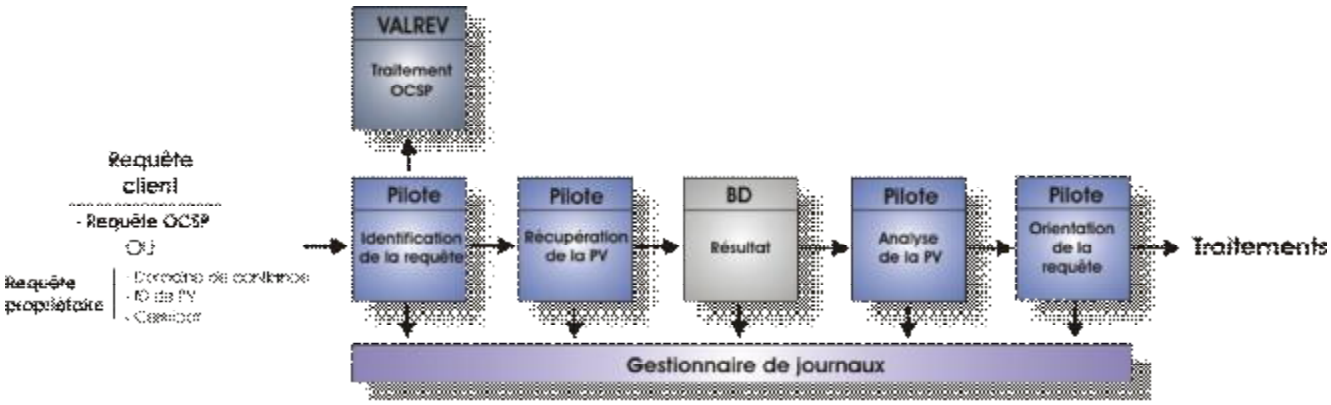
6.4 Le Pilote

Le pilote est le composant qui permet d'identifier le type de demande adressé au SVC et de diriger les requêtes vers les bons composants du SVC.

La partie frontale du pilote reçoit les requêtes des différents clients qui font appel au SVC. La requête peut-être de type :

- OCSP, elle doit respecter le standard RFC2560,
- Requête « étendue », respect du standard RFC2560 en ajoutant l'identifiant du domaine de confiance dans la requête (champ *RequestorName* du RFC2560),
- Requête « enrichie », appel de type propriétaire défini spécifiquement pour les besoins de la DGI.

Le processus d'identification et d'orientation se déroule selon le processus suivant :



Lors de la phase d'identification, le pilote détermine le type de requête adressé par le client. Si la requête est de type OCSP, le pilote adresse la demande directement auprès du composant VALREV (validation de la révocation).

Pour les autres types de requêtes, le pilote récupère l'identifiant de la PV et valide que cette PV peut être utilisée pour ce domaine de confiance.

Si la PV utilisée est acceptée, le pilote la récupère depuis la base de données. La PV est analysée pour connaître les composants qui devront être appelés et les paramètres qui leur seront adressés pour réaliser le processus de validation.

Le Pilote appelle les composants identifiés dans la PV en leur adressant les tests qui devront être effectués sur le certificat client. Il récupère le résultat et décide de passer ou non au composant suivant (cf. section 5 : Cinématique)

6.4.1 Format des requêtes et réponses

Le pilote du SVC accepte 3 types de requêtes :

- Requête OCSP,
- Requête étendue,
- Requête enrichie.

La réponse est adaptée au format de la requête préalablement émise par le client. Cette section présente le contenu et le type de réponse et de requête qui est géré par le SVC.

6.4.1.1 Format OCSP

6.4.1.1.1 Requête

La requête respect le formalise du RFC2560 :

- Version du protocole,
- Service demandé,
- Identifiant du certificat cible (numéro de série, empreinte du DN émetteur, empreinte de la clef publique, identifiant de l'algorithme de l'empreinte),
- Extensions optionnelles qui peuvent être traitées par le SVC,
    - Le champ **signature** sera accepté par le SVC. Il permet au client de signer sa requête de demande de statut. Une requête comportant une signature doit être acceptée par

le répondeur. En revanche, le répondeur ne réalisera pas de vérification de la signature.

- Le champ **nonce** est géré par le SVC. Le nonce lie une requête à une réponse pour se prémunir contre les attaques de type rejeu. Le nonce est inclu dans les extensions de la requête. Il est représenté par une chaîne numérique générée aléatoirement par le client OCSP.

6.4.1.1.2 Réponse

La requête respecte le formalise du RFC2560 :

- Version de la syntaxe de la réponse,
- Nom du répondeur,
- Réponse sur le statut du certificat cible,
- Nonce (si présent dans la requête),
- La date de réponse,
- Identifiant (OID) de l'algorithme de signature,
- Empreinte de la signature,
  - La signature étant obligatoire dans la réponse, le certificat du serveur OCSP et son AC terminale seront adressés dans le champ ***certs.*** Ce champ est optionnel dans le RFC est rendu obligatoire dans le SVC.
- Extensions optionnelles.

  Nocheck est toujours positionné dans la réponse.

6.4.1.1.2.1 Extension

Le champ **id-pkix-ocsp-nocheck** est intégré dans la réponse. Il permet de spécifier au client OCSP de faire confiance au certificat du répondeur OCSP durant sa période de validité. Dans le cas où cette option n'est pas précisée, il incombe à la politique de sécurité du client de décider de vérifier le statut du certificat.

6.4.1.2 Format « étendu »

6.4.1.2.1 Requête

Le format étendu reprend le formalisme d'une requête de type OCSP en utilisant en complément le champ r*equestorName*. Ce champ permet d'identifier le client qui réalise la requête.

Dans le cadre du SVC, le domaine de confiance de l'appelant sera indiqué dans ce champ. L'utilisation de ce champ permet d'étendre les validations à effectuer sur les clients OCSP « classiques ». Pour ce faire, le SVC associe une PV par défaut pour le domaine indiqué dans le champ *requestorName*. L'identifiant positionné dans ce champ sera un OID référençant un domaine de confiance.

6.4.1.2.2 Réponse

Le format de la réponse est le même que pour le format OCSP.

6.4.1.3 Format « enrichi »

6.4.1.3.1 Requête

Le format de la requête est de type XML, il comporte les champs suivants :

- OID du domaine de confiance,
- OID de la politique de validation,
- Le certificat à vérifier,
- Optionnellement : un nonce.

6.4.1.3.2 Réponse

Le format de la requête est de type XML, il comporte les champs suivants :

- Version du SVC,
- Numéro de série du certificat,
- Identifiant de la politique de validation réellement appliquée,
- Statut de la réponse (rejetée ou acceptée),
- Code d'erreur fonctionnelle du rejet,
- Message d'erreur du rejet,
- Code d'erreur technique,
- Message d'erreur technique,
- Date de la réponse,
- Le nonce (si présent dans la requête),
- Liste des composants appelés et les résultats par composants,
    - Révocation
        - Version du protocole OCSP,
        - Nom du répondeur OCSP,
        - Date de production de la réponse,
        - Identifiant du certificat cible,
        - Statut du certificat cible,
        - La date du champ thisupdate de la CRL,
        - La date du champ nextupdate de la CRL,
        - Date de révocation du certificat,
        - Date de récupération par le SVC de la CRL,
    - Certpath
        - Le chemin de certification de certificat cible. Il est fourni sous la forme d'une liste ordonnée allant de l'AC racine jusqu'au certificat terminal. Le format des certificats est le PEM.
- Signature de la réponse,
- Le certificat du répondeur (celui qui a émis la signature) et son AC terminale.

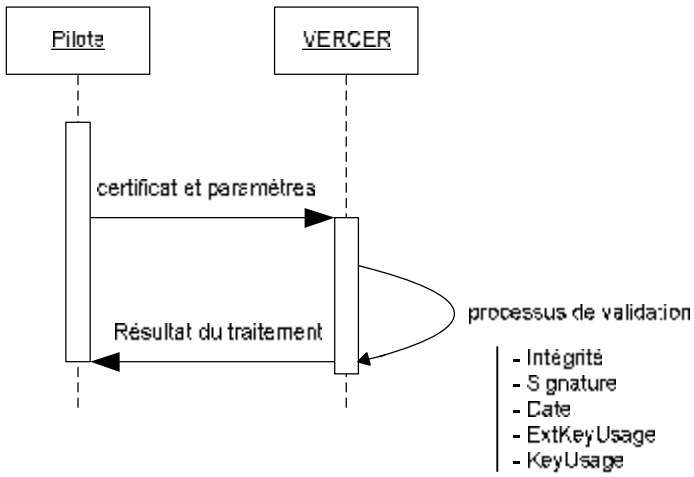
6.5 CER : Bloc fonctionnel de vérification des certificats.

Ce bloc fonctionnel permet de vérifier le contenu d'un certificat X.509v3. Il est appelé par le *pilote* qui lui transmet le certificat et les règles de vérification qui doivent être appliquées sur le certificat. Ce bloc est composé d'un composant logiciel nommé VERCER.

Les règles transmises par le pilote découlent de la politique de validation (PV) définie préalablement par les exploitants. Le pilote ne transmet pas directement la PV au composant VERCER, seulement le contenu spécifique des traitements qui seront effectués par ce composant (le contenu exhaustif des tests est décrit dans la section 7.1).

Le traitement est de type synchrone, le pilote attend la réponse de CER pour continuer le processus de validation.

Le principe d'appel du bloc CER est le suivant :



6.6 REV : Bloc fonctionnel de validation du statut des certificats

Ce bloc permet de connaître le statut d'un certificat en temps réel. Pour ce faire, il est découpé en deux modules pour fournir les services de back-office et de front-office, les modules sont désignés selon la terminologie suivante :

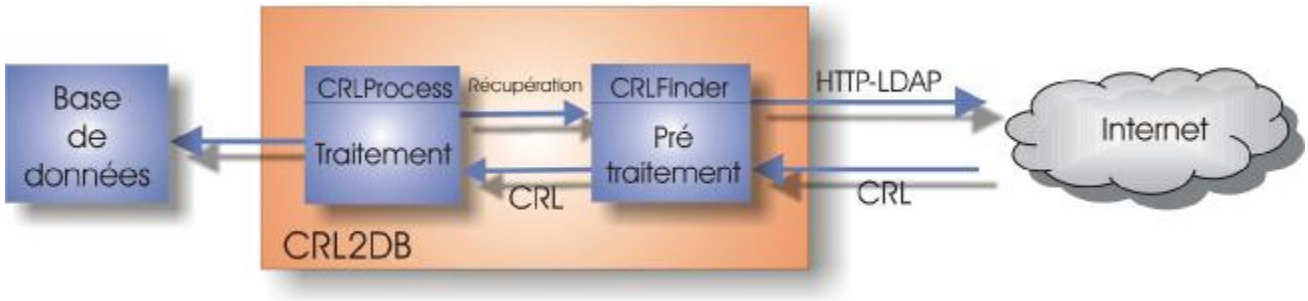
- CRL2DB (back-office),
- VALREV (front-office).

