SPECIFICATIONS ET MISE EN OEUVRE DU STANDARD D'INTEROPERABILITE ENTRE ORGANISMES DE LA SPHERE SOCIALE

Version 1.0

ON-X S.A. *est une société du* ***Groupe ON-X***
15, quai Dion Bouton – 92816 PUTEAUX cedex. Tél : 01 40 99 14 14 – Fax : 01 40 99 99 58.
SA au capital de 3 752 000 Euros. RCS Nanterre B 391 176 971. Siret 00037. Code APE 721 Z.
www.on-x.com

Identification et historique

Identification client

Référence client	CCTP 0592110
Interlocuteur	Thierry LAHALLE – thierry.lahalle@sante.gouv.fr
Interlocuteur	Michel JANIN – michel.janin@cnav.fr
Interlocuteur	Patrick Mery – patrick.mery@cnamts.fr

Identification ON-X

Référence ON-X	2005-1001-001
Version	1.0
Date	03/04/06
Nombre de pages	69
Interlocuteur	Olivier Chapron – Directeur du projet – Consultant Manager 01 40 99 14 14 – olivier.chapron@edelweb.fr
Interlocuteur	Patrick Vigneras – Chef de projet 01 40 99 14 14 – pvigneras@on-x.com

Visa

Fonction	Nom
Rédaction	Patrick VIGNERAS
Vérification	Peter SYLVESTER
Approbation	Olivier CHAPRON

Historique

Date	Auteur	Version	Objet
09/12/05	PVS	0.1	Création du document, version préliminaire
21/12/05	PVS	0.4	Révision interne
06/01/06	PVS	0.5	Report vers spécifications appliquées
17/01/06	PVS	0.6	Structure définitive du document
24/01/06	PVS	0.7	Révision interne
03/02/2006	OCN/PSE	0.81	Révision interne avant envoi aux membres du comité de suivi
14/02/2006	PVS	0.84	Mise à jour après comité de suivi
23/02/2006	OCN	0.9	Version à diffuser pour l'appel à commentaires
17/03/2006	PVS	0.92	Mise à jour après appel à commentaires
21/03/2006	OCN	0.99	Version pré-finale à valider par les organismes
03/04/2006	OCN+PSR +PVS	1.0	Version finale approuvée formellement

Références

Identifiant	Titre
R1	Standard d'interopérabilité inter-organismes – *Olivier CHAPRON, Peter SYLVESTER – version 1.0 (13 juillet 2005)*
R2	http://www.ssi.gouv.fr/fr/
R3	Liberty ID-WSF Security Mechanisms – *Gary Ellison, Sun Microsystems, Inc. – version 1.2 (19 mai 2005)*

Table des matières

1. Introduction

1.1. Objet du document

Ce document présente les spécifications détaillées du Standard d'Interopérabilité des Organismes de la Sphère Sociale [R1].

1.2. Relation avec d'autres documents

Ce document dérive et complète le Standard [R1]. Il est aussi prévu de le dériver en autant de document que d'application du standard. En l'occurrence, trois dérivations sont prévues au 1er février 2006 : RNIAM Portail-à-portail, RNIAM web service et Retraite portail-à-portail.

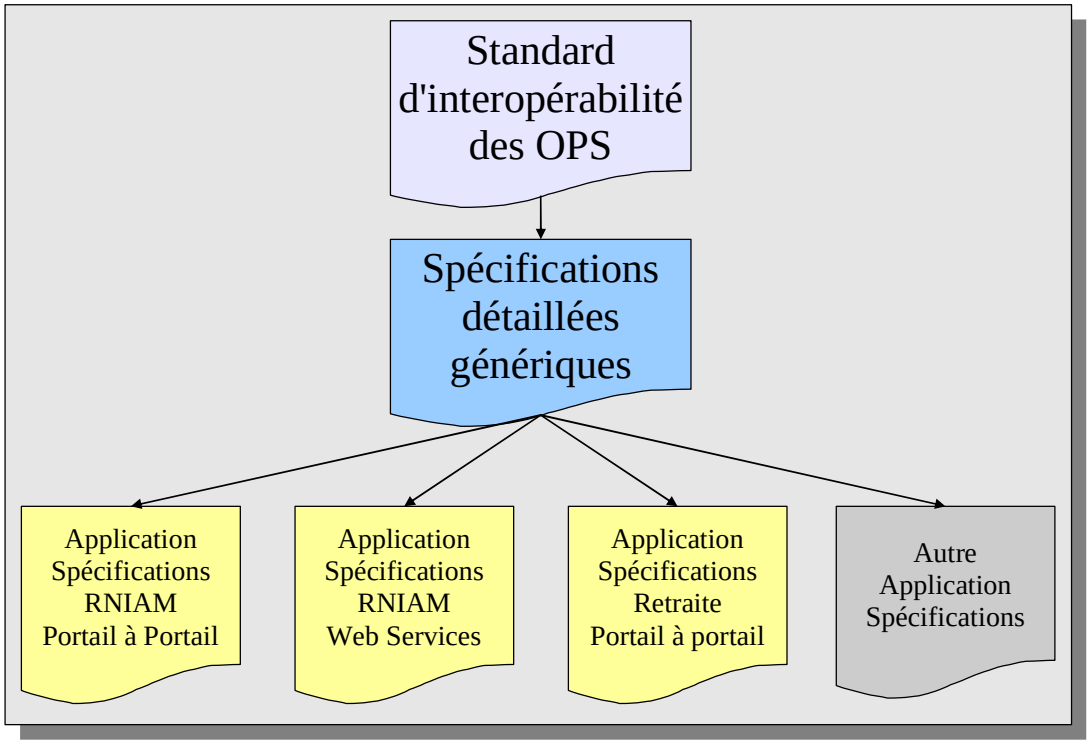


Figure 1 : relation avec d'autres documents

1.3. Organisation et structure du document

La structure du présent document est, en sus de la présente introduction, organisé comme suit :

- Le chapitre 2 – « Eléments d'architecture » définit le périmètre des spécifications et apporte des éclairages sur les contraintes d'implémentation du standard,
- Le chapitre 3 – « Lots à développer » regroupe les blocs fonctionnels à développer,
- Le chapitre 4 – « Lot 1 : Outils d'administration des accords » représente les spécifications détaillées du lot concernant les accords d'interopérabilité,

- Le chapitre 5 – « Lot 2 : Vecteur et proxy Organisme Client » représente les spécifications détaillées du lot concernant la création du vecteur d'identification et sa propagation du côté de l'Organisme Client,

- Le chapitre 6 – « Lot 3 : Vecteur et reverse proxy Organisme Fournisseur » représente les spécifications détaillées du lot concernant la réception et la manipulation du vecteur d'identification du côté de l'Organisme Fournisseur,

- Le chapitre 7 – « Lot 4 : Traces » représente les spécifications détaillées du lot concernant l'enregistrement et l'analyse des traces.

> ✍ *Dans la suite du document, les remarques et commentaires ON-X ne relevant pas des spécifications mais servant à éclairer ou étendre certains propos seront présentés dans le formatage texte courant : texte italique encadré de bleu.*

2. Eléments d'architecture

Une architecture fonctionnelle qui respecte le standard d'interopérabilité comprend plusieurs composants qui sont largement indépendants. L'implémentation des composants doit prendre en compte les besoins et contraintes de l'environnement existant au sein des organismes.

Dans ce chapitre les éléments d'architecture d'implémentation sont confrontés avec leurs environnements respectifs.

Dans plusieurs cas nous proposons plusieurs solutions pour la mise en place du standard, afin de ne pas imposer une complexité trop importante selon les contextes des organismes et permettre plus de flexibilité dans la mise en place du standard dans l'environnement local.

2.1. Périmètre du standard

Le schéma illustre la décomposition fonctionnelle de l'architecture. Pour certains éléments nous préconisons l'utilisation de technologies existantes (ebXML, Apache, OpenSAML), bien que toute implémentation alternative compatible soit acceptable.

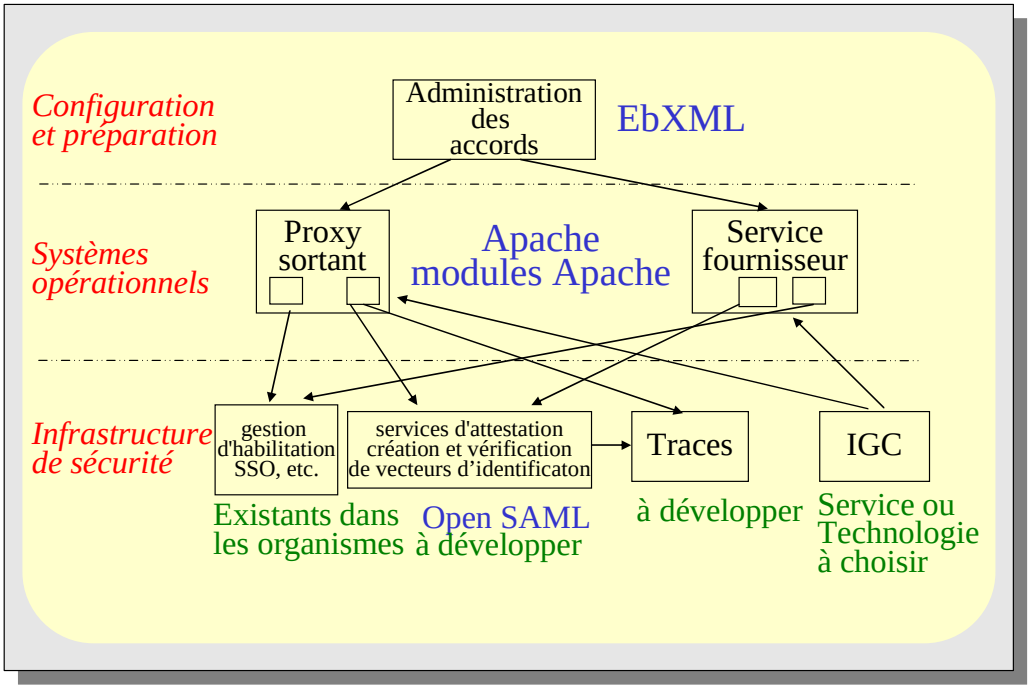


Figure 2 : Décomposition de l'architecture

> ✍ *Concernant le terme Proxy : le Module Proxy agit comme une passerelle au sens de l'adressage de service. Il permet à un client interne d'adresser un service tout en s'abstrayant de l'adressage réel (externe) de ce service. Mais, il rajoute des fonctionnalités, en particulier de sécurité, et joue de ce point de vue plus un rôle de proxy que de passerelle.*

2.1.1. Découpage fonctionnel modulaire

Une architecture fonctionnelle respectant le standard se décline autour des points suivants :

- ❑ L'administration des accords d'interopérabilité,
- ❑ Les techniques de gestion des certificats,
- ❑ L'interconnexion des réseaux,
- ❑ La manipulation des vecteurs d'identification,
- ❑ La gestion des traces.

En termes de blocs fonctionnels en vue d'une réalisation du standard, ces éléments sont réorganisés en quatre lots dans les chapitres suivants.

2.1.2. Eléments génériques et spécifiques

Chaque lot à développer comprend une liste de modules fonctionnels. Ces modules sont de deux ordres du point de vue des développements :

- ❑ Les modules dits génériques dont les fonctions et implémentations sont potentiellement applicables par tous les organismes quelle que soit le domaine applicatif ou les services,
- ❑ Les modules dits spécifiques qui se reposent sur les éléments spécifiques des applications ou services en jeu (exemple environnement RNIAM ou environnement Retraite). Ces modules dépendent donc fortement de l'environnement SI de l'organisme fournisseur.

Par ailleurs, des modules devront vraisemblablement être développés par les organismes (principalement clients) lorsque les spécificités de leur SI ne permettent pas une connexion directe aux applications/solutions développés dans le cadre du standard. Il s'agit principalement d'interfaces.

Dans la suite de ce document le code couleur suivant est utilisé pour les schémas :

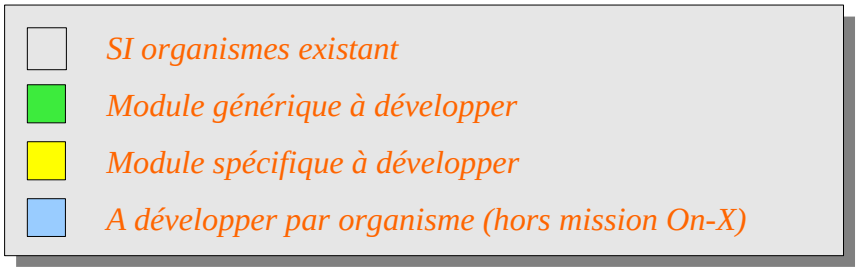


Figure 3 : code couleur des schémas

Le gris correspond à des éléments existants des systèmes d'information ou à des éléments externes au sujet exposé dans le schéma.

Le vert clair correspond aux modules génériques.

Le jaune correspond aux modules spécifiques.

Le bleu clair correspond aux éléments hors standard mais à développer (par exemple les applications utilisant le standard).

2.1.3. Boites à outils

La mise en œuvre des blocs fonctionnels décrits dans les spécifications détaillées doit répondre à une logique de boite à outils. En particulier, les implémentations proposées par les développeurs du standard devront permettre le plus possible le choix des organismes quant à l'utilisation ou non de ces blocs fonctionnels.

2.2. Eléments techniques représentant les accords

La mise en place d'échanges de données entre deux organismes fait l'effet d'un accord (au travers de la convention telle que définie dans le standard). Cet accord inclut une partie descriptive dans laquelle sont indiqués les paramètres techniques précis de l'accord d'échanges de données. Dans la suite de ce document, le terme « accord » fera référence aux éléments techniques des accords entre organismes.