Le module CRL2DB permet de récupérer les CRL auprès des autorités de certification (AC) et d'effectuer des traitements de validation sur celles-ci. Si la vérification des CRL est positive, elle est découpée pour une insertion dans la base de données du SVC. Le traitement appliqué sur les CRL est décrit dans la section 7.2

Le composant VALREV est la partie frontale du service de validation, il fait office de répondeur OCSP. Ce module n'est pas appelé directement par les clients, le pilote lui adresse les requêtes.

6.6.1 CRL2DB

Le composant CRL2DB est découpé en 2 modules :



- Le module CRLFinder permet la récupération des CRL et applique une première série de vérification sur son contenu : vérification de la signature et vérification des dates entre la dernière CRL récupérée et la CRL actuelle.

- Le module CRLProcess récupère les CRL auprès du module CRLFinder. Il effectue un ensemble de traitement sur la CRL (détaillés dans la section 7.2). Si les vérifications sont conformes au résultat attendu, la CRL est découpée pour être insérée dans la base de données.

6.6.2 VALREV

Le module VALREV est la partie frontale du bloc REV, il fait office de répondeur OCSP. Il est constitué de deux services : un répondeur de type OCSP « classique » et un répondeur de type propriétaire.

VALREV OCSP

Il Permet de traiter les requêtes OCSP « classique » en conformité avec les standards (RFC2560).

Les requêtes transmises à VALREV OCSP contiennent le numéro de série du certificat, l'empreinte du DN de l'émetteur du certificat (l'AC), l'empreinte de la clef publique de l'émetteur du certificat et l'identifiant de l'algorithme des empreintes.

Le composant VALREV possède un composant de décision fonctionnelle, TFERR (Traitement des erreurs). Il permet d'appliquer une décision en cas d'indécision de VALREV en fonction de la politique de validation appliquée.

Le principe d'appel du module VALREV est le suivant :

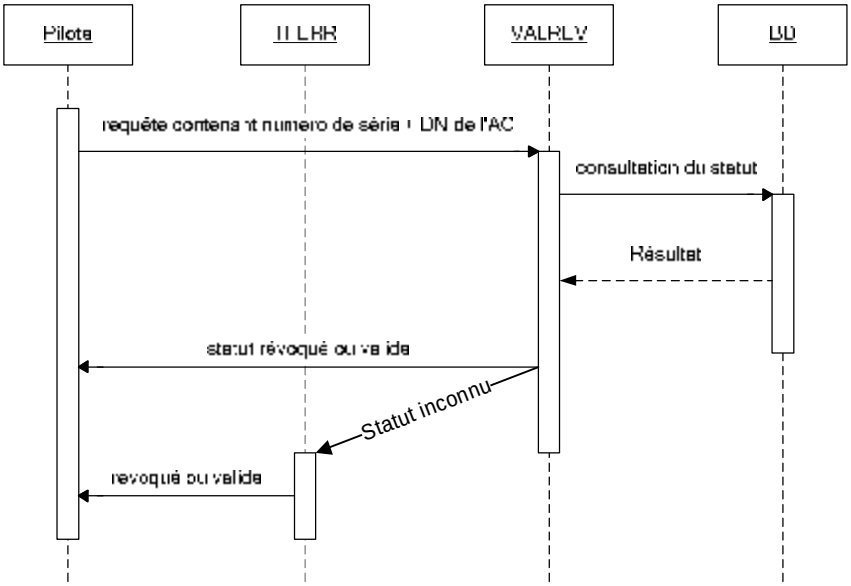


Schéma de fonctionnement de VALREV OCSP

VALREV « enrichie »

À la différence d'un répondeur OCSP « classique », le contenu de la requête adressé par le pilote contient le certificat client.

Le traitement appliqué VALREV est le même que pour une requête OCSP « classique »

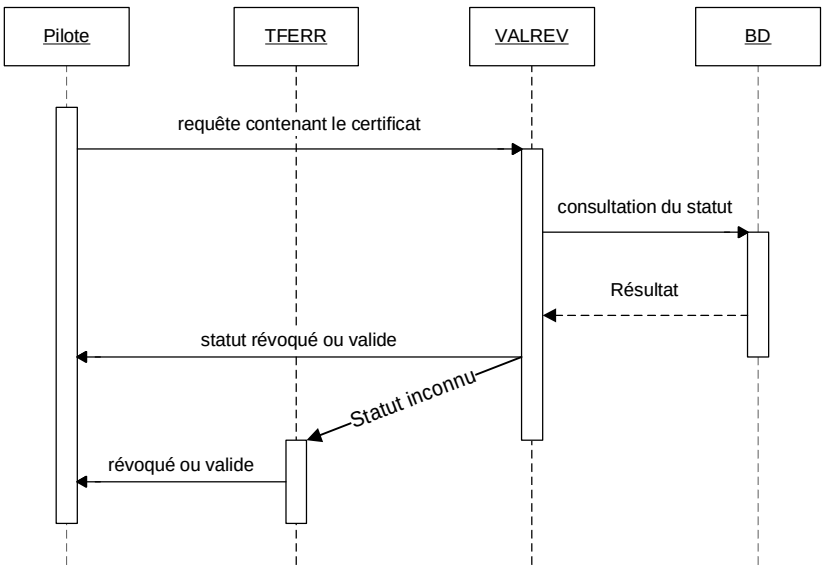


Schéma de fonctionnement de VALREV « enrichie »

6.7 CAC : Bloc fonctionnel de vérification des chemins de certification

Ce composant permet de valider les chemins de certification des certificats d'AC et client. Le bloc CAC divise la tâche de vérification en deux étapes : la première précalcule les chemins de certification, la deuxième étape vérifie que le certificat présenté appartient bien à un chemin prédéfini. Pour ce faire, le bloc fonctionnel *CAC* est constitué de deux modules : VERCAC et PRECAL.

- VERCAC permet de valider le chemin de certification des certificats clients
- PRECAL permet de calculer le chemin de certification des autorités de certification. L'intégration des nouveaux certificats d'AC est réalisée par le biais de l'interface d'administration.

Dans la pratique, la vérification de la conformité par rapport à une politique de certification connue ou l'appartenance à une racine de confiance connue est suffisante pour valider un certificat dans le domaine des applications du MINEFI.

Néanmoins, le service de validation doit implémenter les autres cas en se fondant sur les recommandations de la norme PKIX de l'IETF (*rfc3280.*) En effet, la version X509 v3 a introduit des attributs optionnels qui vont dans le sens de la restriction du domaine d'utilisation d'un certificat, ces attributs peuvent apparaître au niveau des certificats des AC (racine ou non) ou sur les certificats utilisateur. Un algorithme de validation conforme à la norme doit prendre en compte les contraintes supplémentaires suivantes lors de la construction du chemin de certification :

- Contrainte de base : une AC (racine ou non) peut définir un nombre maximal de niveaux valides dans la recherche du chemin de certification. Cette contrainte est destinée à restreindre les transitivités lorsque le chemin traverse les domaines de confiance différents par des accords croisés entre AC ;
- Contrainte sur les noms des AC : lorsqu'elle est présente, cette contrainte s'applique sur les sous branches valides d'une chaîne de certification à partir de la racine. Dans le même ordre d'idée, une contrainte peut interdire le passage par certaines branches exclues du chemin. Cette contrainte est surtout utilisée dans le cas des certifications croisées entre deux AC. Par exemple, lorsqu'une organisation veut restreindre l'accès des certificats externes à un seul sous domaine, cet accès étant régi par le certificat croisé avec l'AC externe ;
- Contrainte sur les équivalences de politique de certification : une AC peut interdire ou restreindre l'acceptation de politiques équivalentes d'un domaine de confiance externe à la sienne. Cette contrainte s'applique surtout dans le cas des certifications croisées entre AC.

Le respect de ces normes garantit l'interopérabilité des certificats d'AC hétérogènes, tout en maintenant le niveau de confiance accordé a priori au service de validation par les différents acteurs de la transaction.

6.7.1 PRECAL

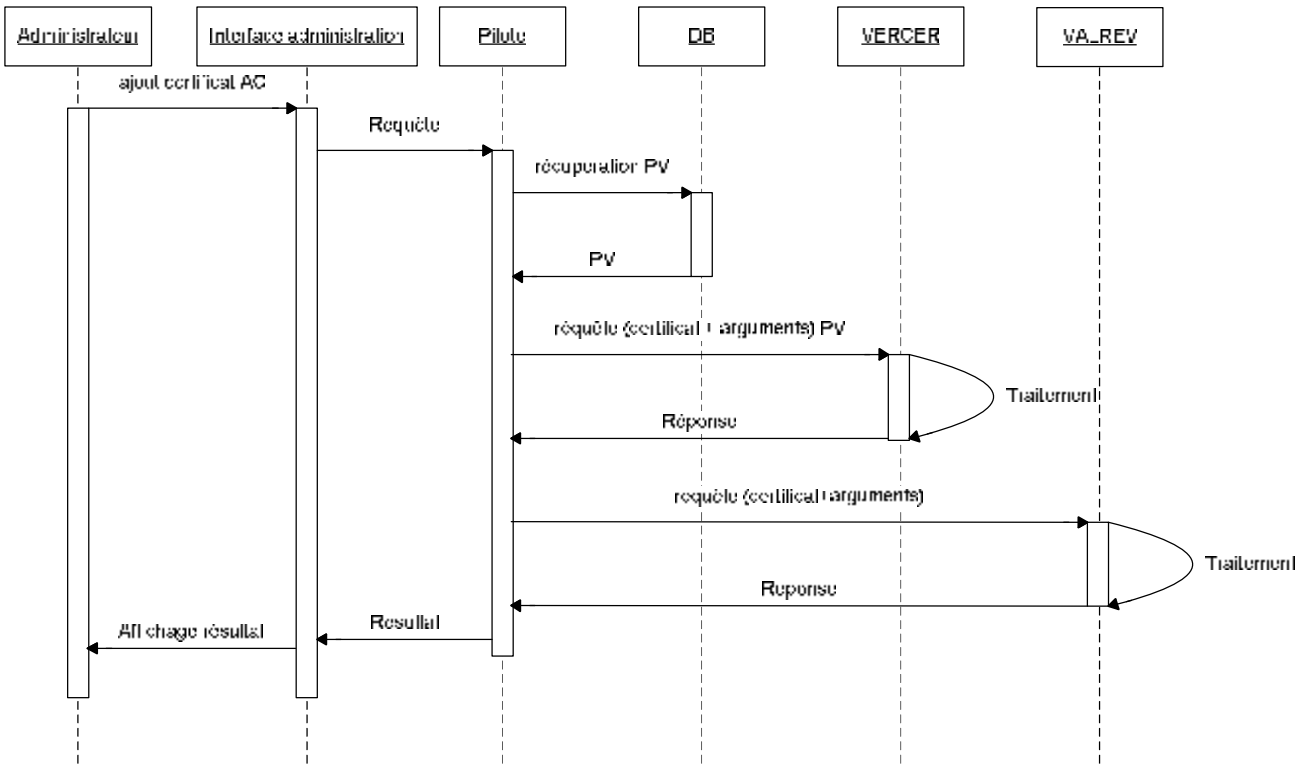
Ce composant permet de calculer le chemin de certification. Le calcul du chemin de certification est pris en charge par le module d'administration lors de l'introduction d'une nouvelle AC dans le système.

L'un des principaux traitements de validation de certificat consiste à construire le chemin de certification pour remonter à la racine de confiance. Cette construction doit parcourir les hiérarchies d'AC en examinant les certificats à chaque étape. La construction des chemins de certification obéit à des règles précisées dans la section 7.3

Le chemin de certification trouvé n'est valide que si les AC appartenant au parcours sont valides, c'est-à-dire qu'elles ne sont pas périmées ni révoquées. Pour accélérer le traitement, le statut de révocation de chaque AC est mis à jour préalablement par un autre processus (VALREV). Si le chemin de certification est valide, le résultat est écrit dans la base de données.

Le chemin calculé d'après les contraintes de la politique de validation est valide pendant toute la durée de vie de l'AC. Mais ce chemin peut changer si l'AC a procédé à des modifications dans sa politique de certification. Dans ce cas, il faut donner la possibilité à l'administrateur de modifier les paramètres de validation, de recalculer le chemin et de mémoriser celui-ci. Le système n'historise pas les chemins précédemment calculés.

Exemple d'Intégration d'une nouvelle AC dans CAC :



Validation d'une nouvelle AC

Les vérifications effectuées sur les AC avant leur validation par l'administrateur sont détaillées dans la section 7.3

6.7.2 VERCAC

Le composant VERCAC permet de valider le chemin de certification d'un certificat donné. Cette opération consiste à rechercher un chemin valide à partir des certificats de confiance présents dans la base, et ceci pour une politique de validation donnée.

Lorsque la politique de validation n'est pas spécifiée par le demandeur du service, la politique par défaut est appliquée (dans notre cas `MINEFI`) pour le domaine de confiance passé en argument.

Le composant analyse le DN de l'émetteur du certificat pour retrouver l'AC terminale.

A partir de l'identifiant de l'AC terminale trouvé dans l'étape précédente, une recherche dans la base de données permet de retrouver tous les AC intermédiaires et l'AC racine. Si ce chemin existe, il faut s'assurer que les AC trouvées ne sont pas révoquées et périmées. Pour chaque AC entrant dans le chemin, la vérification du statut (*revoked*) ou de la péremption (*enddate*) permet de statuer sur la validité de la chaîne de confiance.

Si toutes les AC de la chaîne sont validées, le chemin de certification est validé. Dans le cas contraire, le chemin trouvé n'est pas validé, car une des AC de la chaîne est compromise ou périmée.

Principe de fonctionnement de VERCAC :

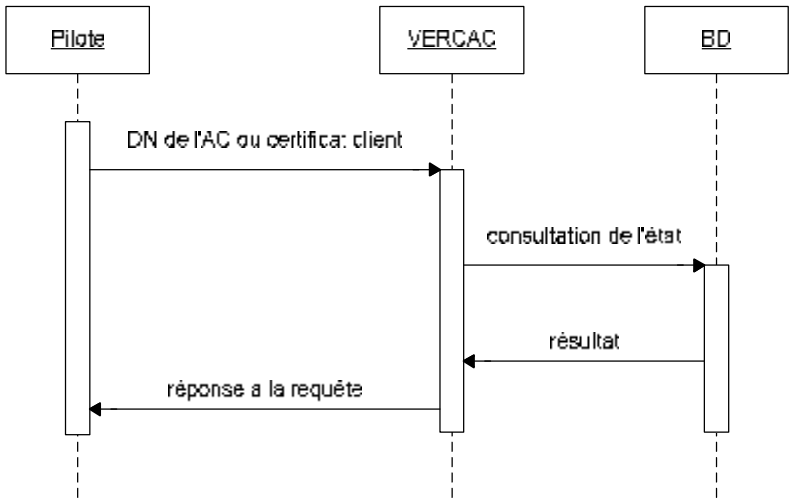


Schéma de fonctionnement de VERCAC

Principe de validation

1. Le pilote adresse au composant VERCAC soit le DN de l'AC du certificat client, soit le certificat client.

   Remarque : Dans le cas d'un appel au SVC par un client de type OCSP, il n'y a pas d'utilisation du composant VERCER. Le pilote adresse alors à VERCAC le DN de l'AC qui est transmis dans la requête client.

   Si VERCER est utilisé lors de la phase de validation, VERCAC reçoit en paramètre le certificat client.
2. VERCAC recherche dans la base les AC référencées par rapport à la PV et au domaine de confiance ;
3. Vérification que le DN de l'AC est connu dans les AC référencées ;
4. Si la vérification est concluante, vérification que l'AC du certificat ou de la requête client est une AC terminale ;
5. Vérification que le chemin de certification pour cette AC terminale est valide (vérification du statut et de la date de péremption des différentes AC de l'itinéraire de certification).

Les causes de rejet sont les suivantes :

- l'AC qui a signé le certificat client n'est pas une AC terminale ;
- une des AC du chemin de certification est révoquée ;
- une des AC du chemin de certification est périmée.

6.8 SIGREP

SIGREP est le composant de signature du SVC. Il réalise l'empreinte des données qui lui sont adressées par le pilote et applique une signature sur l'empreinte.

L'intégration de dispositifs de signature électronique dans le SVC permet de répondre à des besoins de sécurisation que sont l'authentification, la non-répudiation et le contrôle de l'intégrité :

- Authentification : cette propriété permet de connaître sans ambiguïté l'émetteur de la transaction. La personne possédant la clé privée est bien à l'origine de la transaction.
- Non-répudiation : elle permet d'affirmer que le possesseur de la partie privée de l'élément cryptographique est bien celui identifié par le certificat.
- Intégrité : elle permet de contrôler le contenu de l'information reçue, de le comparer à ce qu'il était avant son envoi et ainsi de détecter une éventuelle modification lors du transport.

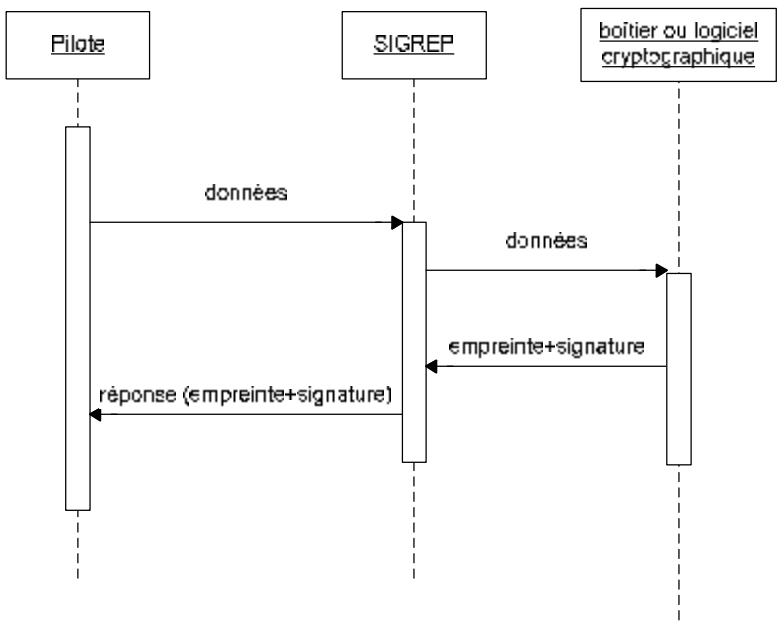


Schéma de fonctionnement du processus de signature

La signature électronique fait appel à des opérations cryptographiques complexes (génération de l'empreinte des données, signature à l'aide de la clé privée, algorithmes mathématiques…). Elle est

consommatrice de ressources et nécessite des processus sécurisés pour protéger la clef privée et le processus de signature.

L'utilisation d'une carte cryptographique permettrait de protéger la clef privée du SVC et diminuer le temps de traitement lors du processus de signature électronique.

L'intégration d'une carte cryptographique est un élément secondaire dans le processus de signature, il pourra être fait appel éventuellement à une solution cryptographique logicielle comme OpenSSL.

Cependant, l'utilisation d'une carte cryptographique n'a pas été retenue pour le SVC de niveau 2 lors de sa première implémentation. La carte initialement prévue ne permettait pas de gérer les serveurs composés de biprocesseurs.

7 Description des vérifications

Les différents modules du SVC et leur fonction de base ont été présentées dans les sections précédentes. Chaque module effectue un lot de vérifications qui recouvre de nombreuses opérations, présentés dans le même ordre que celui qui est utilisé par le SVC lors du processus de validation.

On notera d'ores et déjà qu'un certificat est un document électronique qui est lui-même signé par une autorité de certification qui respecte la norme X.509v3, et que l'ensemble des opérations qui sont décrites pour un certificat client sont également vrais pour tous les certificats des chemins de certification.