La mise en place des accords inter-organismes se décompose en trois étapes :

- La publication des informations des clients et des services fournisseur,
- La mise en correspondance des informations fournies par les deux parties client et fournisseur (en soit : l'accord),
- Le déploiement de l'accord sur chacun des systèmes.

2.2.1. Flux de mise en place des services

Les accords (les éléments techniques de la convention) font l'objet d'une publication de la part du ou des Organismes Clients et du ou des Organismes Fournisseurs ; la combinaison des deux représentant l'accord d'interopérabilité.

Les flux de mise en place des services relèvent du domaine de l'organisation des échanges de données entre organismes.

> ✍ *En reprenant la terminologie d'ebXML (e-Business XML) selon l'OASIS, les publications correspondraient aux CPP (Collaboration Protocol Profile), l'accord d'interopérabilité correspondrait au CPA (Collaboration Protocol Agreement).*

2.2.2. Création des publications Organisme Client

La publication des informations de l'Organisme Client correspond à une structure de données contenant au moins les éléments ci-dessous. Ces données sont obligatoires ou optionnelles.

2.2.2.1. Définition des services visés

La liste des services visés est fournie sous forme d'URI au format indiqué au paragraphe *2.5.2 Dénomination de service*. Chaque entrée de cette liste peut-être accompagnée d'un descriptif sous forme de chaîne de caractères. Elément optionnel.

2.2.2.2. Définition d'une autorité de certification

En ce qui concerne les certificats numériques, un certificat d'autorité (de certification) doit être fourni pour permettre à l'Organisme Fournisseur de pouvoir authentifier les connexions de l'Organisme Client ainsi que réaliser et valider les signatures des assertions SAML. Elément obligatoire.

2.2.2.3. Définition d'administrateurs privilégiés

Un administrateur privilégié a pour tâche de gérer les aspects techniques relatifs aux accords (déploiement et mises à jour). Chaque administrateur privilégié est défini par un DN (Distinguished Name au sens LDAP). Chaque DN est utilisé en conjonction avec un certificat de l'autorité. Elément optionnel.

2.2.3. Création des publications Organisme Fournisseur

La publication des ressources de l'Organisme Fournisseur fait l'effet d'une structure de données contenant au moins les éléments ci-dessous. Ce sont les données qui font référence.

2.2.3.1. Définition des services et association PAGM/URI

Le fournisseur fournit une liste de services. A chaque service est associée une URI (voir l'adressage de service au paragraphe *2.5.2 Dénomination de service*). Pour chaque service une description sous forme de chaîne de caractères est fournie, pouvant être utilisée pour affichage par chaque application de l'Organisme Client. Elément obligatoire.

Pour chaque URI sera fournie une liste de PAGM. Elément obligatoire.

Il est possible de ne pas associer de PAGM à une URI afin d'indiquer un accès libre sans création de vecteur d'identification (par exemple pour un sous-répertoire d'images).

2.2.3.2. Contraintes d'accès aux services

Par PAGM et/ou par URI il peut y avoir des éléments supplémentaires utilisés par les organismes fournisseurs pour gérer des contraintes sur les droits au niveau applicatif. Cette partie est à rapprocher des règles d'attribution des PAGM telles que définies par l'Organisme Client. Elément optionnel.

Les contraintes sont exprimées sous les formes suivantes :

- présence ou non d'un attribut,
- valeur d'un attribut.

Les exemples d'attributs sont : indications géographiques, indications de localisation, indication d'appartenance à une organisation.

2.2.3.3. Définition d'une autorité de certification

En ce qui concerne les certificats numériques, un certificat d'autorité (de certification) doit être fourni pour permettre à l'Organisme Client de pouvoir authentifier l'Organisme Fournisseur lors des échanges. Elément obligatoire.

L'utilisation du terme « Autorité » est purement technique.

2.2.3.4. Définition d'administrateurs privilégiés

Un administrateur privilégié a pour tâche de gérer les aspects techniques relatifs aux accords (déploiement et mises à jour). Chaque administrateur privilégié est défini par un DN (Distinguished Name). Chaque DN est utilisé en conjonction avec un certificat. Elément optionnel.

2.2.4. Accord d'interopérabilité entre Organisme Client et Organisme Fournisseur

Il s'agit d'une structure de données reprenant les données des deux publications. Le résultat doit comprendre :

- La liste des services avec descriptions,
- La liste des PAGM,
- La liste des associations URI – PAGM,
- La liste des certificats utilisés,
- Le numéro de version valide du format de vecteur d'identification. Ce numéro de version permet de gérer les évolutions éventuelles du format du vecteur d'identification,
- La durée de vie minimale des traces.

2.2.5. Mise en place de l'accord

La mise en place de l'accord correspond au déploiement des données techniques concernant la configuration des systèmes d'exploitation :

- La base de données des PAGM : celle-ci doit contenir la dernière liste en date des associations PAGM – URI,
- La base d'habilitation : elle doit contenir l'association des identifiants locaux (utilisateurs, groupes, rôles,…) avec les PAGM,
- Les certificats : la mise à disposition des certificats pour l'authentification des communications entre les organismes ainsi que des certificats pour signature/validation des assertions SAML.

L'annexe technique de l'accord est un document formalisé permettant d'automatiser la création des éléments de configuration.

✍ *Les règles d'attribution des PAGM aux utilisateurs définies par les organismes clients ne font pas partie des éléments décrits dans les spécifications détaillées. De même pour l'attribution de profils applicatifs côté Organisme Fournisseur.*

✍ *Le choix de la technique CPP-CPA n'est pas imposé dans le cadre de ces spécifications. Néanmoins si ce choix n'est pas retenu par le constructeur/éditeur, il devra proposer une technique équivalente qui sera commune et à implémenter par les organismes.*

2.3. Protections techniques par des moyens cryptographiques

L'échange des transactions doit respecter plusieurs besoins de sécurité. Pour respecter certains besoins, des moyens cryptographiques sont utilisés :

- Le vecteur d'identification est signé numériquement. La signature numérique est basée sur la cryptographie asymétrique, utilisant les bi-clefs numériques {clef publique, clef privée}. Voir le document [R2].

- Par ailleurs, les communications entre organismes sont chiffrées par la technique SSL.

2.3.1. Utilisation des bi-clés/certificats

Cas de la signature : ce paragraphe est une illustration du fonctionnement à but de compréhension.

L'Organisme Client possède la bi-clef {clef publique, clef privée}, la clef publique est fournie à l'Organisme Fournisseur. La clef privée sert à l'Organisme Client lors de la création de la signature du vecteur d'identification, la clef publique sert à l'Organisme Fournisseur lors de la reconnaissance de la signature.

Dans cet exemple, la mise en place d'un tel mécanisme entraîne que :

- Chaque Organisme Client possède au moins une bi-clef,

- Chaque service d'un Organisme Fournisseur doit connaître toutes les clefs publiques des Organismes Clients pour vérifier les signatures,

- La mise à jour d'une bi-clef chez un Organisme Client implique une mise à jour de la connaissance des clefs publiques pour tous les services des Organismes Fournisseurs.

De manière pratique, les utilisations applicatives usuelles se basent sur les certificats numériques plutôt qu'uniquement sur les bi-clefs (ainsi en est-il des systèmes SSL/TLS, XML-DSIG, S/MIME,…).

Pour répondre à ces usages, chaque organisme devra désigner au moins une Autorité de Certification produisant les certificats.

✍ *Il n'y a pas de lien entre la validité d'un certificat et la durée prévue des accords. L'un et l'autre sont renouvelables indépendamment.*

✍ *Le choix du type de gestion de clés n'entre pas dans les spécifications du standard (il concerne l'organisation interne de chaque organisme vis à vis de la cryptographie). Néanmoins, l'application du standard implique pour les organismes de mettre en œuvre les clés pour la signature des vecteurs d'identification et le chiffrement des échanges, et par conséquent de protéger ces clés.*

Nous recommandons de mettre en place une procédure pour permettre d'invalider l'utilisation d'un certificat. Cela comprend un moyen local pour invalider un certificat d'un partenaire, et une procédure de notification permettant à un organisme d'invalider ces certificats.

2.3.2. Authenticité du vecteur d'identification

Le vecteur d'identification sera signé numériquement. Les organismes doivent donc disposer au moins d'un certificat numérique X509 à cette fin.

2.3.3. Authenticité et confidentialité des échanges

La communication entre organismes est protégée par le protocole TLS. Pour une authentification mutuelle de serveur et de client chaque partenaire dispose au moins d'un certificat numérique X509.

La confidentialité est assurée par la sélection d'un protocole de chiffrement fort tel qu'AES pour la session TLS.

2.4. Transmission du vecteur d'identité

L'inclusion du vecteur d'identification se fait dans deux contextes :

- Le contexte portail-à-portail,
- Le contexte web service.

2.4.1. Transmission portail à portail

En extension au format HTTP, l'élément d'entête **X-IOPS-Vecteur-Identification** est défini. Il contient le vecteur d'identification en tant qu'assertion SAML signée, formatée en Base-64. Cette extension devra être effectuée sous forme d'un paramètre du serveur http qui doit la prendre en compte, la valeur par défaut de l'extension est « X-IOPS-Vecteur-Identification ».

✍ *Le standard prévoit la transmission en enrichissant la requête HTTP par une autre entête. Très récemment, une proposition d'insertion d'assertion SAML à l'intérieur de la couche SSL par une extension était apparue au sein de l'IETF. Cette approche est très prometteuse, car cela évite la spécification du nom de l'entête.*

✍ *Cette approche est l'inverse de ce qui existe actuellement en termes d'implémentation de certificat X.509 client par des systèmes du type 'Appliance SSL', où un certificat client est transformé en en-tête HTTP.*

✍ *En conclusion, une approche du second type est préférable dans le contexte actuel.*

2.4.2. Web service sécurisé et SOAP

SOAP permet, par l'extension D-SIG, de signer les échanges entre applications Web Service.

L'intégration du standard d'interopérabilité par un organisme dans une architecture préexistante d'échange entre applications utilisant SOAP et D-SIG doit se faire de manière transparente en respectant l'utilisation de WS Sécurisé.

La façon d'insérer le vecteur d'identification dans la requête n'a pas en soit d'impact sur le standard. Toutefois, pour permettre le développement de composants génériques, le format suivant, se basant fortement sur les spécifications de Liberty Alliance (voir le document [R3]), est recommandé :

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/ soap/envelope/"
xmlns:sb="urn:liberty:sb:200 3-08"
xmlns:pp="urn:liberty:id-sis-pp:2003-0 8"
xmlns:sec="urn:liberty:sec:2003-08">

<s:Header>
        <sb:Correlation s:mustUnderstand="1"
                id="A13454...245"
                actor="http://schemas.../next"
                messageID="uuid:efefefef-aaaa-ffff-ccc c-eeeeffffbbbb"
                timestamp="2112-03-15T11:12:12Z"/>

        <wsse:Security>

                <!-- description du vecteur d'identification au format SAML -->
                <saml:Assertion
                        xmlns:saml="urn:oasis:names:tc:SAML:1.0:assert ion"
                        MajorVersion="1" MinorVersion="0"
                        AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0 NKU="
                        Issuer="idp.example.com"
                        IssueInstant="2004-04-01T16:58:33.173Z">

                        …

                        <ds:Signature>...</ds:Signature>

                </saml:Assertion>

                <ds:Signature>
                        <ds:SignedInfo>

                                <ds:Reference URI="#A13454...245">
                                        <ds:DigestMethod      Algorithm="http://www.w3.org/2000/09/xmldsig#s
ha1"/>
                                        <ds:DigestValue>GyGsF0Pi4xPU...</ds:Dig estValue>
                                </ds:Reference>
                                <ds:Reference URI="#MsgBody">
```