7.1 VERCER : Vérification des certificats

L'intégrité du certificat est garantie lors de l'opération de vérification de la signature du certificat avec la clef publique de l'autorité qui l'a émis. Si la signature du certificat est correctement vérifiée, cela signifie d'une part qu'il a bien été émis par l'autorité de certification indiquée et d'autre part qu'aucune des données qu'il contient n'a été modifiée depuis son émission.

Une première vérification doit permettre de valider que le certificat contient les champs de base du protocole X.509 en version 3 :

- Le champ **version**, qui contient un nombre entier servant à indiquer la version du certificat :
    - La valeur 0 désigne un certificat v1,
    - La valeur 1 désigne un certificat v2,
    - La valeur 2 désigne un certificat v3 ;

  Seule la version 3 sera acceptée par le processus de validation du SVC.

- Le champ **serialNumber** (numéro de série) contient un nombre entier servant à désigner le numéro de série du certificat. Le contrôle sera effectué sur la présence du champ. Il sera ensuite utilisé par le composant de vérification du statut en relation avec l'AC qui a émis le certificat.

- Le champ **Signature Algorithm** (description de la signature) identifie le type de l'empreinte et l'algorithme de signature utilisé pour signer le certificat. Le contrôle sera effectué sur la présence du champ, le contenu sera utilisé pour valider la signature.

- Le champ **Issuer** (émetteur) est le nom distinctif X.500 de l'AC qui émit le certificat. Le contrôle sera effectué sur la présence du champ, le contenu permettra de valider le signataire.

- Le champ **subjectPublicKeyInfo** comprend un identificateur d'algorithme et la valeur de la clé publique du détenteur. Le contrôle sera effectué sur la présence du champ.

- Le champ **subject** (sujet) spécifie le DN (norme X.500) du propriétaire du certificat. Le contrôle sera effectué sur la présence du champ.

- Le champ **Validity period** (période de validité) Il s'agit de vérifier qu'au moment de l'utilisation de la clef privée correspondant au certificat, ce dernier était bien dans sa période de validité. Pour cela, il est nécessaire de vérifier que le moment d'utilisation est antérieur à la date qui figure dans le champ « valide à partir de « (valeur *notBefore* du champ *validity*) et postérieur à la date qui figure dans le champ « valide jusqu'au » (valeur *notAfter* du champ *validity*).

- Le champ **signatureValue** (signature de l'AC) correspond à la signature de l'Autorité de Certification qui a émis le certificat. Cette signature est effectuée en passant l'ensemble du certificat au travers d'une fonction de hachage, puis en chiffrant le résultat à l'aide de la clé privée de l'AC. Ce champ doit être vérifié par récupération du certificat de l'AC qui a émis le certificat en question.

- **Extensions** :

  Les certificats X.509 version 3 offrent un mécanisme pour permettre aux AC d'ajouter des informations supplémentaires concernant la clé publique du détenteur, la clé publique de l'émetteur, les LCR de l'émetteur, ou pour imposer des contrôles de gestion. Ces champs sont définis pour le standard X.509v3. Le SVC n'est pas tenu de valider toutes les extensions du certificat. Les sections subséquentes décrivent les extensions qui seront étudiées par le composant de vérification des certificats.

  Un champ d'extension est marqué comme critique ou non critique. Dans les deux cas le composant de vérification des certificats ne prendra pas en compte cette marque, il pourra traiter le reste du certificat en ne tenant pas en compte ce champ.

  Le champ **KeyUsage** (utilisation de la clef) renseigne sur l'utilisation qui doit être faite de la clé. Selon la politique de validation utilisée, la valeur de *KeyUsage* doit être validée spécifiquement. Par exemple, dans le cadre de l'ADP, la *KeyUsage* doit contenir : digital signature. Le test de validation de ce champ est défini lors de la création de la PV.

  Initialement, l'utilisation des champs *KeyUsage* est restreinte selon l'algorithme cryptographique utilisé pour générer la clef publique du certificat. Cependant, en relation avec l'extension *ExtKeyusage* le composant de VERCER devra vérifier le contenu du champ KeyUsage par rapport à l'utilisation faite du certificat. La vérification ne sera pas réalisée directement au niveau de l'extension *KeyUsage,* mais en complément de *ExtKeyUsage.*

  Type d'utilisation de clef

  - digitalsignature (0)
    - Ce bit est positionné lorsque la clé du sujet est utilisée pour la signature en tant que mécanisme. Autrement dit le chiffrement d'une empreinte avec la clef privée.
  - nonRepudiation (1)
    - similaire à *digitalsignature* mais en tant que service (autrement dit la volonté d'ajouter un service de non répudiation à une signature). Pour du S/MIME, bit positionné avec *digitalSignature* typiquement.

  - keyEncipherment (2)

  - chiffrement de clé (clé secrète d'algorithme symétrique généralement, ou clé RSA éphémère en SSL/TLS)
- dataEncipherment (3)
  - Utilisé lorsque la clé publique est utilisée pour le chiffrement de données utilisateurs (ne doit pas être utilisé pour le chiffrement de clef de session, crl, certificat)
- keyAgreement (4)
  - Ce bit est positionné quand la clef publique est utilisée pour la négociation de clé de session.
- KeyCertSign (5)
  - Positionné lorsque la clé publique du sujet est utilisée pour signer les demandes de certificat. Généralement utilisé par les certificat d'AC.
- cRLSign (6)
  - Positionné lorsque la clé publique du sujet est utilisée pour signer les listes de certificats révoqués (LCR, delta LRC, …). Généralement utilisé dans les certificats d'AC.
- encipherOnly (7)

  Ce bit n'a pas de sens en l'absence du bit keyAgreement. Quand ces deux bits sont positionnés, la clé publique du sujet peut être utilisée uniquement pour chiffrer les données durant la négociation de clef de session.
- decipherOnly (8)
  - Ce bit n'a pas de sens en l'absence du bit keyAgreement. Quand ces deux bits sont fixés, la clé publique du sujet peut être utilisée uniquement pour déchiffrer les données durant la négociation de clef de session.

Le champ additionnel **ExtKeyUsage** indique, en supplément du champ *KeyUsage*, un ou plusieurs buts supplémentaires d'utilisation de la clef publique certifiée. Il est important de valider la conformité en relation avec les bits de clef utilisée dans chaque cas. Il est à prévu dans l'interface d'administration de pouvoir ajouter des nouveaux types d'*ExtKeyUsage* et des bits *keyUsage* pour évoluer selon les standards ou des besoins spécifiques. Les champs *ExtKeyUsage* actuellement supportés par le SVC sont:

- Authentification de serveur Web (**serverAuth**) :
  - Bits d'utilisation de clé possiblement compatibles : digitalSignature, keyEncipherment ou
  - KeyAgreement

- Authentification d'un client (**ClientAuth**) :
  - Bits d'utilisation de clé possiblement compatibles : digitalSignature et (ou) KeyAgreement

- Signature de code exécutable (**codeSigning**)
  - Bits d'utilisation des clefs qui peuvent être uniformes : digitalSignature

- Protection de courrier électronique (**emailProtection**)
  - Bits d'utilisation de clé possiblement compatibles : digitalSignature, nonRepudation et (ou) KeyAgreement ou Keyencipherment

- Liaison de hachage d'un objet à une heure obtenue d'une source convenue (**timeStamping**)

  - Bits d'utilisation de clé possiblement compatibles : nonRepudation et/ou digitalSignature

- Serveur de réponse OCSP (**OCSPSigning**)
  - Bits d'utilisation de clé possiblement : digitalSignature et/ou nonRepudiation

- Le champ **privateKeyUsagePeriod** indique la période d'utilisation de la clef privée correspondant à la clé publique certifiée. Il est uniquement présent dans les certificats de signature. Cette extension utilise deux attributs : *notBefore* et *notAfter*. La clé privée associée au certificat ne doit pas réaliser d'opération de signature avant et après la période de temps spécifiée par les deux attributs. Dans le cas d'un certificat qui a une valeur *KeyUsage* à *digitalSignature* et que le champ *privateKeyUsagePeriod* est présent, les deux attributs de temps seront vérifiés.

Spécificité des certificats d'AC :

- Le champ **basicConstraints** identifie si le sujet du certificat peut remplir le rôle d'AC en utilisant la clé privée pour signer les certificats. Il définit aussi la contrainte de longueur du chemin de certification. PRECAL doit s'assurer que l'attribut booléen **cA** est activé de façon appropriée (c-a-d qu'il contient la valeur « faux » pour les certificats utilisateurs et la valeur « vrai » pour les certificats d'AC). La présence du champ à la valeur « vrai », sera vérifiée par le module VERCER.

  - Le champ **pathLenConstraint** ne peut exister que si l'attribut **cA** est à « vrai » et le champ *KeyUsage* contient la valeur *keyCertSign.* il indique le nombre maximum de certificats intermédiaires non auto signés à la suite de ce certificat dans la chaîne de certification. Un certificat est considéré comme auto signé si les noms distinctifs de l'émetteur et du sujet sont identiques et non vides. Le dernier certificat d'une chaîne ne peut donc être auto signé, c'est généralement un certificat d'entité finale. Une longueur de zéro indique que seulement un seul certificat peut suivre dans la chaîne. **Cette longueur est donc supérieure ou égale à zéro.** En l'absence de cette contrainte, aucune limite n'est imposée. Cette contrainte est vérifiée par le module PRECAL.

7.2 REV

Ce bloc fonctionnel permet d'obtenir le statut d'un certificat en temps réel. Pour ce faire, il utilise le protocole OCSP défini dans le RFC2560. Le composant fait appel à deux notions : le back-office et le front-office. Le back-office fait référence au traitement qui est effectué par le service CRL2DB pour assurer les taches internes au bloc REV. Le front Office VALREV fait référence au service qui traite les requêtes des clients.

7.2.1 CRL2DB : Composant de récupération et d'intégration de CRL

Les AC utilisent les LCR pour diffuser des avis de révocation de certificat. Les LCR sont conservées dans la base de données sous forme d'enregistrements et les applications appellent VALREV pour s'assurer que les certificats d'utilisateurs ne sont pas révoqués. Les champs d'une LCR identifient l'émetteur (c.-à-d., l'AC), la date et l'heure de production de la LCR, la date de production de la

prochaine LCR et les certificats révoqués des utilisateurs. Une AC peut également ajouter des champs additionnels d'information supplémentaire sur une entrée particulière, ou des champs additionnels portant sur l'ensemble de la LCR (voir la section 7.2.2).