```
                                    <ds:DigestMethod          Algorithm="http://www.w3.org/2000/09/xmldsi
g#sha1"/>
                                    <ds:DigestValue>YgGfS0pi56pu...</ds: DigestValue>
                              </ds:Reference>
                        </ds:SignedInfo>
                        <ds:KeyInfo>
                              <wsse:SecurityTokenReference>
                                    <wsse:Reference URI="#2sxJu9g/vvLG9sAN9bKp/8q0NKU="
                                          ValueType="saml:Assertion" />
                              </wsse:SecurityTokenReference>
                        </ds:KeyInfo>
                        <ds:SignatureValue>
                              HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhwBdFNDElgscSXZ5Ekw= =
                        </ds:SignatureValue>
                  </ds:Signature>
            </wsse:Security>
</s:Header>
<s:Body wsu:Id="MsgBody">
</s:Body>
</s:Envelope>
```

2.5. Interconnexion réseau, adressage et présentation de service

L'accession à un service de l'Organisme Fournisseur à travers un portail sortant de l'Organisme Client nécessite de distinguer proprement ces deux points d'accès. En outre, le portail sortant propose une fonctionnalité de présentation de service propre à chaque Organisme Client.

2.5.1. Interconnexion réseau

L'interconnexion des réseaux ne rentre pas dans le cadre du standard, en dehors d'une contrainte évidente : les services fournisseurs doivent être visibles par les portails/proxys des organismes clients. En d'autres termes, les proxys clients doivent disposer d'une adresse IP au moins visible par le système reverse proxy du fournisseur et vice-versa.

> ✍ *Ceci ne signifie en aucune façon que les plans d'adressage plus large entre les organismes doivent être mis en commun.*

En prenant en compte le modèle proxy-reverse proxy défini par le standard, l'adressage d'un service d'un Organisme Fournisseur par un client pourrait, par exemple, se faire en trois grandes zones :

- Adressage du service par le client selon le plan d'adressage interne à l'Organisme Client,
- Adressage du service, après translation d'adresse par l'Organisme Client, selon un plan d'adressage publié dans l'accord d'interopérabilité par l'Organisme Fournisseur,
- Adressage du service, après translation d'adresse par l'Organisme Fournisseur, selon le plan d'adressage interne à l'Organisme Fournisseur

La translation d'adresse se fait à l'intérieur des proxy et reverse proxy au niveau applicatif.

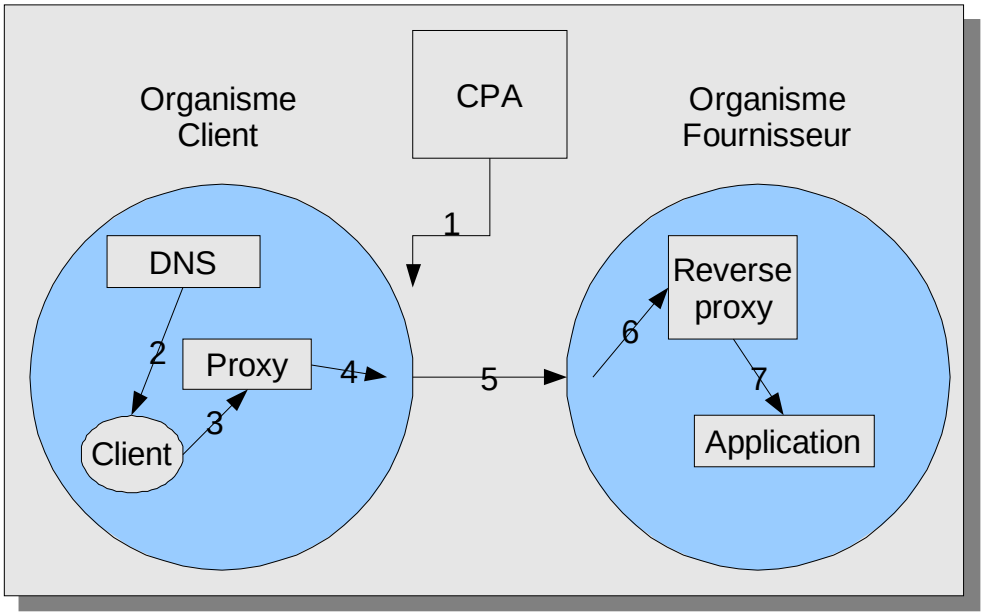


Figure 4 : Principe de communication entre services

Selon cette figure, l'adressage d'un service chez l'Organisme Fournisseur par un client suit ces étapes :

1 - L'accord d'interopérabilité (ici CPA selon ebXML) indique quelle est l'adresse affectée au service par l'Organisme Fournisseur,

2 - A la demande du Client, le DNS de l'Organisme Client fournit une adresse interne à l'Organisme Client,

3 - Le Client envoie une requête à cette adresse, qui est routée (selon le routage interne à l'Organisme Client) vers le Proxy, lequel ajoute les informations d'autorisation nécessaire et envoie vers la passerelle externe la requête,

4 - La passerelle externe effectue une translation d'adresse entre l'adresse interne affectée au service et celle affectée (adresse publique) par le CPA,

5 - Routage vers le point d'entrée de l'Organisme Fournisseur, une nouvelle translation d'adresse remplace l'adresse publique (CPA) par une adresse interne à l'Organisme Fournisseur,

6 - Le routage interne de l'Organisme Fournisseur fait transiter la requête à travers le Reverse Proxy ou le frontal du service visé,

7 - Le Reverse Proxy ou le frontal du service visé effectue les vérifications nécessaires, transforment les vecteurs d'identification en fonction des besoins du service visé.

Bien que ce principe ne soit en rien imposé par le standard, il permet de montrer qu'une adresse unique est suffisante au client pour atteindre le service de l'Organisme Fournisseur.

2.5.2. Dénomination de service

Le standard ne spécifie pas de convention de dénomination (DNS) pour les services visés par les accords d'interopérabilité. De manière générale, comme indiqué au paragraphe précédent, l'adressage de service

ne nécessite qu'une adresse IP. Toutefois, lors de la mise en place d'accords entre organismes, pour assurer la facilité l'installation et la maintenance des systèmes, il est demandé de suivre les règles suivantes :

- Un Organisme Fournisseur doit pouvoir gérer l'ensemble de ses services de manière indépendante du nom de ces services et en particulier la répartition sur des machines différentes d'une manière indépendante. Par exemple, le changement de la répartition de services sur plusieurs serveurs de l'Organisme Fournisseur ne doit pas changer le nom du service. Cela implique donc un nom DNS par service (pas nécessairement plusieurs adresses),

- Un service visé est nommé par un nom DNS de la forme **service**.**nom-de-domaine-de-l-organisme**. Par exemple, dans le cas du RNIAM, le nom de service peut être de la forme **rniam.cnav.fr**,

- Dans un organisme client, un service est représenté par un portail sortant. Afin d'éviter une infrastructure complexe comme un DNS spécifique ou des plaques réseau avec adressage identique pour le portail, le portail sortant doit être capable au niveau applicatif de récrire des URL, donc les noms de services. Ainsi le portail peut être accédé par l'organisme client sous un autre nom géré par lui-même. Exemple : **portail-rniam.cnamts.fr**.

Le tableau suivant précise les éléments d'adressage, en particulier en ce qui concerne la notion de service :

Nom	**Définition/Commentaires**
Service	Groupe cohérent de fonctions mis à disposition de l'Organisme Client par l'Organisme Fournisseur dans le cadre de l'échange. Le service est nommé par un nom DNS, par exemple **rniam.cnav.fr**.
Service visé	Le service visé se réfère à la fois au service lui même ainsi qu'aux sous-groupes de fonctions de ce service proposé par l'Organisme Fournisseur dans les accords d'interopérabilité. Ainsi, le service visé est nommé par un nom DNS s'il s'agit du groupe complet (par exemple **rniam.cnav.**fr) ou par le même nom DNS suivi d'un préfixe de chemin s'il s'agit d'un sous-groupe du service (par exemple **rniam.cnav.fr/images** où **/images** est le préfixe de chemin). C'est cet élément que l'on retrouve dans le vecteur d'identification.
Adresse locale Organisme client	Le service visé doit être connu par l'application cliente (le navigateur ou l'application web service) par un nom local, ce qui simplifie l'administration de DNS au sein de l'Organisme Client. Par exemple le service **rniam.cnav.fr** peut être visé par l'application cliente avec le nom **rniam-portail.cnamts.fr**, le portail de l'Organisme Client se chargera alors de transcrire l'adresse locale en adresse externe.
Adresse externe	Pour un service il s'agit du nom tel qu'il est publié dans les accords d'interopérabilité.

Service publié	Il s'agit du service tel qu'il est publié dans les accords ainsi que de l'ensemble des sous-groupes du service publiés de même dans les accords. Si un sous-groupe de service n'est pas publié, il ne peut pas être un service visé.
URL visée	L'URL complète représentant aussi bien une fonction ou une ressource particulière d'un service que le portail accueillant plusieurs services. Il est important de ne pas confondre service visé et URL visée.

Dans le reste du document il n'est fait référence qu'au service lui-même. Cela inclura autant le service en tant que tel que les sous-groupes du service.

> ✍ *La différence faite ici entre le service et ses sous-groupes est importante du point de vue nom-de-service : l'adressage du service ne devant pas imposer chez l'Organisme Fournisseur une implémentation (en particulier matérielle) de l'accès au service. Néanmoins, du point de vue du standard, cette différence n'a pas d'impact.*

En exemple de dénomination de service, un Organisme Fournisseur (nommé fournisseur) met à disposition un service (nommé service) composé de, au moins, une fonction (nommée fonction1). Il a alors le choix lors de la publication (la convention) de définir ce service comme :

- Un unique service (**service.fournisseur**) dont tous les PAGM associés doivent être transmis à toute requête dont le nom d'hôte de l'URL est **service.fournisseur**. Il n'y a alors qu'un seul service publié,

- Plusieurs services indépendants (**service.fournisseur** et **fonction1.fournisseur**) : du point de vue d'un Organisme Client le cas est identique au cas précédent à l'exception des fonctions qui sont ventilées sur deux services distincts. Il y a alors deux services publiés,

- Un unique service (**service.fournisseur**) et un sous groupe (**service.fournisseur/fonction1**). Dans ce cas les PAGM associés dans la convention à la fonction1 doivent être transmis à toute requête dont le nom d'hôte de l'URL est **service.fournisseur** et le préfixe de chemin est **/fonction1**. Toutes les autres requêtes dont le nom d'hôte de l'URL est **service.fournisseur** doivent être accompagnées des PAGM associés dans la convention au service lui même. Il y a alors aussi deux services publiés mais l'un (**fonction1**) sert d'exception en termes d'attribution de PAGM à l'autre service (**service**). Typiquement, un sous-groupe de service peut permettre d'accéder aux images du service en étant associé à aucun PAGM (« service gratuit »).

L'Organisme Fournisseur décide, par exemple, de publier selon le troisième cas (service.fournisseur et service.fournisseur/fonction1). Du point de vue du standard, l'Organisme Client peut donc *viser* les deux services qu'il trouve dans la convention : **service.fournisseur** et **service.fournisseur/fonction1**. Ce sont les noms que son proxy doit utiliser. Dans son organisation interne, l'Organisme Client utilise un nommage local pour accéder aux services, par exemple **service-fournisseur.client** et **service-fournisseur.client/fonction1**. Le proxy se charge alors de transcrire les noms locaux en noms externes.

2.5.3. Présentation de service

Le standard prévoit la possibilité de communiquer des éléments textuels pour la présentation dans des menus d'un portail. L'implémentation portail doit être capable de prendre en compte ces éléments.

La présentation de menus doit aussi prendre en compte les besoins PAGM pour chaque service afin de ne présenter aux utilisateurs que les services accessibles selon leur profil.

2.6. Vecteur d'identification

Le vecteur d'identification est une structure de données qui décrit de façon abstraite les éléments permettant de transporter entre les organismes les informations d'autorisation nécessaires pour le standard. Le standard spécifie l'encodage du vecteur en utilisant des assertions SAML.

Dans le modèle Portail-à-Portail autant que dans le modèle Web Service le vecteur d'identification est construit de manière générique.

2.6.1. Eléments du vecteur d'identification

L'objet du présent paragraphe est d'identifier l'origine des éléments rentrant dans la composition du vecteur d'identification.

Pour reprendre la description faite dans [R1] (*Standard d'interopérabilité inter-organismes*) le vecteur d'identification comprend les éléments suivants :

N°	Elément du vecteur d'identification
1	Numéro de version pour le format du vecteur d'identification
2	Identifiant de vecteur unique pour tous les organismes
3	Identifiant de l'Organisme Client
4	Identifiant du demandeur ou de l'application de départ, éventuellement dépersonnalisé
5	Date de création
6	Durée de vie de l'habilitation
7	Identifiant de l'Organisme Fournisseur de service
8	Service visé (sous forme d'URI sans partie locale)
9	Liste des PAGM valides pour le demandeur
10	Autres attributs (indication géographique, localisation, niveau de sécurité,…) - facultatif
11	Niveau d'authentification initiale (moyen ou niveau de moyen avec lequel l'authentification initiale du demandeur est réalisée) - facultatif
12	Signature numérique délivrée par l'organisme de départ

Ce vecteur est construit, ainsi que défini par le standard, dans le but d'assurer l'authenticité de l'attribution par l'Organisme Client des autorisations (les PAGM). Du point de vue de l'Organisme Fournisseur l'authenticité est garantie car :

- le vecteur est signé (avec les éléments d'identification de l'Organisme Client, la durée de validité et l'identifiant du vecteur) ce qui garantit que le vecteur a été créé par l'Organisme Client,

- le vecteur lui-même est transmis sur une liaison dont l'extrémité est authentifiée garantissant que le vecteur provient bien d'un Organisme Client.

2.6.1.1. Eléments extraits de l'accord d'interopérabilité

- 1 : ceci est un élément d'évolutivité,
- 3 : représentation d'un DN,
- 7 : représentation d'un DN,
- 8 : voir le paragraphe 2.5.2.

2.6.1.2. Eléments extraits du Système Local uniquement

- 2 : élément unique de référence, en particulier utile/nécessaire pour l'analyse des traces,
- 5 : horodatage de la dernière modification du vecteur d'identification,
- 4 : formatage commun avec l'Organisme Fournisseur. L'Organisme Fournisseur s'en servira pour le traçage mais seul l'Organisme Client connaît sa signification,
- 10 : tout attribut que l'Organisme Client souhaite ou doit fournir,
- 11 : indication permettant de montrer que l'attribution du ou des PAGM s'est faite en conformité avec l'accord d'interopérabilité.

Concernant les attributs supplémentaires (entrée 10) : le vecteur d'identification ayant une vocation sécuritaire, les attributs du vecteur ne doivent être que des attributs de sécurité.

2.6.1.3. Eléments déduits d'une combinaison entre l'accord d'interopérabilité et le Système Local

- 9 : la liste de PAGM est générée en fonction des éléments 1, 2, 4, 5 et 8.

2.6.1.4. Eléments obtenus par combinaison entre l'accord d'interopérabilité et les moyens techniques de signature

- 6 : normalement de courte durée (exemple quelques minutes), elle ne peut excéder la durée de validité des moyens techniques de signature (par exemple la date de péremption d'un certificat numérique),
- 12 : la signature nécessite, d'une manière ou d'une autre, un moyen technique de signature.

2.6.2. Schéma de construction du vecteur d'identification et de l'assertion SAML

Nous distinguons deux modes possibles de construction,

- **le mode 1** où le client (navigateur web dans le cas portail à portail) envoie sa requête sans autorisation. Ce mode s'applique aux modèles Portail-à-Portail et Web Service,

- **le mode 2** où le client insère une autorisation dans sa requête avant l'envoi. Ce mode s'applique au modèle Web Service uniquement (où l'application joue le rôle d'un proxy évolué).

Dans le mode 1 le proxy se chargera d'insérer la bonne assertion SAML, dans le mode 2, il aura à opérer la validation de la signature. Par contre le mode 2 étend les possibilités du standard en permettant, par exemple, l'affichage dynamique, dans le cadre d'un portail interne, des services qu'un utilisateur peut atteindre en fonction des autorisations qu'il est en droit d'obtenir et l'utilisation de signatures par l'application appelante.

Les deux modes doivent être implémentables dans le cadre de la mise en place du standard (et par conséquent des développements des briques fonctionnelles permettant sa mise en place). Les deux modes sont mis en œuvre par les mêmes blocs fonctionnels.

2.6.2.1. Mode de construction 1

Ce mode est utilisable dans les modèles Portail-à-Portail et Web Service.

L'application (dans le modèle Portail-à-Portail il s'agit du navigateur) transmet la requête au proxy et en fonction de cette requête le proxy se charge de la construction du vecteur d'identification en deux étapes (les chemins 3 et 4) : récupération d'une structure contenant les informations du vecteur d'identification puis formatage en assertion SAML et signature de l'assertion.

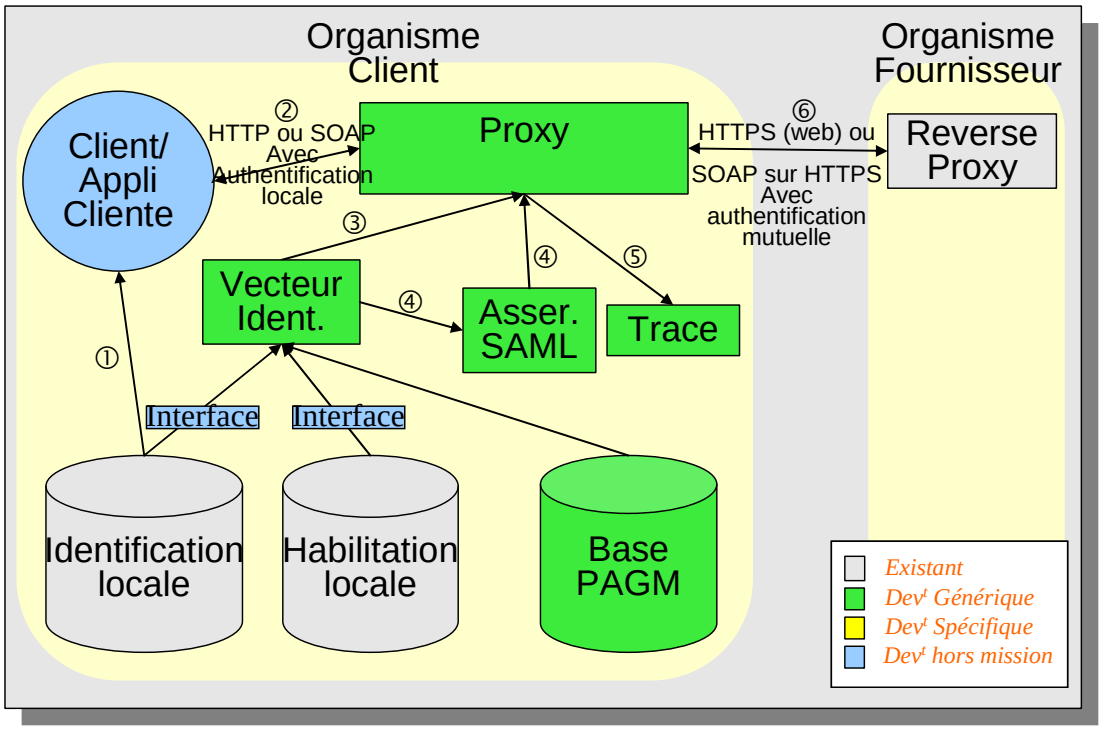


Figure 5 : Le proxy construit le vecteur d'identification

Dans la figure ci-dessus le module vecteur d'identification accède aux données locales par l'intermédiaire d'interfaces normalisées (telles que LDAP ou un système AAA – Authentication-Authorisation-Accounting pour Authentification-Habilitation-Traçabilité). Ces interfaces sont décrites dans le lot 2 (voir chapitres suivants).

Si ces interfaces n'existent pas au sein du SI d'un organisme client, une implémentation d'une interface normalisée devra être ajoutée au système local (les deux boites bleues « Interface »). A noter que l'interface vers l'Identification Locale a deux objectifs : un objectif de vérification de données et un objectif de récupération éventuelle d'informations supplémentaires.

> ✎ *Dans le cadre de la rédaction des spécifications détaillées objets de la présente mission, l'objectif est d'identifier toutes les interfaces nécessaires aux organismes clients présents autour de la table, afin de les intégrer au lot 2. Cette identification est réalisée dans le cadre des réunions avec chaque organisme client [action en cours].*

2.6.2.2. Mode de construction applicatif selon le modèle Web Service

L'application cliente est ici responsable de la récupération des vecteurs d'identification potentiels que l'utilisateur peut obtenir (chemins 2 et 3). Lors de la requête réelle (*flèches bleu foncé*) le proxy se charge de vérifier que le vecteur d'identification passé est valide (chemin 5).

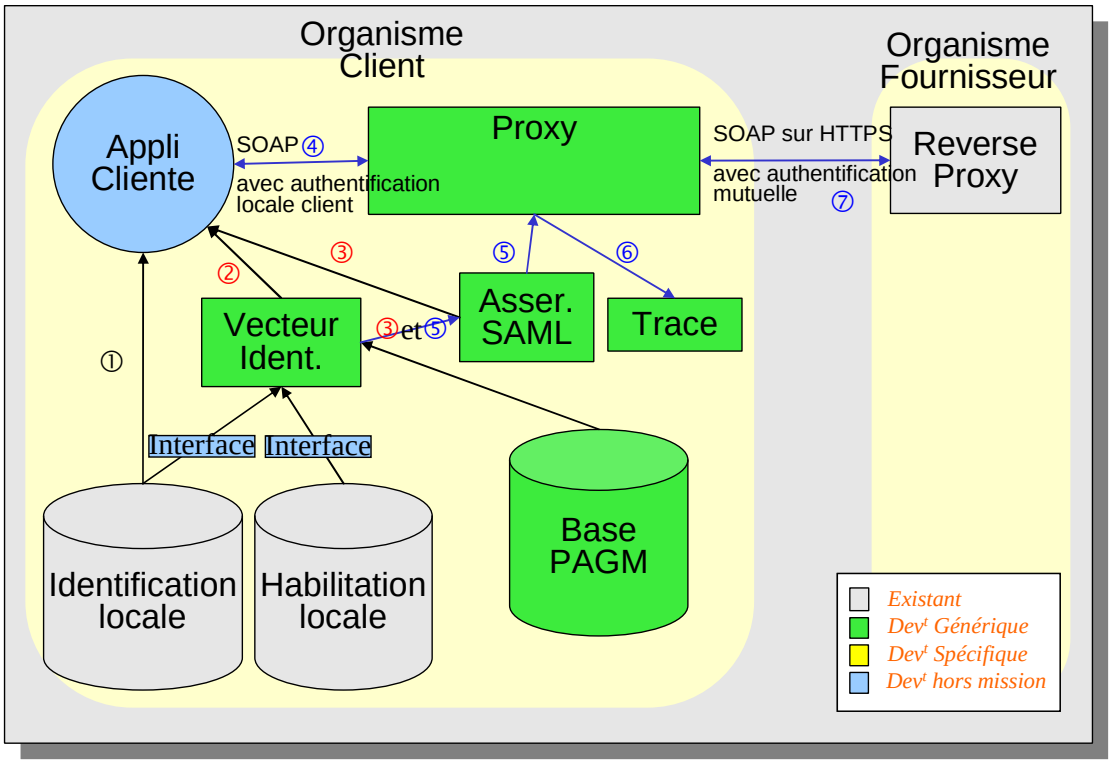


Figure 6 : L'application construit son vecteur d'identification

De même que pour la figure précédente, la figure ci-dessus présente aussi les interfaces (les deux boites bleues « Interface ») normalisées vers les données locales pour utilisation par le module vecteur d'identification. En d'autres termes l'application cliente ne fait pas partie du périmètre de confiance.

2.7. Gestion de sessions en mode portail

L'utilisation du standard ne prévoit pas l'utilisation de session au niveau applicatif à des fins d'authentification et leur gestion avec le client à travers des cookies (une technique souvent utilisée par des systèmes à base de mot de passe). La nature de l'utilisation des proxys en chaîne interdit cette technique de bout en bout, en particulier du point de vue du fournisseur. Toutefois, si le fournisseur doit

impérativement utiliser une technique de cookie de session (malgré le fait que chaque requête est accompagnée de son vecteur d'identification), celle-ci peut être simulée par son Reverse-proxy. De manière générale :

- Un système d'authentification peut utiliser cette technique entre le proxy et le client et doit se protéger contre l'utilisation de cookie venant du système du fournisseur. L'organisme client ne peut pas utiliser l'affichage des pages web du fournisseur pour intégrer une URL de « logout ».

- Le fournisseur ne peut pas compter sur le système client pour gérer des cookies ; il doit impérativement les gérer par des reverse proxy. En outre, un système de fournisseur peut

    - se permettre de créer des sessions pour chaque requête, car il n'y a plus d'interaction avec l'utilisateur,

    - ou encore, pour des raisons de performance, peut maintenir un cache de cookies au niveau des reverse proxy.

2.8. Traces

Les traces, de manière générale, sont de deux ordres : les traces de journalisation à but d'audit et les traces de fonctionnement à but de surveillance technique. Le standard impose les traces de journalisation, en tant que traces génériques, comme moyen de contrôle à-posteriori pour maintenir la confiance inter-organismes.

2.8.1. Traces de journalisation

Le standard indique, parmi ses principes de base, que toute création de vecteur d'identification doit être tracée afin de permettre le contrôle à posteriori. Cela correspond à la possibilité de chacun des organismes de conserver, à toutes fins utiles, les informations d'historique du déroulement des échanges avec les autres organismes.

Chaque accord d'interopérabilité listera un certain nombre d'informations/traces à conserver par chacun des organismes, ainsi que la durée de cette conservation. Cette liste a pour objet de compléter les traces jugées nécessaires (c'est à dire en plus du vecteur d'identification), par exemple en fonction de contraintes légales particulières, ainsi que le cadre d'utilisation de ces traces.

Les traces de journalisation, dont la conservation est obligatoire dans le cadre de la mise en place de l'interopérabilité des organismes, proviendront :