La LCR doit utiliser la syntaxe *CertificateList* telle que définie dans le standard RFC3280.

Le composant CRL2DB traite les CRL et vérifie la présence des champs suivants :

- Le champ **signatureAlgorithm** indique l'algorithme pour la signature de la LCR.
- Le champ **signatureValue** contient la signature de l'AC qui a émis la LCR. La signature doit être validée en utilisant le certificat de l'AC qui l'a émise. Il est aussi nécessaire d'avoir validé lors de la création de la LCR (depuis l'interface d'administration) que le certificat qui a émis la LCR possède une valeur de *KeyUsage* à cRLSign (le droit de signer des LCR).
- Le champ **version** indique qu'il s'agit d'une LCR v2. Sinon, ce champ est absent ou indique V1.
- Le champ **issuer** indique le nom distinctif de l'émetteur de la CRL. La présence de ce champ est vérifiée et son contenu validé en rapport avec le certificat de l'AC qui a émis cette CRL.
- Le champ **thisUpdate** indique à quel moment la LCR a été produite.
- Le champ **nextUpdate** indique à quel moment sera effectuée la prochaine mise à jour de la LCR ; si le champ PKIX a été sélectionné dans l'interface d'administration, ce champ est pris en compte pour définir la période de récupération de la prochaine CRL. Sinon, le paramétrage de récupération de la LCR est réalisé depuis l'interface d'administration en définissant une période de récupération (en minutes) des CRL.
- Le champ **revokedCertificates** contenant la ou les séquences du ou des champs *userCertificate* (qui permettent d'identifier les certificats d'utilisateur et d'AC), *revocationDate*, et *crlEntryExtensions* pour indiquer le numéro de série de chaque certificat révoqué, l'heure de la révocation, et les champs additionnels.

7.2.2 Extensions

Le champ ou les champs **crlExtensions et crlEntryExtensions**

Les CRL utilisent deux types d'extensions

- Les champs *crlExtensions* ajoutent de l'information à la LCR et sur son émetteur, et fournissent les mécanismes pour contrôler la taille des LCR.
- Les champs *crlEntryExtensions* de LCR ajoutent de l'information sur une entrée particulière dans la LCR.

7.2.2.1 Les extensions du champ *crlExtensions*

- Le champ **authorityKeyIdentifier** identifie la clé publique qu'il faut utiliser pour vérifier la signature de la LCR. Il permet de distinguer les différentes clés utilisées par la même AC. Il peut contenir l'identificateur de clé implicite ou un identificateur de certificat explicite. Ce champ est utile lorsqu'une AC utilise plus d'une clé. Les AC n'utilisent pas le principe des doubles clefs. CRL2DB ne traitera pas ce champ.

- Le champ **cRLNumber** transporte un numéro de séquence croissante monotone pour chaque LCR émise par une AC donnée. Ce champ permet de connaître une éventuelle perte dans la récupération d'une CRL. Les CRL actuellement utilisées dans le cadre du référencement MINEFI n'utilisent que très rarement cette extension. Il sera appliqué une vérification pour les CRL qui comportent cet identifiant. CRL2DB vérifiera la continuité dans les numéros de CRL ; En cas de défaut dans la linéarité de numéro une alerte sera adressée au service de supervision. La CRL sera acceptée pour intégration dans le service de validation.

- Le champ **deltaCRLIndicator** indique qu'une LCR est de type delta. La présence de ce champ est vérifiée. Ce champ doit permettre au composant CRLFinder de spécifier le type de CRL récupérée (complète, Delta, partielle).

- Le champ **issuingDistributionPoint** désigne le point de distribution correspondant à une LCR particulière et indique si celle-ci est limitée aux seules révocations de certificats d'entité finale, de certificats d'AC ou à un seul ensemble restreint de causes. Il indique que la LCR peut contenir des entrées d'autorités de certification autres que l'autorité qui a signé et émis la LCR. Les champs suivants peuvent être présents :

    - *onlyContainsUserCerts* : contient seulement des certificats utilisateurs.
    - *onlyContainsCACerts* : contient seulement des certificats d'AC. Si ce champ contient la valeur « vrai », cette CRL est une LAR (liste d'AC révoquées).
    - *indirectCRL* : spécifie que la CRL est indirecte (émise par une AC différente de celle qui a émis le certificat.). la vérification doit permettre de valider le certificat qui a émis cette CRL.

    Remarque : si la valeur de *onlyContainsUserCerts* et de *onlyContainsCACerts* sont simultanément à « faux », le processus CRLProcess doit vérifier si la CRL ne contient pas une AC révoquée dans sa liste. La CRL ne sera pas intégrée à la base de données avant la réalisation de cette vérification.

  Les Champs *indirectCRL, onlyContainsCACerts* et *onlyContainsUserCerts* doivent être vérifiés et analysés en cas de présence de l'un de ces champs.

7.2.2.2 Les extensions du champ *crlEntryExtensions*

- Le champ **reasonCode** indique la raison de la révocation du certificat. Il n'est pas pris en compte dans la version 2 du SVC, car ce champ n'est pas utilisé correctement par les Autorités de certification référencées

- Le champ **holdInstructionCode** est un identificateur d'instruction pour la décision à prendre en présence d'un certificat suspendu. Le SVC ne gère pas la notion de certificat suspendu.

- Le champ **invalidityDate** indique la date à laquelle on a appris ou soupçonné que la clé privée avait été compromise ou que le certificat devait, pour toute autre raison, être considéré comme incorrect. Cette date peut être antérieure à la date de révocation indiquée dans

l'entrée de LCR, c'est-à-dire la date à laquelle l'AC a procédé à la révocation. Ce champ n'est pas utilisé par les Autorités de certification. Il ne sera pas pris en compte par le SVC.

- Le champ **certificateIssuer** identifie l'émetteur du certificat associé à une entrée d'une LCR indirecte (c.-à-d. une LCR dont le bit de l'indicateur indirectCRL est activé dans le champ *issuingDistributionPoint*). Ce champ est à vérifier en cas de présence de CRL indirecte.

7.2.3 Traitement à effectuer sur la CRL

Cette section décrit le traitement qui est effectué sur les CRL lors de leur récupération par le module CRL2DB :

1. Vérifier que la date du champ *thisUpdate* est supérieure au dernier enregistrement du champ *thisupdate* contenu dans la base de données ;
2. Vérifier que l'heure de récupération se situe entre les valeurs des champs thisUpdate et nextUpdate ;
3. Vérifier la signature de la LCR en utilisant la clé publique du certificat et des paramètres de l'émetteur, s'il y a lieu ;
4. Vérifier l'itinéraire de certification du certificat de signature de l'émetteur de la LCR**.** La vérification implique une validation de la date de validité et du statut des différents certificats du chemin de certification. En complément, cette analyse doit être aussi présente dans l'interface d'administration, lors de l'intégration de nouvelles CRL ;
5. S'il s'agit d'une CRL version v2;
    - Vérifier que le champ CRLNumber contient une valeur supérieure à celle de la dernière LCR que possédait le SVC;
    - Vérifier que l'émetteur de la LCR est l'émetteur du certificat (ou qu'il correspond à la valeur indiquée dans le champ additionnel *cRLDistributionPoints* ;
6. Vérifier que le nom du sujet dans le certificat X.509 de l'émetteur de la LCR correspond au nom de l'émetteur de la LCR
    - Si non, vérifier que le bit indirectCRL est activé dans le champ *issuingDistributionPoint.* Par la suite, vérifier la validité du certificat identifié dans le champ **certificateIssuer.**
7. Vérifier si le numéro de série de certificat figure dans la LCR. Si un certificat qui figure dans la LCR est le certificat d'AC, le SVC doit émettre une alerte et les processus suivants sont activés :
    - L'administrateur doit être prévenu de la révocation d'une AC
    - l'AC doit être révoquée ;
    - si les champs *onlyContainsUserCerts* et *onlyContainsCACerts* sont à « FAUX », la CRL ne doit pas être saisie automatiquement par le module CRL2DB. Elle sera conservée pour être intégrée manuellement, après validation par l'administrateur que les certificats utilisateurs ont bien été préalablement révoqués par l'AC. Pour les autres, la CRL est insérée automatiquement.

Spécificité des DeltaCRL

En cas de réception d'une DeltaCRL par le module CRL2DB, les points suivants sont vérifiés :

- Vérifier que l'émetteur de la DeltaCRL est le même que celui de la CRL complète ;
- Si une CRL complète inclut une extension *issuingDistributionPoint* (IDP), CRL2DB vérifie que la DeltaCRL contient aussi les mêmes valeurs pour le champ IDP (*onlyContainsUserCerts* et

onlyContainsCACerts), et effectue le traitement du point 7.2.3.7. Si le champ est absent dans la CRL complète, vérifier qu'il est aussi absent de la DeltaCRL.

7.2.4 VALREV : composant de vérification du statut d'un certificat

7.2.4.1 VALREV OCSP

Le composant de vérification des statuts des certificats se réfère au RFC2560 (OCSP) et prend en compte les restrictions imposées par le programme Copernic sur le protocole.

L'ensemble du standard OCSP est respecté pour les champs et les mécanismes définis comme obligatoires dans le RFC2560.

Les champs optionnels retenus sont énoncés dans la section 6.4.1

7.2.4.2 VALREV « étendu »

Ce service est accessible à travers un appel OCSP standard. Comme il n'est pas prévu dans la norme OCSP la possibilité d'adresser le certificat au serveur, la validation du chemin de certification se basera sur le DN de l'AC passé dans la requête. En revanche, ce service a besoin de connaître le domaine de confiance de l'appelant pour pouvoir appliquer la politique de validation par défaut définie pour chaque domaine de confiance. Dans notre cas, le domaine de confiance de l'appelant sera positionné dans l'extension *RequestorName* de la requête OCSP. Ce service est dénommé "service OCSP étendu ".

Le traitement de validation est le même que pour le composant VALREV OCSP.

7.2.4.3 VALREV « enrichie »

Ce service est accessible à travers une classe cliente JAVA dont le protocole est propriétaire. Cet appel est destiné aux applications désirant effectuer un contrôle plus strict en appliquant une politique de validation différente de la politique par défaut du domaine de confiance : c'est l'argument supplémentaire à passer par rapport au deuxième cas. Le résultat est plus complet, notamment dans le contenu de la réponse.