- des blocs techniques objets des présentes spécifications détaillées,

- des blocs techniques propres à chaque organisme.

Les présentes spécifications détaillées décrivent :

- la nature des traces de journalisation propres aux modules génériques et spécifiques objet des présentes spécifications détaillées,

- les conditions de collecte et stockage de ces traces, avec les mécanismes de sécurité associés au stockage,

- l'interface de communication sortante par rapport au module « traces » du standard permettant à un organisme d'exporter des traces des modules objets des spécifications détaillées vers un outil de gestion de trace préexistant au sein de l'organisme,

- l'implémentation doit disposer d'un outil de visualisation et traitement des traces de journalisation (exemple d'une interface web conviviale accessible pour des surveillants et auditeurs).

La fonction de traçage décrite dans ce document est, au sein d'un système d'information donné, un des éléments de l'ensemble des traces de ce système. Ainsi, si un vecteur d'identification est tracé, l'identifiant du demandeur, qui est une donnée relative (sur le long terme cet identifiant peut ne plus exister ou être modifié ou affecté à un autre demandeur), peut-être rapproché d'autres traces de journalisation du système d'information indiquant la signification de cet identifiant.

> ✍ *Les traces de journalisation propres aux blocs techniques hors spectre des présentes spécifications détaillées ne seront pas décrites. Elles devront être listées dans le cadre de la mise en place des accords d'interopérabilité entre organismes.*

2.8.2. Traces techniques

Les traces techniques (ou traces de fonctionnement à but de surveillance technique) concerne le fonctionnement interne des implémentations du standard. Bien que ces traces ne soient pas imposées par le standard lui-même, les besoins de surveillance des systèmes d'information existants nécessitent leur présence et leur compatibilité à leur contexte d'exploitation.

Ces traces peuvent être utilisées dans des opérations de supervision, pour la création de statistiques ou pour l'analyse de dysfonctionnements, etc.

Il convient, lors de l'implémentation, d'utiliser les interfaces de traçage prévues par chaque contexte d'implémentation. Par exemple, l'API de traçage du serveur Apache, ou Log4J pour un contexte Java.

Les traces techniques ne seront pas décrites dans les présentes spécifications détaillées, parce que fournies par les briques techniques mises en places par le « constructeur » de la solution.

2.9. Structures applicatives Web Service

Un point particulier des applications Web Service doit être pris en compte : le standard doit pouvoir s'intégrer aux différentes structures applicatives Web Service que les organismes sont ou seront amenés à mettre en œuvre.

Ceci conduit à deux typologies d'échanges :

- Les structures synchrones et asynchrones,

- Les structures simples (une application échange avec une autre application) ou complexes (la structure applicative est un graphe où plusieurs applications échangent entre elles).

L'impact de ces différentes structures se situe au niveau de la manipulation des vecteurs d'identification et de la gestion des réponses par les applications elles-mêmes.

Le cas d'une structure simple et synchrone est évident. L'Organisme Client crée le vecteur d'identification et le transmet à l'Organisme Fournisseur ; l'Organisme Fournisseur vérifie le vecteur d'identification et

renvoie la réponse adéquate. La réalisation du standard repose donc dans la mise en place d'un côté des fonctions de l'Organisme Client et de l'autre côté des fonctions de l'Organisme Fournisseur.

Les autres cas d'échanges reposent sur les structures usuelles asynchrones et complexes :

- Mécanisme avec résultat par demande (de type polling),
- Service de notification de réponse.

2.9.1. Structure applicative Web Service avec résultat par demande

Dans le cadre de ce mécanisme, le standard est réalisé de la même manière que pour le cas des structures synchrones simples. Ce cas peut être réalisé par exemple en 3 Web Services pour :

- l'envoi d'une requête,
- l'interrogation de la liste des réponses,
- la récupération d'une réponse.

Les deux derniers cas sont représentés ensemble dans le schéma, car ils ont les mêmes caractéristiques d'interfaçage avec les systèmes back-end.

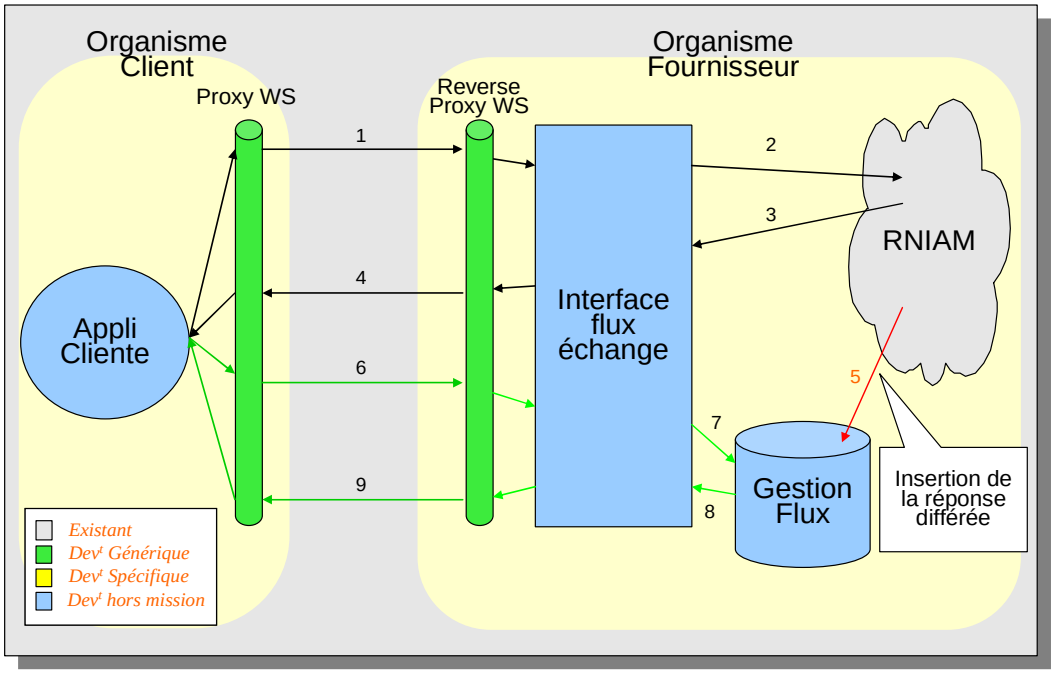
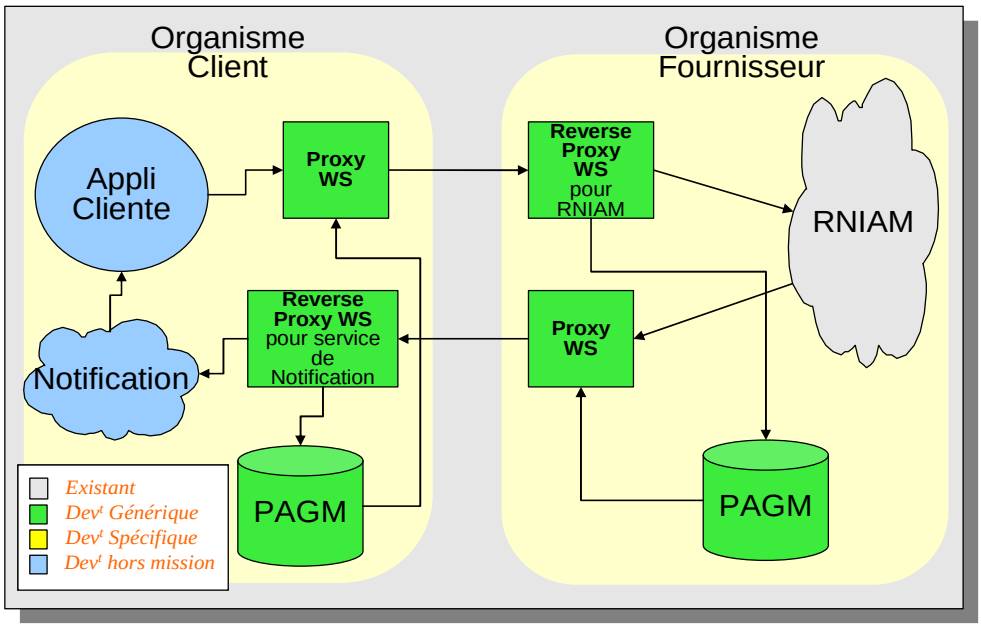


Figure 7 : Echange web service type messagerie avec RNIAM

2.9.2. Structure applicative Web Service avec service de notification

Dans le cadre d'un service de notification de réponse, l'Organisme Client devient lui aussi fournisseur d'un service, l'Organisme Fournisseur devenant client de ce service. Le standard est donc réalisé en mettant en place tant du côté Organisme Client que du côté Organisme Fournisseur des fonctions de l'Organisme Client et des fonctions de l'Organisme Fournisseur :



563 **Figure 8 : Echange web service type messagerie avec RNIAM**

564 ✍ *La description ou la mise en œuvre des mécanismes de messagerie ou des services de notification*
565 *n'entre pas dans le cadre de la prestation. Il est par contre nécessaire d'indiquer comment le*
566 *standard est réalisé en fonction des diverses structures applicatives que chaque organisme pourrait*
567 *mettre en œuvre.*

568 ✍ *L'organisation des web services (en mode asynchrone ou workflows plus avancés) n'a pas d'impact*
569 *sur l'implémentation du standard car ce dernier ne concerne que l'interface entre organismes.*
570 *Néanmoins le cas de WS avec notification est plus complexe au plan i) de l'organisation et ii) du*
571 *service à fournir.*

3. Lots à développer

Les développements seront réalisés à travers quatre grands lots :

- Administration des accords, concernant Organismes clients et Organismes fournisseurs
- Vecteurs et proxy Organismes Clients, qui concerne la création et l'insertion des vecteurs d'identification dans les requêtes sortantes,
- Vecteurs et reverse-proxy Organismes Fournisseurs, qui concerne l'interception, l'analyse et la validation des vecteurs d'identifications dans les requêtes entrantes,
- Traces, servant à tracer les opérations d'insertion et d'interception des vecteurs d'identification, concernant Organismes clients et Organismes fournisseurs.

Ce découpage en lot respecte une logique fonctionnelle mais n'impose en rien le découpage et l'implémentation à définir par le constructeur/éditeur. Ainsi, un ou plusieurs modules fonctionnels de ces lots peuvent très bien être implémentés en un ou plusieurs modules logiciels indifféremment.

Le constructeur/éditeur s'attachera cependant à respecter le principe de « boîte à outils » exposé précédemment.

3.1. Lot 1 : Administration des accords

Les outils d'administration des accords ont pour objectif de fournir un moyen d'alimenter les autres modules en éléments de configuration (liste de PAGM, URL, certificats,…) de façon automatisée. Il s'agit des éléments fonctionnels suivants :

- Outil de publication Organisme Client, il permet de récapituler dans un format d'échange normalisé les besoins d'un Organisme Client,
- Outil de publication Organisme Fournisseur, il permet de récapituler dans un format d'échange normalisé les besoins d'un Organisme Fournisseur,
- Outil de création des accords, il met en commun les publications d'un Organisme Fournisseur et d'un Organisme Client afin de créer l'annexe technique d'une convention d'interopérabilité,
- Outil de mise en œuvre des accords, il utilise l'accord (l'annexe technique à la convention) pour paramétrer les systèmes des Organisme Client et Organisme Fournisseur.

3.2. Lot 2 : Vecteur et proxy Organisme Client

Le lot 2 regroupe tous les outils servant à la création et l'insertion de vecteurs d'identification signés dans les requêtes partant vers les Organismes Fournisseurs. Il est composé de trois modules :

- Module Proxy : il gère la présence du vecteur d'identification dans les requêtes et la bonne redirection de ces requêtes vers les Organismes Fournisseurs,
- Module Vecteur d'Identification : il est le module d'information permettant de corréler des identifications locales avec des vecteurs d'identification,

- Module Assertion SAML : il fournit les assertions SAML contenant les vecteurs d'identification signés.

3.3. Lot 3 : Vecteur et reverse proxy Organisme Fournisseur

Le lot 3 regroupe tous les outils servant à l'interception, l'analyse et la validation de vecteurs d'identification signés dans les requêtes arrivant des Organismes Clients. Il est composé de quatre modules :

- Module Reverse-Proxy : il gère la présence du vecteur d'identification dans les requêtes et la bonne redirection des requêtes vers les services visés locaux,
- Module Vecteur d'Identification : il valide le contenu des vecteurs d'identification,
- Module Assertion SAML : il valide la signature des vecteurs d'identification,
- Module transcription de vecteur d'identification : il permet de transcrire les PAGM transportés par le vecteur d'identification en identifications et autorisations locale.

3.4. Lot 4 : Traces

Les traces renforcent la confiance en permettant le contrôle à posteriori. Pour remplir cette fonction, le lot Traces est composé de deux modules :

- Module Traces : à proprement dit le module trace permet d'insérer des traces (le vecteur d'identification signé au format SAML) dans une base et de les relire,
- Outil d'analyse de trace : il permet l'analyse des traces et le contrôle à posteriori.

4. Lot 1 : Outils d'administration des accords

Ce lot regroupe les blocs fonctionnels (sous forme d'outils) servant à la mise en place des accords. En termes d'implémentation ils représentent essentiellement un format normalisé d'échange de données contenant les éléments de configuration des systèmes de chaque organisme. De ce point de vue, le format d'échange approprié est un format XML.

> ✍ *Dans le document du standard figure un exemple d'utilisation des techniques ebXML pour administrer ces accords dont nous suivons la logique de préparation. Voir aussi le paragraphe 2.2 Eléments techniques représentant les accords pour une description des flux de mise en place des services.*

4.1. Outil de publication Organisme Client

4.1.1. Rôle de l'outil

Il produit un fichier au format XML contenant les éléments de paramétrage de l'interopérabilité souhaités par l'Organisme Client. Il propose une interface permettant au sein de l'Organisme client de déclarer les éléments de configuration le concernant.

Cet outil doit permettre de signer ce document.

4.1.2. Interface d'entrée

4.1.2.1. Flux numéro 1 : la base des PAGM

La liste courante des associations PAGM – URI pour l'Organisme Fournisseur visé, si elle existe dans la base des PAGM, est insérée parmi la liste des paramètres à écrire dans le fichier résultat. L'outil permet de modifier cette liste.

4.1.2.2. Flux numéro 2 : gestion des certificats

Selon la configuration du système de l'Organisme Client, l'outil doit être capable d'insérer les certificats nécessaires à la mise en place du standard.

4.1.2.3. Flux numéro 3 : base d'identification

L'outil permet de rentrer la liste des administrateurs privilégiés et de la modifier.

4.1.2.4. Flux numéro 4 : éléments paramétrables

L'outil permet d'indiquer des valeurs pour les paramètres suivants :

- Services visés, chacun accompagné d'un descriptif,
- Numéro de version valide du format de vecteur d'identification,
- Durée de vie minimale et maximale des traces.

4.1.3. Interface de sortie

4.1.3.1. Flux numéro 1 : fichier normalisé de publication

L'outil génère un fichier lisible au format XML contenant l'ensemble des paramètres reçus en entrée.

4.2. Outil de publication Organisme Fournisseur

4.2.1. Rôle de l'outil

Il produit un fichier au format XML contenant les éléments de paramétrage demandés à l'Organisme Fournisseur. Il propose une interface permettant au sein de l'Organisme fournisseur de déclarer les éléments de configuration le concernant.

4.2.2. Interface d'entrée

4.2.2.1. Flux numéro 1 : gestion des certificats

Selon la configuration du système de l'Organisme Fournisseur, l'outil doit être capable d'insérer les certificats nécessaires à la mise en place du standard.

4.2.2.2. Flux numéro 2 : base d'identification

L'outil permet de rentrer la liste des administrateurs privilégiés et de la modifier.

4.2.2.3. Flux numéro 3 : la base des PAGM

La liste courante des associations PAGM – URI, si elle existe, est insérée parmi les paramètres. L'outil permet de modifier cette liste.

4.2.2.4. Flux numéro 4 : éléments paramétrables

L'outil permet d'indiquer des valeurs pour les paramètres suivants :

- Liste des services proposés, chacun accompagné d'un descriptif,
- Liste des associations entre les services et les PAGM sont, si nécessaire, les associations entre les PAGM et les URL visant certaines fonctions des services. Ici le fournisseur peut indiquer les éléments du service en libre accès,
- Numéro de version valide du format de vecteur d'identification,
- Durée de vie minimale et maximale des traces.

4.2.3. Interface de sortie

4.2.3.1. Flux numéro 1 : fichier normalisé de publication

L'outil génère un fichier lisible au format XML contenant l'ensemble des paramètres reçus en entrée.

4.3. Outil de création des accords

4.3.1. Rôle de l'outil

Cet outil a pour but de combiner les fichiers de publication après vérification de consistance créés par les deux outils de publication, afin de générer les éléments techniques de la convention d'interopérabilité entre les Organismes Client et Fournisseur.

4.3.2. Interface d'entrée

4.3.2.1. Flux numéro 1 : le fichier normalisé de publication Organisme Client

Les paramètres du fichier Organisme Client sont regroupés en :

- des paramètres communs qui ont une équivalence avec ceux du fichier Organisme Fournisseur,
- les paramètres spécifiques au client.

4.3.2.2. Flux numéro 2 : le fichier normalisé de publication Organisme Fournisseur

Les paramètres du fichier Organisme Fournisseur sont regroupés en :

- des paramètres communs qui ont une équivalence avec ceux du fichier Organisme Client,
- les paramètres spécifiques au fournisseur.

4.3.3. Interface de sortie

4.3.3.1. Flux numéro 1 : le fichier des éléments techniques des accords

Ce fichier, au format XML, combine le contenu des deux fichiers de l'Organisme Client et de l'Organisme Fournisseur. Il est le fichier des éléments techniques des accords et, à ce titre, est annexé à la convention passée entre les deux organismes établissant les modalités d'interopérabilité.

Ce fichier contient trois parties :

- une partie pour les paramètres spécifiques Organisme Client,
- une partie pour les paramètres spécifiques Organisme Fournisseur,
- une partie pour les paramètres communs.

Ce fichier est signé en utilisant les bi-clés/certificats fournis par les deux fichiers de publication et distribué aux deux partenaires.

4.4. Outil de mise en œuvre des accords

4.4.1. Rôle de l'outil

A partir du fichier des éléments techniques des accords, des outils déploient les paramètres configurant l'interopérabilité entre l'Organisme Client et l'Organisme Fournisseur sur le système d'information des deux organismes. Le déploiement sur chacun des deux systèmes est indépendant.

L'outil fonctionne soit en mode Organisme Client soit en mode Organisme Fournisseur.

4.4.2. Interface d'entrée

4.4.2.1. Flux numéro 1 : le fichier des éléments techniques des accords

Chaque élément de ce fichier est déployé sur le système d'information de l'organisme.

4.4.3. Interface de sortie

4.4.3.1. Flux numéro 1 : gestion des certificats

Le certificat fourni par l'organisme partenaire est inséré dans le système de gestion des certificats.

4.4.3.2. Flux numéro 2 : base de PAGM

L'outil insère dans la base la liste des associations URI – PAGM avec l'identifiant de l'organisme partenaire et le descriptif accompagnant chaque URI.

4.4.3.3. Flux numéro 3 : éléments de configuration du module de trace

La durée de vie minimale et maximale des traces pour l'organisme partenaire est intégrée.

> ✍ *Remarques de sécurité : la mise en place de ces accords ne peut pas être entièrement automatisée. Le traitement doit être coordonné et respecter les contraintes de sécurité de chaque organisme.*

5. Lot 2 : Vecteur et proxy Organisme Client

Le déploiement, côté Organisme Client du vecteur d'identification est composé de trois modules. Le module de construction du vecteur d'identification rassemble dans une structure de données tous les éléments concernant chaque requête ; le module de construction d'assertion SAML produit une assertion SAML signée contenant cette structure ; le module proxy, lors d'une requête client, fait appel aux deux autres modules pour la construction de cette assertion, ou vérifie l'assertion si elle est déjà fournie dans la requête et redirige la requête vers l'Organisme Fournisseur approprié.

5.1. Situation dans le standard

Ainsi que spécifié dans le standard, les fonctions locales d'attributions d'autorisation ne sont pas du ressort du standard. Les trois modules « Vecteur d'Identification », « Assertion SAML » et « Proxy » ont pour objectif de produire une autorisation inter-organismes en se basant sur l'autorisation locale :

- Module Vecteur Identification : module d'information, il renvoie des vecteurs d'identification,
- Module Assertion SAML : module de validation et de formatage de vecteurs d'identification,
- Module Proxy : insertion de vecteurs d'identification valides dans des requêtes clients et redirection vers les Organismes Fournisseurs.

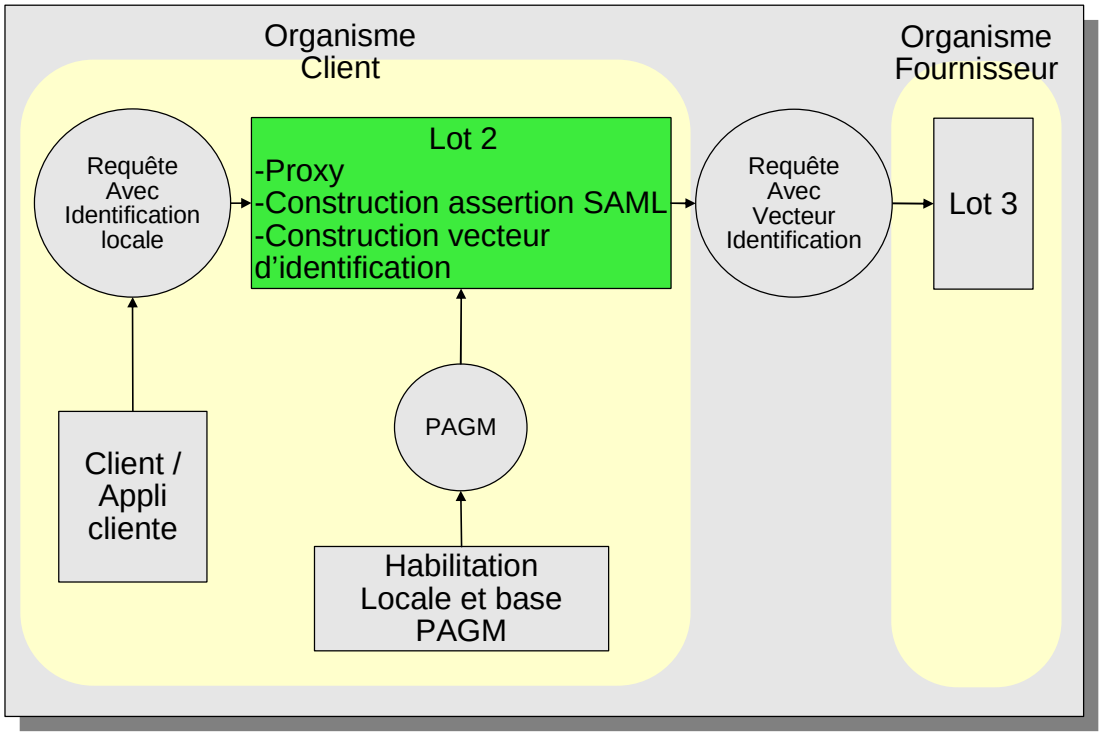
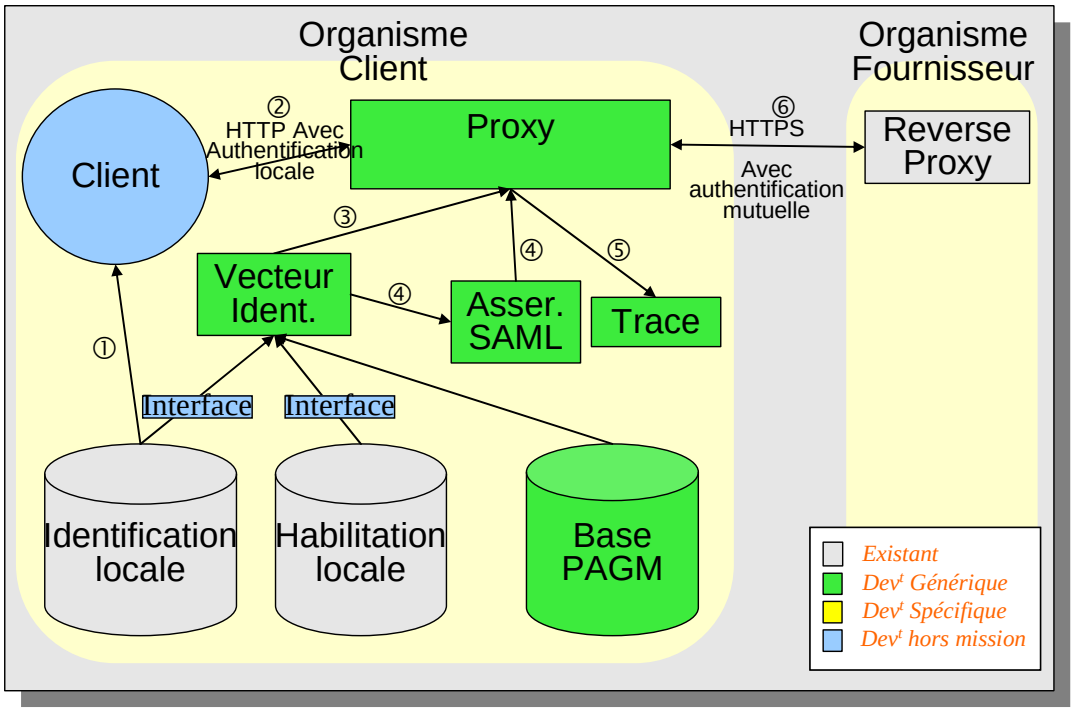


Figure 9 : Situation du lot 2 dans le standard

Les interactions entre les modules du lot 2 peuvent suivre différentes logiques. Le mode le plus simple est indiqué dans le schéma ci-dessous, il suit le modèle Portail-à-Portail :



744 **Figure 10 : Arrangement des modules du lot 2 selon le modèle Portail-à-Portail**

745 Les différents modules de ce lot se déclinent selon les deux modèles Portail-à-Portail et Web Service.
746 Toutefois, la différence essentielle entre les deux lots tient dans l'utilisation des modules de ce lot et non
747 dans les fonctionnalités de ces modules. Voir les paragraphes « 2.6.2.1 Mode de construction 1 » et
748 « 2.6.2.2 Mode de construction applicatif selon le modèle Web Service ».

749 En particulier même si le modèle Portail-à-portail n'a qu'un mode de fonctionnement, où l'application –le
750 navigateur– envoi sa requête au proxy sans assertion SAML, le modèle Web Service a deux modes de
751 fonctionnement :

752 ❑ Un mode similaire au modèle Portail-à-Portail où c'est le proxy qui gère l'ensemble des
753 autorisations,

754 ❑ Un mode spécifique où chaque application cliente a la possibilité de faire appel aux modules
755 Vecteur d'identification et Assertion SAML, par exemple à des fins de pré-configuration d'affichage
756 selon le type d'application ou d'utilisateur.

757 ## 5.2. Module proxy client modèle portail à portail

758 ### 5.2.1. Rôle du module

759 Ce module modifie les requêtes HTTP reçues en insérant le vecteur d'identification en fonction des
760 éléments de cette requête, de ceux de l'accord d'interopérabilité ainsi que du système local de gestion des
761 identifications. Si la requête comprend déjà un vecteur d'identification, il le valide. Puis il permet la
762 redirection de la requête vers l'Organisme Fournisseur du service visé.

763 Le module proxy a aussi un rôle de transcription d'adresses. En fonction des éléments de configuration il
764 remplace les adresses locales demandées par le client en adresses externes. De même, à réception des
765 données de l'Organisme Fournisseur il transcrit les adresses externes en adresses locales.

Le module proxy utilise les deux modules « vecteur d'identification » et « assertion SAML ».

Le module proxy trace les assertions SAML validées lorsqu'une requête est sur le point d'être redirigée vers l'Organisme Fournisseur adéquat.

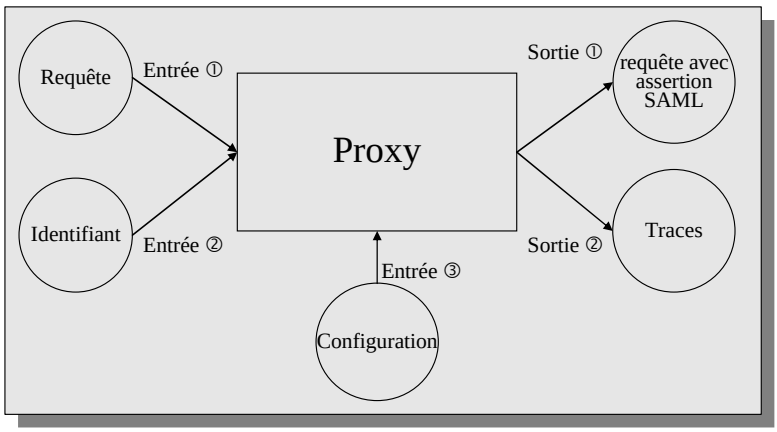


Figure 11 : Flux d'entrées/sorties du module proxy

5.2.2. Interface d'entrée

5.2.2.1. Flux numéro 1 : la requête

L'application intégrant le module proxy doit fournir au module proxy la requête envoyée par le navigateur au format HTTP. Le service visé doit être compris dans l'entête **Host**, selon les règles de nommage indiquées au paragraphe *2.5.2 Dénomination de service*. Par exemple :