Le type de requête et de réponse sont définis dans la section 6.4.1.3

7.3 Composant de validation des chemins de certification (PRECAL)

Le service de validation des certificats reçoit par le biais d'une interface d'administration les certificats des différentes AC et leurs chemins de certification. Les traitements suivants sont appliqués sur les certificats d'AC :

Spécificité des certificats d'AC

Les champs suivants ne seront pas implémentés dans le SVC, car ils ne sont pas respectés par les autorités de certifications référencées par le MINEFI :

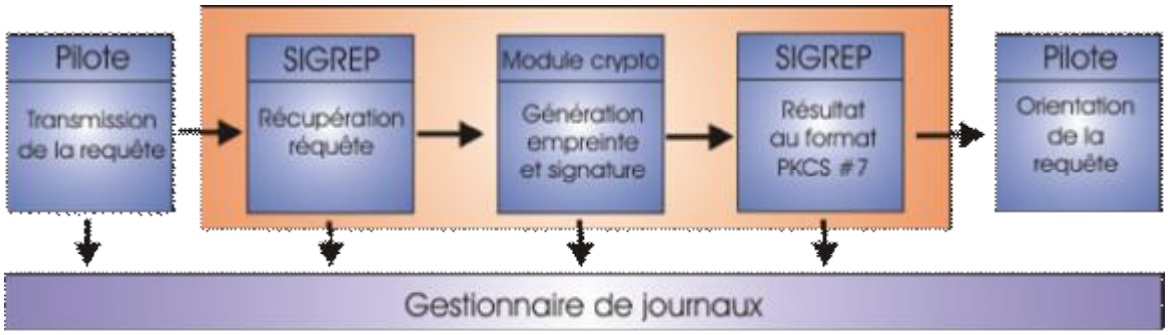
- Le champ **nameConstraints** utilisé uniquement dans les certificats d'AC, indique un espace de nom dans lequel doivent figurer tous les noms de sujet des certificats subséquents du chemin de certification (ces restrictions s'appliquent aussi bien au nom du sujet qu'à son nom alternatif).
    - Les restrictions sont exprimées à l'aide de deux contraintes, les permissions (*permittedSubTrees*) et les exclusions (*excludedSubtrees*). À part la définition de la contrainte littérale la longueur minimale doit être zéro et la longueur maximale non spécifiée (absente).

- Le champ **policyConstraints** précise les contraintes selon lesquelles il faut soit une identification de politique de certification explicite, soit empêcher l'équivalence de politiques pour le reste du chemin de certification. L'extension définissant les contraintes politiques peut être utilisée dans les certificats émis pour une AC. Les contraintes s'appliquent sur la chaîne de validation de deux manières : Soit Pour interdire l'équivalence politique (*inhibitPolicyMapping*) ou, soit pour obliger que chaque certificat dans la chaîne contienne un identifiant acceptable de sa politique (*requireExplicitPolicy*).
    - Si le champ *inhibitPolicyMapping* est présent, sa valeur indique le nombre de certificats additionnels qui peuvent apparaître dans la chaîne avant que l'équivalence de politique ne soit plus permise. Par exemple, une valeur de 1 indique que l'équivalence de politique peut être traitée dans les certificats émis par le sujet du certificat, mais pas dans les certificats ultérieurs.
    - Si le champ *requireExplicitPolicy* est présent, sa valeur indique le nombre de certificats additionnels qui peuvent apparaître avant qu'une politique explicite soit nécessaire pour la chaîne en entier. Dans ce cas, il est nécessaire, pour tous les certificats dans le chemin restant, d'avoir un identifiant acceptable de politique dans leur extension de politiques de certification (*certificat policies*). Un identifiant acceptable de politique est un identifiant de politique demandé par le vérificateur de la chaîne de certification ou l'identifiant d'une politique qui a été déclarée comme équivalente à un identifiant valide (via *policy mapping*).

- Le champ InhibitAnyPolicy peut-être utilisé dans les certificats des AC pour indiquer que l'OID spéciale de politique (***anyPolicy***) n'est plus une politique explicite à partir d'un certain niveau. La valeur indique le nombre de certificats additionnels qui peuvent apparaître dans la chaîne avant que la politique (***anyPolicy***) ne soit plus acceptée.

7.4 SIGREP

SIGREP est le composant de signature du SVC. Il réalise l'empreinte des données qui lui sont adressées par le pilote et signe l'empreinte.



Principe de fonctionnement :

- Le pilote adresse au composant de signature la réponse qui sera adressée au client
- Le composant SIGREP adresse la réponse au module cryptographique par le biais d'une interface de communication. En cas d'utilisation de cryptographie logicielle, le composant SIGREP s'adresse soit à OpenSSL, soit à une librairie JAVA pour la génération de la signature;
- Le module cryptographique génère l'empreinte depuis la réponse ;
- L'empreinte est signée ;
- Le résultat est encaspulsé dans un format PKCS #7 ;
- Le composant SIGREP adresse le résultat au pilote.

Caractéristiques :

:

- Le format d'empreinte est de type SHA-1 ou SHA-2 ;
- Le format de la signature est de type PKCS #7.

8 Supervision

Le SVC sera supervisé par le logiciel Nagios. Les spécifications de la supervision sont détaillées dans les spécifications techniques détaillées du SVC.

Le principe de supervision est de fournir un ensemble de variables auprès du service de supervision. Les variables sont contenues dans des conteneurs compartimentés par zone. Les collecteurs adressent leurs données au service de supervision Nagios, tel que représenté dans le schéma suivant :

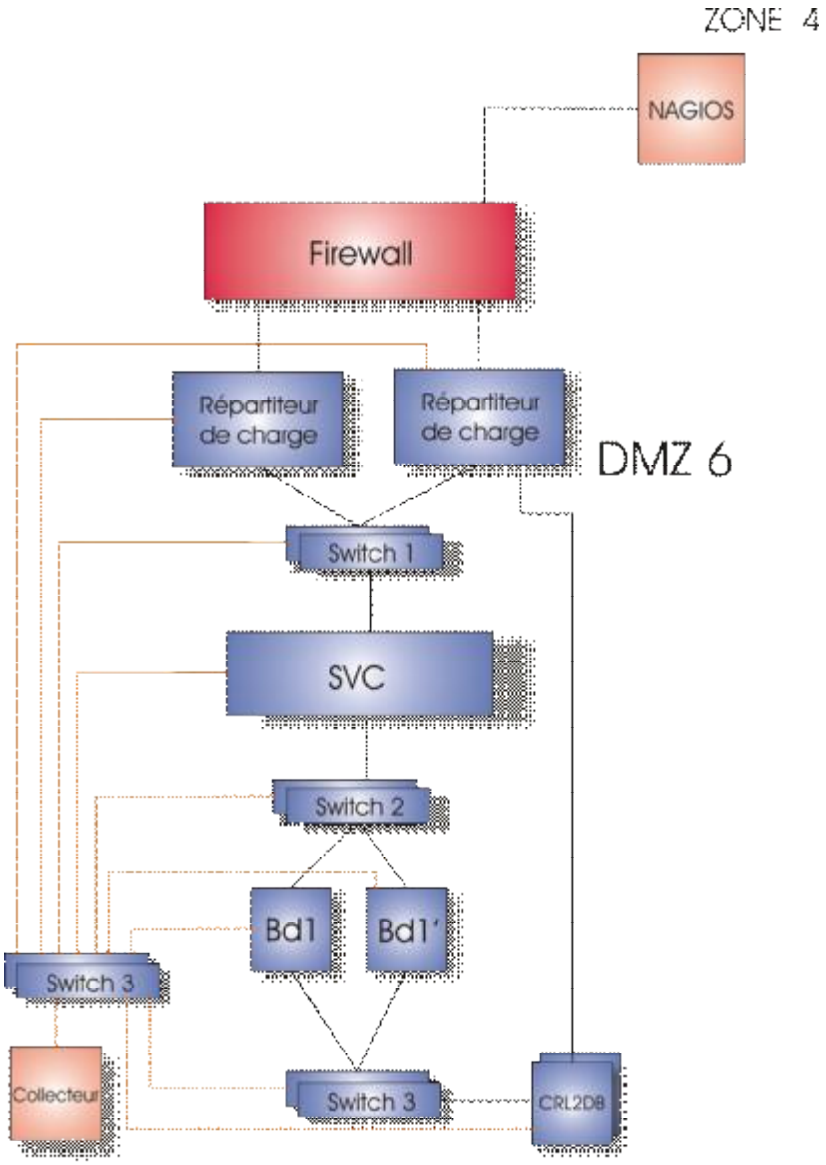
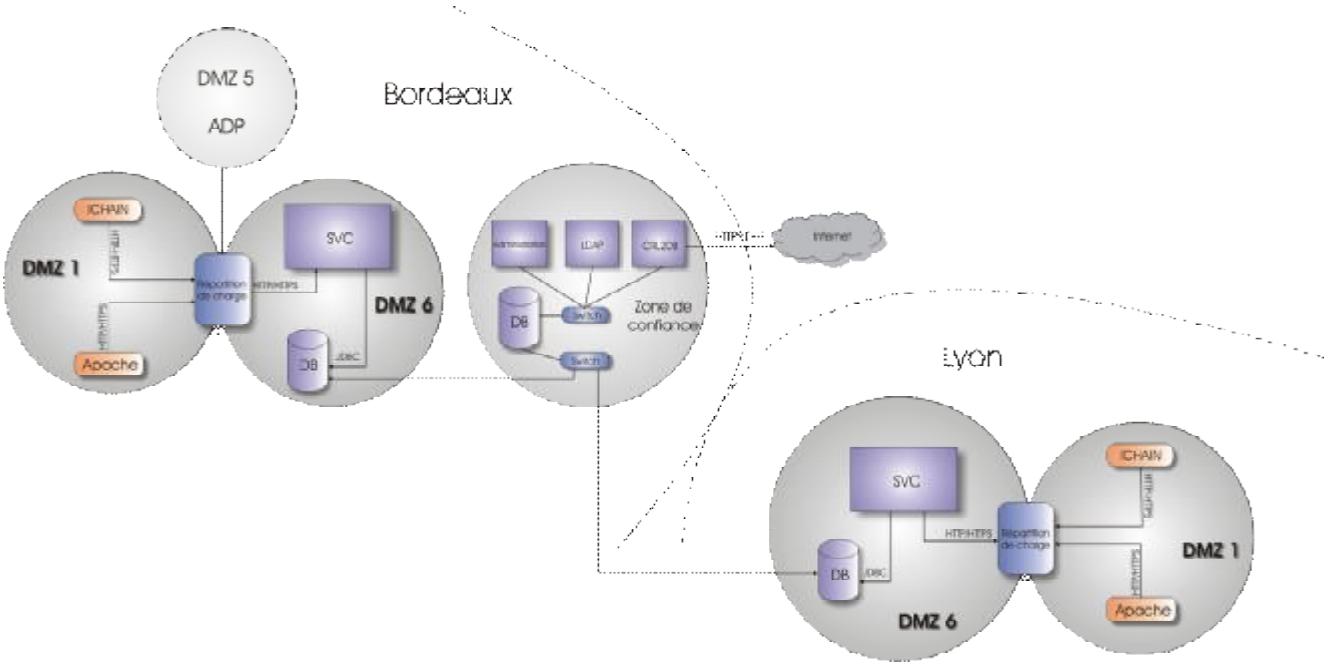


Schéma de principe de la collecte de la supervision

9 Architecture générale

Le SVC à été développé pour permettre une mutualisation des bases référentielles et des modules Back Office. Cette mutualisation permet de déployer le back-office sur un seul et générer plusieurs sites de validation de certificat.