```
GET / HTTP/1.x
Host: nom-local-rniam.domaine-local-cnav.fr
```

Il est de même possible de fournir avec la requête l'élément d'entête **X-IOPS-Vecteur-Identification** pour forcer l'utilisation d'un vecteur d'identification particulier.

```
X-IOPS-Vecteur-Identification: assertion SAML encodée en base-64
```

Si cet élément est fourni, le module proxy le valide ou, en cas de non validation, retourne une erreur (voir sur l'interface de sortie *5.2.3.1 Flux numéro 1 : la requête*). Si cet élément n'est pas fourni, le module proxy en génère un à l'aide du module Assertion SAML.

La validation du vecteur d'identification consiste à :

- Décoder de base-64 vers assertion SAML le champ X-IOPS-Vecteur-Identification,
- Vérifier la signature à l'aide du module Assertion SAML,
- Vérifier le service visé à l'aide de la valeur du champ Host avec le chemin de la requête,
- Vérifier l'identification à l'aide des informations identifiant définies à *Flux numéro 2 : l'identifiant*.

5.2.2.2. Flux numéro 2 : l'identifiant

L'application intégrant le module proxy doit fournir au module proxy l'élément d'identification de l'utilisateur requérant. Cet élément doit être fourni à l'aide des deux variables d'environnement **AUTH_TYPE** et **REMOTE_USER**.

Le module proxy doit s'accommoder de cet élément pour soit valider le vecteur d'identification fourni soit fournir un vecteur d'identification valide.

5.2.2.3. Flux numéro 3 : éléments de configuration

Les éléments à prendre en compte sont :

- La valeur du paramètre X-IOPS-Vecteur-Identification (par défaut : « X-IOPS-Vecteur-Identification »),
- La liste de transcription de nom de services locaux en nom de services externes.

5.2.3. Interface de sortie

5.2.3.1. Flux numéro 1 : la requête

Le module proxy doit, à l'aide des deux éléments en entrée, fournir la requête en sortie vers l'Organisme Fournisseur du service visé. Le format de la requête en sortie reste celui de la requête en entrée. Seuls les éléments **Host** et **X-IOPS-Vecteur-identification** de l'entête de la requête sont affectés. Le champ **Host** reçoit la transcription du nom local de service en nom externe de service ; le champ **X-IOPS-Vecteur-Identification** reçoit le vecteur d'identification en tant qu'assertion SAML signée, au format Base-64.

Si le module proxy ne peut fournir de vecteur d'identification valide, il renvoie au requérant le message d'erreur type *HTTP/1.0 401 Unauthorized*.

Si la requête d'origine contient un élément X-IOPS-Vecteur-identification valide, il est conservé tel quel. Autrement, l'insertion du vecteur d'identification dans la requête en sortie consiste à :

- Générer une assertion SAML signée en utilisant le module Assertion SAML, la valeur du champ Host avec le chemin de la requête et l'identifiant fourni par AUTH_TYPE et REMOTE_USER,
- Encoder l'assertion SAML en base-64 et l'insérer dans l'entête de la requête en tant que valeur du champ X-IOPS-Vecteur-identification.

5.2.3.2. Flux numéro 2 : le module de Traces

Le module Proxy trace les assertions SAML validées lorsqu'une requête est sur le point d'être redirigée vers l'Organisme Fournisseur adéquat.

5.2.4. Performances

Les éléments de performances concernent deux points :

- La création d'assertions SAML,

- Le traçage des assertions SAML.

La création et le traçage d'une assertion SAML par requête pose le problème d'accès à des pages contenant, par exemple, beaucoup d'images. Un accès à une page démultiplie les requêtes. Pour réduire les créations et traçages, le module proxy peut embarquer un mécanisme de cache. Les stratégies d'utilisation du cache peuvent être, par exemple, de réutiliser (transmission pour la requête elle même et non-enregistrement pour le traçage) l'assertion SAML pour le même identifiant tant que la durée de validité du vecteur n'est pas écoulée. En outre, il est prévu et décrit précédemment que le fournisseur de services peut spécifier des parties du service en accès libre pour lesquelles le module ne fait aucune vérification ni création SAML (exemple des images).

5.2.5. Administration du module

Ceci dépend du contexte d'implémentation du serveur Web utilisé (Apache, JBOSS, Java, etc.) et utiliser les éléments d'administration standards prévus.

5.3. Module proxy client modèle web service

5.3.1. Rôle du module

Ce module modifie les requêtes reçues en insérant le vecteur d'identification en fonction des éléments de cette requête, de ceux de l'accord d'interopérabilité ainsi que du système local de gestion des identifications. Si la requête comprend déjà un vecteur d'identification, il le valide. Puis il permet la redirection de la requête vers l'Organisme Fournisseur du service visé.

Le module proxy a aussi un rôle de transcription d'adresses. En fonction des éléments de configuration il remplace les adresses locales demandées par le client en adresses externes.

Le module proxy utilise les deux modules « vecteur d'identification » et « assertion SAML ».

Le module proxy trace les assertions SAML validées lorsqu'une requête est sur le point d'être redirigée vers l'Organisme Fournisseur adéquat.

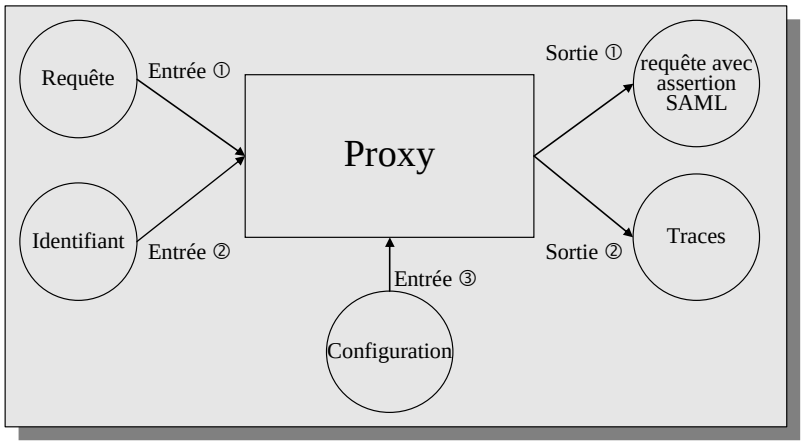


Figure 12 : Flux d'entrées/sorties du module proxy

5.3.2. Interface d'entrée

5.3.2.1. Flux numéro 1 : la requête

L'application intégrant le module proxy doit fournir au module proxy la requête envoyée par l'application web service SOAP, encapsulé dans une requête HTTP de type POST. Le service visé doit être compris dans l'entête **Host**, selon les règles de nommage indiquées au paragraphe *2.5.2 Dénomination de service*. Par exemple :