Le schéma suivant présente l'architecture pour le site de Bordeaux et de Lyon.

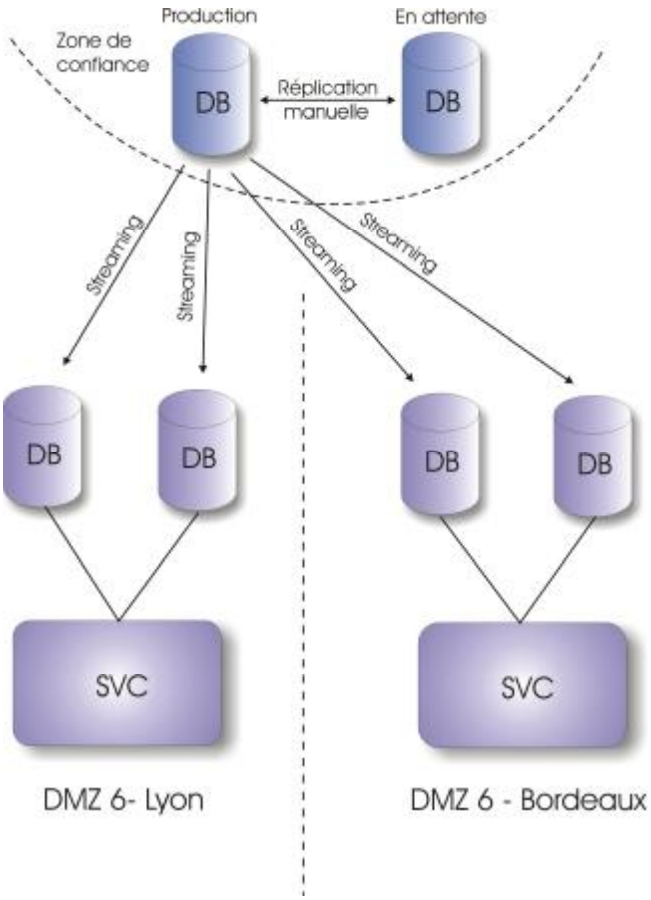


Le back-office est déployé dans la zone de confiance pour permettre une mutualisation de la récupération des CRL et de leur référencement dans la base de données maître.

10 Base de données

L'architecture de base de données proposée dans le cadre du SVC de niveau doit permettre d'avoir un référentiel de données commun pour les différents sites de validation.

Le composant de récupération des CRL (CRL2DB) est positionné sur un point unique de l'architecture, pour permettre d'obtenir un unique référentiel pour l'ensemble des sites de production. Le composant CRL2DB alimente une base de données positionnée dans la zone de confiance. La base de données de la zone de confiance alimente les bases de données des différents sites de production (Bordeaux et Lyon). Le schéma présente l'architecture de déploiement des bases de données.



Principe de sauvegarde des bases de données

10.1 Sauvegarde en ligne et hors ligne

Le composant CRL2DB qui récupère les CRL depuis Internet, réalise une vérification du contenu de la base de données et ne dépose que le Delta par rapport au contenu de la CRL. Ce mode de fonctionnement permet de considérer que la réplication de base de données n'est pas un élément critique et ne nécessite pas une sauvegarde en temps réel. En cas de perte de la base de données, une restauration manuelle sera effectuée sur une machine dédiée. Le Delta entre les dernières CRL et ceux contenus dans la base de données sera mis à jour automatiquement par le composant CRL2DB.

Les bases de données en production sont positionnées dans des DMZ de la PAS. La base de données référentielle et pour sa part positionnée dans la zone de confiance. La charte de sécurité de la DGI ne permet pas d'initialiser des connexions entre la PAS et la zone de confiance. Pour résoudre cette problématique de réplication entre bases de différentes zones, une réplication de type STREAMING est préconisée entre la zone de confiance et la PAS. Le Streaming permet de répliquer une base de données dans en initiant la connexion depuis la base maître en utilisant un protocole HTTP, pour le transfert du flux de données.

Pour sécuriser les bases de données de production, deux bases par sites seront en fonction simultanément, les serveurs frontaux du SVC utiliseront un mécanisme de réparation de charges fournies par le Listener des bases de données pour appeler tels ou tels base.

Synthèses

La zone de confiance utilisera une machine positionnée en attente en cas de perte de la base de données maître. La réplication utilisera un mécanisme de réplication manuelle.

Les bases en DMZ seront répliquées par la base de la zone de confiance en utilisant une méthode de STREAM encapsulé dans un flux HTTP.

Les bases en DMZ seront toutes en fonction simultanément, un processus de répartition de charge intégrée sur les serveurs frontaux géré la répartition des appels des SVC vers les bases situées en DMZ.

11 Service d'archivage

Les données qui seront archivées sont définies selon leur criticité et les moyens disponibles pour mettre en place l'archivage. Les données critiques identifiées pour être sauvegardées sont :

- Le contenu de la base de données de référence
- Les politiques de validation
- Les fichiers journaux des serveurs de la plate-forme SVC

L'archivage est effectué de la manière suivante :

- Pour la base de données, un export de base sera réalisé périodiquement, le résultat de cet export sera copié sur bande.
- Les fichiers journaux sont gérés par logrotate et conserver sur bande pour une durée d'un mois.
- Pour les politiques de validation, une copie de sauvegarde sera effectuée et adressée à l'archivage de l'ADP.

L'archivage ADP est réalisé de manière manuelle, une disquette contenant l'ensemble des politiques de validation seront copiés par le biais d'un script sur disquette et transmises à l'ADP pour archivage.

12 WebService

Les différents AC référencés et CRLs sont conservés au sein du SVC pour effectuer les différents contrôles sur les certificats. Le SVC met à disposition des clients une interface d'appel pour récupérer les certificats des ACs et les CRL.

Les certificats d'AC peuvent être utilisé pour les besoins d'authentification au sein des serveurs ou pour conserver les contextes de signature au sein de l'ADP.

Ce service est nommé WebService, il permet via un appel SOAP de créer une requête permettant de récupérer les certificats et des informations sur les AC selon différents modes d'appel :

- Appel pour récupérer l'ensemble des ACs
- Obtention d'une AC à partir de son DN
- Obtention d'une chaîne de certification pour une AC particulière
- Obtention de la dernière CRL d'une AC particulière
- obtention de l'ensemble des CRL entre deux dates données à partir d'un DN d'AC
- Obtention de l'ensemble des DN et des certificats des AC d'un domaine de confiance à partir d'un OID de domaine de confiance
- A partir d'un OID de domaine de confiance et d'un OID de PV, obtention de :
    - liste des DN des 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
    - l'identifiant de PV réellement utilisé pour le calcul (identifiant et version de la PV active).
    - le certificat de l'AC

Le SVC fournit au client qui souhaite faire appel au WebSErvice, la classe et la méthodologie d'appel.

13 Administration

13.1.1 Présentation générale

L'interface d'administration permet de maintenir à jour la base de données sur laquelle s'appuie le SVC. Elle permet l'administration fonctionnelle du SVC par la gestion :

- du paramétrage des tables de la base,
- le déclenchement manuel du précalcul des chemins de certification.
- Génération des politiques de validation
- L'Intégration des différentes AC et de leur chemin
- La consultation des informations précités

13.1.2 Rôles applicatifs

L'interface d'administration est accessible en `https` (HTTP/SSL V2) à des administrateurs authentifiés. Il existe deux profils d'administrateurs :

- les « exploitants », dont les actions sur l'interface ne modifient pas le paramétrage en production,
- les « responsables », dont les actions sur l'interface modifient le paramétrage en production.

Seules les actions des responsables donnent lieu à archivage.

Chaque administrateur est soit exploitant, soit responsable.

Les exploitants peuvent effectuer les actions suivantes :

- afficher la liste synthétique des AC,
- afficher le détail d'une AC et ses LCR/LAR,
- afficher la liste des PV (OID, version, nom, isValid, isActive),
- afficher le détail d'une PV (ensemble des champs),
- créer une nouvelle PV. La saisie d'une PV et son enregistrement la place dans l'état « à valider »),
- modifier une PV existante (ce qui crée dans la base une nouvelle version, dans l'état « à valider », de la PV correspondante, pour le même OID).

Les responsables peuvent effectuer les actions suivantes :

- créer un domaine de confiance (table `trustdom`),
- afficher la liste des AC,
- afficher le détail d'une AC et les URI de ses LCR/LAR,
- créer une nouvelle AC et les URI de ses LCR/LAR et lui affecter une liste de domaines de confiance,

- modifier une AC et ses LCR/LAR,
- afficher la liste des PV (OID, version, nom, isValid, isActive),
- afficher le détail d'une PV (ensemble des champs),
- valider ou refuser les PVs en état « à valider »,
- activer une PV validée (ce qui désactive l'éventuelle PV active de même OID),
- déclencher manuellement le précalcul des chemins de validation. (un rapport est affiché à la fin du traitement sous la forme d'une liste de DN, ce qui permet à l'administrateur de valider le calcul ou d'effectuer les éventuelles actions correctrices requises. S'il valide le calcul, le résultat est écrit dans la base de données qui servira de base pour valider les certificats finaux),
- déclencher une vérification fonctionnelle du système par envoi d'une requête au SVC avec un certificat de test (un ensemble de certificats de test est présent sur le serveur d'administration), et visualiser la réponse au format XML,
- afficher l'ensemble des logs accessibles,
- afficher la liste des utilisateurs de l'application,
- créer un nouvel utilisateur (nom, prénom, login, mot de passe, email, reçoit ou non les emails de supervision, rôle, état autorisé ou suspendu),
- modifier un utilisateur,
- créer de nouvelles valeurs de extKeyUsage.
- Alimenter les certificats des AC,
- Vérifier la complétude des imports de LCR,
- Visualiser une liste de révocation dans le temps, lors des contrôles a posteriori,
- relancer une récupération de LCR,
- Importer manuellement une LCR.

Alimentation des certificats des AC

- D'importer un nouveau certificat AC,
- D'importer manuellement une LCR,
- De créer les types de LCR publiés par l'AC.

Import des certificats

L'import s'effectue à travers le dépôt du fichier certificat au format DER ou PEM. Cette opération consiste à :

- stocker le contenu du fichier dans la base de données,
- cette création est automatique, elle est basée sur le parcours du chemin de certification. Ceci impose que la création des AC commence par l'AC racine.
- Lors de l'ajout d'une AC, celle-ci est positionnée dans un état :
-
  - Attente : lors de l'intégration initiale de l'AC
  - Terminé : intégration de la LCR de l'AC.

- Terminé sans LCR : Identique au statut Terminé mais l'AC est opérationnelle sans déclaration de LCR.
- Mise à jour : AC en cours de mise à jour de la LCR (récupération automatique)
- Erreur : AC en erreur. Certaines LCR de cette AC sont périmées ou ne sont pas correctes.

Le dialogue de visualisation des certificats passe à travers un masque de recherche sur les critères suivants :

- DN ou un élément du DN,
- ou la totalité des AC.

Le masque suivant ramène pour chaque AC trouvée :

- la hiérarchie des AC.

Le détail par AC affiche tous les champs du certificat sélectionné. Un lien vers les listes de révocation permet de visualiser les LCR associés à l'AC. Cette liste indique dans un premier temps :

- La liste des types de LCR liée à l'AC. Cette liste fournit les renseignements stockés dans la table CRL :
  - Signataire de la LCR, type de la LCR,
  - La date de la dernière mise à jour,
  - La date de la prochaine mise à jour.

Création des types de LCR

Cette interface permet de créer les types de LCR publiés par une AC. Les renseignements sont issus du contenu des LCR de l'AC. Le masque de saisie permet de saisir les champs suivants :

- `idcrlissuer` : identifiant de l'AC signataire de LCR,
- `idcrl` est automatique générée à partir d'une séquence qui garantit son unicité,
- `subject` : s'agit-il d'une LAR ou d'une LCR,
- `type` : complet, partiel ou delta LCR,
- `thisUpdate` et `NextUpdate` seront mis à jour lors du prochain import de LCR.

À part le cas d'une LCR complète, il peut exister plusieurs occurrences de LCR pour une AC lorsque les listes sont partielles ou delta.

Importer manuellement une LCR

Cette interface permet d'importer manuellement une LCR dans le cas ou l'importation automatique a échoué, oui pour initialiser une nouvelle AC. Le traitement est identique à l'importation automatique.

Compléter manuellement une LCR

Cette procédure permet de pallier l'absence de LCR d'une AC, lorsque celui-ci est dans l'impossibilité de publier sa LCR. La procédure manuelle (envoi de la LCR par fax) consiste donc à compléter éventuellement la dernière liste connue et à mettre le statut de l'AC à T(erminé).

AC sans LCR

Dans le cas où une AC ne peut fournir de CRL, une AC peut être positionnée à « Terminer sans CRL » permettant de répondre par la positive en cas de vérification de son statut de révocation. Ce cas est utilisé essentiellement pour les AC Racines qui ne fournissent pas de CRL. Ce qui permet à la finalité de valider une chaîne de certification complète sans avoir obtenu de LAR.

Vérifier la complétude des LCR.

La vérification doit être appliquée pour chaque LCR d'une AC. Pour chaque LCR, il suffit de vérifier que :

- La date courante est inférieure à `nextUpdate + graceAfter`, si `nextUpdate` existe. Dans le cas contraire, la LCR est périmée
- La date courante ne doit pas être inférieure à `thisUpdate`, dans le cas contraire cela signifie que ce champ a été mis à jour manuellement, et non pas par `crlProcess`.

A la fin de la vérification, un rapport indiquera par AC :

- la liste des LRC publiées, leur type
- pour chaque LCR
  - `crlnum`
  - `basenum`
  - `thisUpdate`
  - `nextUpdate`

Relancer une récupération de LCR

Cette fonctionnalité permet de re-synchroniser la base en cas d'incident. Le traitement consiste à empiler une requête à la place de crlProcess, le type de la requête est de type (E)xceptionnel.

Important : les responsables n'ont pas accès aux autres fonctions, qui sont dédiées aux exploitants.

13.1.3 Contraintes sur les données

Création d'une AC

La soumission de création d'une AC effectue un appel au SVC pour vérification du certificat de l'AC. La PV invoquée doit stipuler que l'indicateur *cA* du champ additionnel *basicConstraints* doit valoir « vrai ».

Création d'une CRL

Le certificat de l'AC qui émet la CRL (AC déléguée dans le cas d'une CRL indirecte) doit avoir la valeur *cRLSign* parmi les valeurs du champ *keyUsage.*

Le chemin de certification du certificat de l'AC émettrice d'une CRL doit être valide (vérification dans la table *validpath* – le précalcul des chemins de validation doit avoir été déclenché entre la saisie de l'AC et la saisie de la CRL pour que cette vérification soit possible).

Import manuel d'une CRL

La version présente dans la CRL doit être identique à la version déclarée dans la table *crl* (champ *version*).

Création d'une PV

- Chaque nouvelle PV créée se voit affecter un identifiant *idvp* résultant de la concaténation entre un préfixe (paramètre de l'application) et un identifiant entier (incrément sous la forme d'une séquence).
- La première version de chaque PV porte le numéro 1. Chaque nouvelle version de cette PV porte le numéro de version suivant, par incrémentation séquencielle.
- Parmi les versions de PV de même identifiant, une seule version de la PV est active à un instant donné. L'activation d'une version de PV désactive automatiquement la version de cette PV précédemment active.
- La version active de la PV par défaut ne peut pas contenir l'appel à VERCER et doit contenir l'appel à VALREV.
- Les champs de description d'une PV (stockés dans la table refparam) doivent respecter les cardinalités décrites dans le document des spécifications techniques détaillées du présent document).
- Règles de gestion pour les champs *requiredExplicitPolicy* et *policyOID*
- Si *requiredExplicitPolicy* est true, le champ *policyOID* est obligatoire,
- Si *requiredExplicitPolicy* est *false*, le champ *policyOID* est vide.
- Règles de conversion pour *extendedKeyUsage* : Pour la déclaration de la liste des *extendedKeyUsage* d'une PV, l'interface présente à l'utilisateur la liste des *extendedKeyUsage* sélectionnables (disponibles dans la table *extendedKeyUsage* de la base de données - *serverAuth*, *clientAuth*, *codeSigning*, *emailProtection*, *timeStamping* et *OCSPSigning* dans un premier temps). Lors de la création de la PV en base, les choix de l'utilisateur sont convertis en OID selon la correspondance présente dans cette table.

Association entre une PV et un domaine de confiance

- Une même PV peut être utilisée par plusieurs domaines de confiance, et un domaine de confiance peut comporter plusieurs PV (plusieurs OID), dans plusieurs versions différentes.
- Pour un domaine de confiance donné, une seule PV est la PV par défaut.
- La PV par défaut ne peut pas contenir VERCER.

Gestion des alertes

Certaines alerte de haut niveau (récupération de CRL impossible, URL indisponible, …) les responsable peuvent recevoir une alerte par courrier électronique. Cette notification est configurable au niveau de la gestion des utilisateurs de l'interface d'administration.

-

14 Gestion des erreurs

Décrit l'ensemble des messages d'erreurs géré par le SVC. Les identifiants par module sont affectés de la manière suivante :

100-149 Pilote
150-199 VERCER
200-249 VALREV
250-299 VERCAC
300-349 SIGREP
350-399 CRLFinder
400-449 CRLPRocess
450-499 PRECAL

L'exhaustivité des messages d'erreurs est fournit par composant dans la section suivante.

14.1 CRL2DB

14.1.1 CRLFinder

Code	Message	Signification
350	Signature CRL incorrecte	signature de la CRL invalide
351	Thisupdate absent	Absence du champ *thisupdate*
352	signatureAlgotrithm Absent	Absence du champ signatureAlgotrithm
353	CRL Invalide	format de CRL invalide
354	Problème récupération CRL	Erreur de récupération : 404 pas de CRL à cette URL
355	Erreur du serveur de récupération	500 erreur interne du serveur de récupération des CRL
356	Serveur indisponible	503 Serveur de récupération des CRL indisponible

14.1.2 CRLProcess

Code	Message	Signification
400	Signature CRL incorrecte	signature de la CRL invalide
401	Thisupdate absent	Absence du champ *thisupdate*
402	nextupdate absent	Absence du champ *nextupdate*
403	signatureAlgotrithm Absent	Absence du champ signatureAlgotrithm
404	*revokedCertificates* Absent	Absence du champ *revokedCertificates*
405	CRLnumber incorrecte	Champs CRLnumber incorrecte
406	Chemin de certification invalide	Chemin de certification invalide
407	*issuer* de la CRL invalide	Champ *issuer* invalide
408	Emetteur de la CRL invalide	L'emetteur n'est pas celui de CRL et ce n'est pas une CRL Indirecte
409	indirectCRL et certificateIssuer Absent	Absence des champs indirectCRL et certificateIssuer
410	Erreur d'accès base	Problème d'Insertion dans la base

14.2 PRECAL

Code	Message	signification
450	Le calcul ne peut-être réalisé	Calcul du chemin de certification impossible
451	Certificat invalide	Format des certificats non reconnue
452	Connexion base de données impossible	Problème de connexion à la base de données

14.3 SIGREP

• Code	• Message	• signification
• 300	• Requête malformé	• Requête malformé
• 301	• La carte cryptographique ne répond pas	• La carte cryptographique ne répond pas
• 302	• Certificat SVC en erreur	• Ne peut atteindre le certificat du SVC

•