```
POST / HTTP/1.1
Host: nom-local-rniam.domaine-local-cnav.fr
SOAPAction: "http://nom-local-rniam.domaine-local-cnav.fr"
```

Il est de même possible de fournir avec la requête le vecteur d'identification au format SAML pour forcer l'utilisation d'un vecteur d'identification particulier.

Si cet élément est fourni, le module proxy le valide ou, en cas de non validation, retourne une erreur (voir sur l'interface de sortie *5.2.3.1 Flux numéro 1 : la requête*). Si cet élément n'est pas fourni, le module proxy en génère un à l'aide du module Assertion SAML.

La validation du vecteur d'identification consiste à :

- Extraire de l'assertion SAML les éléments formant le vecteur d'identification,
- Vérifier la signature à l'aide du module Assertion SAML,
- Vérifier le service visé à l'aide de la valeur du champ Host avec le chemin de la requête,
- Vérifier l'identification à l'aide des informations identifiant définies à *Flux numéro 2 : l'identifiant*.

5.3.2.2. Flux numéro 2 : l'identifiant

L'application intégrant le module proxy doit fournir au module proxy l'élément d'identification de l'application requérante (ou de l'utilisateur de l'application). Cet élément doit être fourni à l'aide des deux variables d'environnement **AUTH_TYPE** et **REMOTE_USER**.

Le module proxy doit s'accommoder de cet élément pour soit valider le vecteur d'identification fourni soit fournir un vecteur d'identification valide.

5.3.2.3. Flux numéro 3 : éléments de configuration

Les éléments à prendre en compte sont les données de l'environnement local permettant :

- d'identifier/authentifier l'utilisateur ou l'application à l'origine de la requête, tel qu'un annuaire LDAP,
- La liste de transcription de nom de services locaux en nom de services externes.

5.3.3. Interface de sortie

5.3.3.1. Flux numéro 1 : la requête

Le module proxy doit, à l'aide des deux éléments en entrée, fournir la requête en sortie vers l'Organisme Fournisseur du service visé. Le format de la requête en sortie reste celui de la requête en entrée. Dans l'entête HTTP seul l'élément **Host** est affecté : il reçoit la transcription du nom local de service en nom externe de service. Dans l'entête de la requête SOAP, seul l'élément contenant le vecteur d'identification est affecté : il contient le vecteur d'identification en tant qu'assertion SAML signée.

Si le module proxy ne peut fournir de vecteur d'identification valide, il renvoie au requérant le message d'erreur type *env:Sender*.

Si la requête d'origine contient un vecteur d'identification valide, il est conservé tel quel. Autrement, l'insertion du vecteur d'identification dans la requête en sortie consiste à :

- Générer une assertion SAML signée en utilisant le module Assertion SAML, la valeur du champ Host avec le chemin de la requête et l'identifiant fourni par AUTH_TYPE et REMOTE_USER et
- Encoder l'assertion SAML et l'insérer dans l'entête de la requête.

5.3.3.2. Flux numéro 2 : le module de Traces

Le module Proxy trace les assertions SAML validées lorsqu'une requête est sur le point d'être redirigée vers l'Organisme Fournisseur adéquat.

5.3.4. Performances

Les éléments de performance sont similaires à ceux exposés au paragraphe *5.2.4 Performances*.

5.3.5. Administration du module

L'administration du module dépend du type de proxy mis en place et du type de protocole sous-jacent permettant de véhiculer les messages SOAP.

5.4. Module de construction du vecteur d'identification

5.4.1. Rôle du module

Le rôle du module de construction du vecteur d'identification est de fournir un vecteur d'identification valide pour un identifiant donné (un utilisateur ou une application) ainsi que pour un service visé d'un Organisme Fournisseur. Le vecteur d'identification est une structure de donnée décrite au paragraphe *2.6.1 Eléments du vecteur d'identification*, à l'exception de la signature.

A l'aide de ce vecteur d'identification une application (un navigateur web ou une application web service) pourra accéder au service visé indiqué dans ce vecteur, à condition que ce vecteur soit signé (à l'aide du module de construction SAML) par l'Organisme Client.

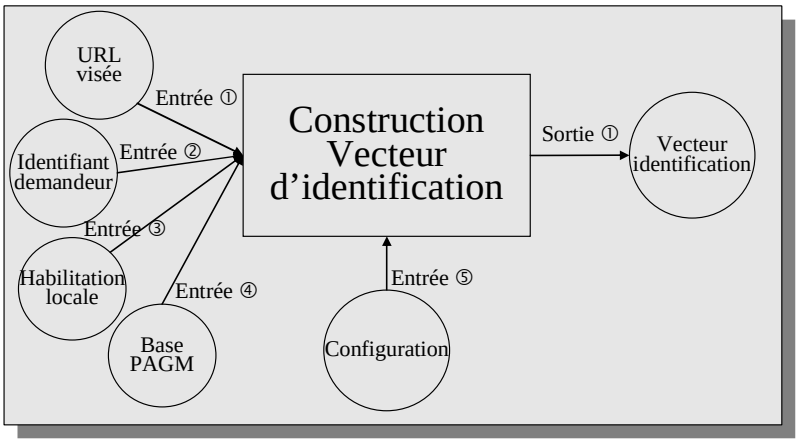
Les fonctions de ce module sont :

- Récupérer la liste de PAGM valides pour un service visé donné,

908 - Récupérer la liste de PAGM valides pour un identifiant (utilisateur ou application) donné,

909 - Récupérer la liste de PAGM valides pour un identifiant et un service donnés,

910 - Fournir un vecteur d'identification non signé.



911 **Figure 13 : Flux d'entrées/sorties du module de construction du Vecteur d'Identification**

912 ### 5.4.2. Interface d'entrée

913 #### 5.4.2.1. Flux numéro 1 : l'URL visée

914 L'URL visée doit être préfixée par le nom du service visé (voir les règles de nommage au paragraphe *2.5.2*
915 *Dénomination de service*). Il est entendu que « URL visée » correspond à l'URL vue localement par le
916 client (navigateur web ou application web service).

917 #### 5.4.2.2. Flux numéro 2 : l'identifiant du demandeur

918 L'identifiant est composé de deux éléments :

919 - L'identifiant de l'utilisateur ou de l'application proprement-dit,

920 - Le type d'authentification permettant de sélectionner l'interface interne d'accès au système local
921 d'habilitation.

922 #### 5.4.2.3. Flux numéro 3 : la base d'habilitation locale

923 La définition de ce flux vient des éléments de configuration.

924 En se basant sur l'identifiant le module de construction accède au système local d'habilitation pour
925 connaître la liste des PAGM affectés à cet identifiant. L'accession au système local est due au fait que la
926 gestion des autorisations (et donc l'association entre PAGM et utilisateur ou application) est gérée hors du
927 standard.

928 L'accession au système local doit se faire à travers l'une des interfaces applicatives suivantes :

929 - LDAP,

930 - AAA,

- Solutions spécifiques…

D'autres interfaces applicatives, plus spécifiques peuvent être implémentées dans ce module au cas par cas. Ceci est défini dans les documents d'applicabilité du standard.

> ✍ *Dans le cadre de la rédaction des spécifications détaillées objets de la présente mission, l'objectif est d'identifier toutes les interfaces nécessaires aux organismes clients présents autour de la table, afin de les intégrer au lot. Cette identification est réalisée dans le cadre des réunions avec chaque organisme client [action en cours].*

5.4.2.4. Flux numéro 4 : la base des PAGM

La définition de ce flux vient des éléments de configuration.

En se basant sur le service visé le module de construction recherche dans la base des PAGM pour connaître la liste des PAGM requis pour accéder au service visé. Le résultat de cette recherche est une liste de liste de PAGM.

L'accès à la base de PAGM doit se faire par configuration locale dans le cas où les PAGM ne sont pas directement attribués par l'habilitation locale (flux numéro 3).

5.4.2.5. Flux numéro 5 : éléments de configuration

Les éléments de configuration de ce module sont basés sur l'accord d'interopérabilité et sur la structure du système local d'habilitation. Les éléments suivants doivent être définis :

- Le type d'accès par défaut au système local d'habilitation (LDAP,…),
- L'identification du système local d'habilitation,
- La configuration d'association rôle/PAGM en cas de besoin,
- Des paramètres pour accéder au serveur LDAP ou AAA.

5.4.3. Interface de sortie

5.4.3.1. Flux numéro 1 : le vecteur d'identification

En réponse à la demande d'un vecteur d'identification en fonction de tous les éléments en entrée, le module renvoie soit une erreur d'identification, soit une erreur d'autorisation soit un vecteur d'identification contenant une liste de PAGM.

Chaque vecteur d'identification est retourné sous la forme d'une structure telle que définie au paragraphe *2.6.1 Eléments du vecteur d'identification*.

> ✍ *L'association entre les PAGM et un identifiant donné peut s'effectuer à travers une indirection, par exemple les rôles. Si plusieurs rôles sont attribués à un identifiant alors les PAGM attribués à chaque rôle sont, naturellement, attribués à l'identifiant. Pour les vecteurs d'identification, cela peut prendre deux formes : soit une liste de vecteurs d'identification (un vecteur par rôle) soit un seul vecteur d'identification dont la liste de PAGM est la concaténation de tous les PAGM attribués à l'identifiant.*

5.4.4. Performances

Les performances de ce module dépendent de l'accès à la base PAGM et au système local d'habilitation.

5.5. Module de construction de l'assertion SAML

5.5.1. Rôle du module

Le rôle du module de construction de l'assertion SAML est de signer un vecteur d'identification et de le renvoyer sous forme d'assertion SAML ou de créer un nouveau vecteur d'identification, de le signer et de le renvoyer sous forme d'assertion SAML.

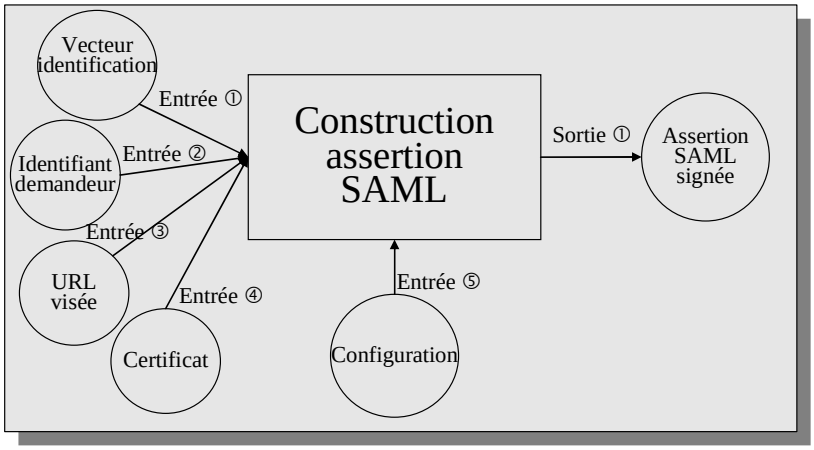


Figure 14 : Flux d'entrées/sorties

5.5.2. Interface d'entrée

5.5.2.1. Flux numéro 1 : le vecteur d'identification

Le vecteur d'identification est fourni au module dans le but d'être validé et signé.

5.5.2.2. Flux numéro 2 : l'identifiant du demandeur

Cet élément est nécessaire car il se peut que le vecteur d'identification ne contienne qu'un identifiant dépersonnalisé. Il est composé de deux éléments :

- L'identifiant de l'utilisateur ou de l'application proprement-dit,
- Le type d'authentification permettant de sélectionner l'interface interne d'accès au système local d'habilitation.

5.5.2.3. Flux numéro 3 : l'URL visée

La validation du vecteur d'identification est réalisée par l'appel au module Vecteur d'Identification avec les éléments suivant :

- Identifiant du demandeur,
- Service visé (contenu dans le vecteur d'identification),
- L'URL du service visé.

Si le vecteur d'identification renvoyé par le module de construction du vecteur d'identification ne correspond pas à celui passé en paramètre (flux numéro 1) il y a erreur d'autorisation. Si le module Vecteur d'Identification renvoie une erreur d'identification, le module Assertion SAML renvoie aussi l'erreur d'identification.

Toutefois, si aucun PAGM n'était fourni en entrée au module Assertion SAML dans le vecteur d'identification alors le module signe le vecteur d'identification renvoyé par le module Vecteur d'Identification.

5.5.2.4. Flux numéro 4 : le certificat

La clé privée associée à un certificat servant à la signature provient des accords d'interopérabilité. Dans le cadre de ce module le certificat est défini dans les éléments de configuration. Il sert à la signature des vecteurs d'identifications.

5.5.2.5. Flux numéro 5 : éléments de configuration

Les éléments de configuration sont :

- Le certificat servant à la signature des assertions SAML

5.5.3. Interface de sortie

5.5.3.1. Flux numéro 1 : l'assertion SAML signée

En réponse à la demande d'un vecteur d'identification en fonction de tous les éléments en entrée, le module renvoie soit une erreur d'identification, soit une erreur d'autorisation soit un vecteur d'identification signé au format SAML. Il utilise pour cela le module de construction de vecteur d'identification ainsi que le certificat défini dans les éléments de configuration.

5.5.4. Performance

Les performances de ce module sont liées d'une part aux performances du module Vecteur d'Identification (voir le paragraphe *5.4 Module de construction du vecteur d'identification)* et d'autre part aux performances de génération de la signature. L'amélioration de performance peut reposer sur l'utilisation de moyen matériel de génération de signature.

6. Lot 3 : Vecteur et reverse proxy Organisme Fournisseur

L'application du standard côté Organisme Fournisseur englobe la réception des requêtes en provenance des Organismes Clients, autorisation à l'exécution locale et leur redirection vers les services adéquats. Il s'agit du pendant aux fonctions Vecteur et proxy de l'Organisme Client.

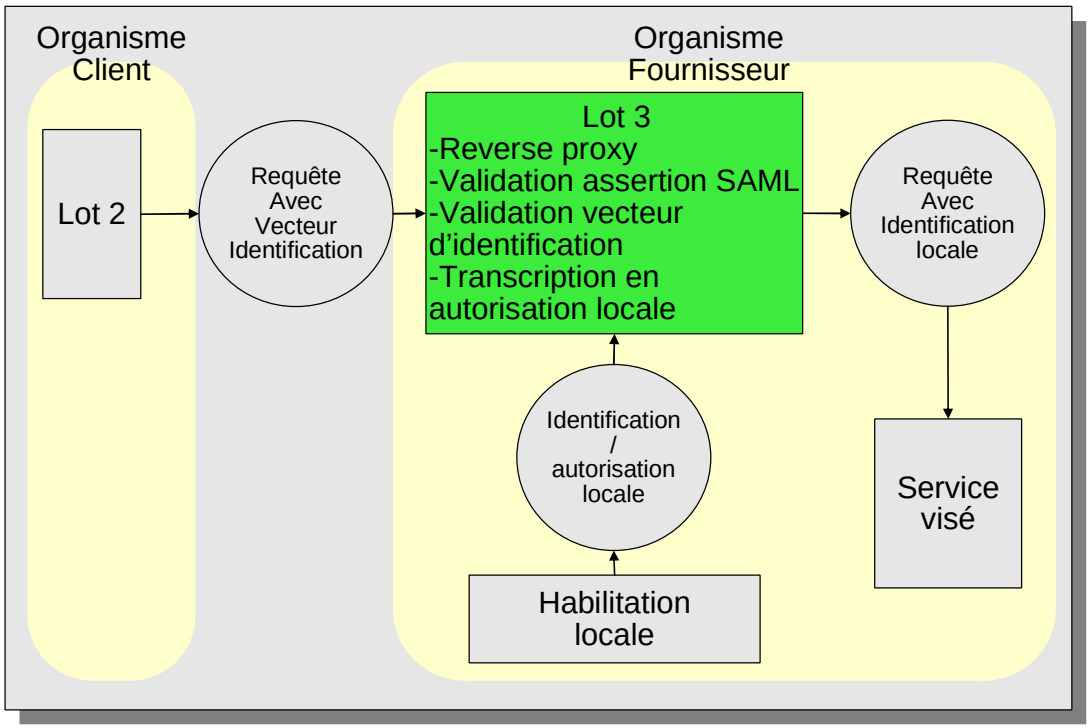
6.1. Situation dans le standard

Les modules du lot Vecteur et Reverse Proxy Fournisseur ont pour objectif de fournir une autorisation locale en remplacement de l'autorisation inter-organisme transportée dans les requêtes en provenance des Organismes Clients. Il s'agit donc d'intercepter les requêtes dès leur entrée dans le système de l'Organisme Fournisseur, en amont des mécanismes locaux de vérification des autorisations.

Le lot regroupe quatre modules :

- Module Reverse-proxy : il intercepte, dans les requêtes, les vecteurs d'identifications signés sous forme d'assertion SAML,
- Module Assertion SAML : il valide la signature de l'assertion,
- Module Vecteur d'identification : il valide les informations du vecteur d'identification,
- Module Transcription : il remplace le vecteur d'identification validé par les informations locales d'autorisation.

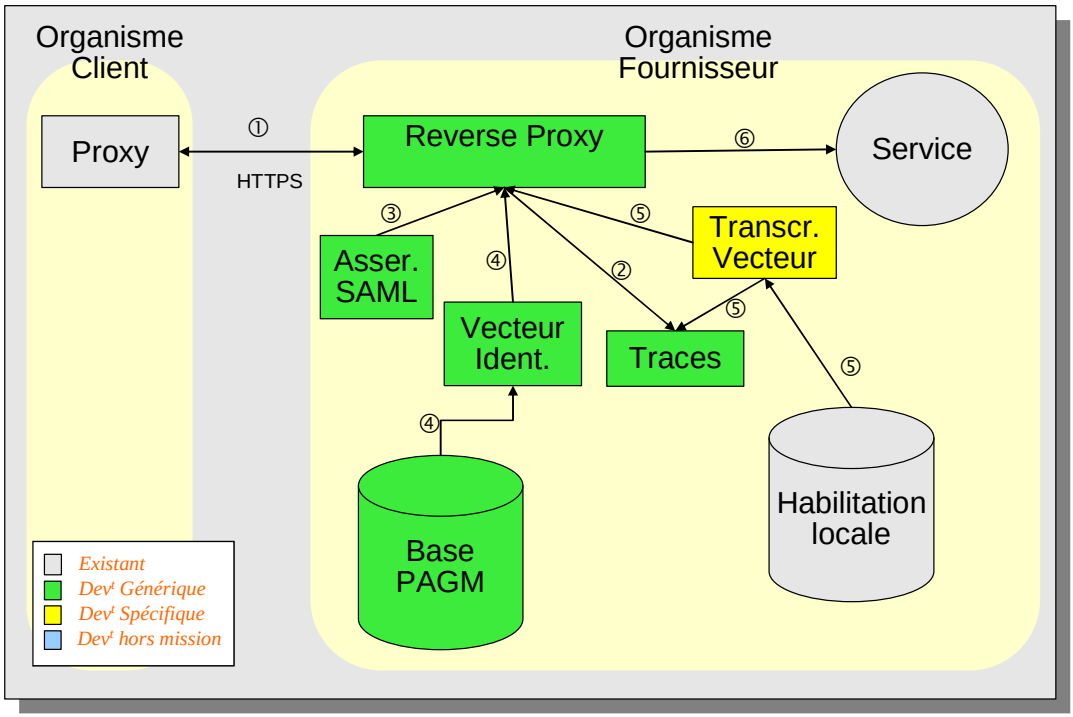
> ✍ *Concernant le terme* Reverse-proxy *: de manière similaire au module proxy, le module reverse proxy agit comme une passerelle faisant le lien entre les adresses externes et les adresses locales au fournisseur. Il agit en outre en reverse proxy dans le sens où il permet de vérifier des habilitations quant à l'accès à des applications, et de s'abstraire de l'architecture même de ces applications (par exemple la distribution de l'application sur plusieurs machines).*



1035 **Figure 15 : Situation du lot 3 dans le standard**

1036 ✍ *A l'image du lot Vecteur et proxy côté Organisme Client, le module reverse-proxy côté Organisme*
1037 *Fournisseur est décliné en deux versions, l'une pour le modèle Portail-à-Portail, l'autre pour le*
1038 *modèle Web Service.*

1039 L'arrangement des modules du lot 3 suit le schéma suivant :



1040 **Figure 16 : Arrangement des modules du lot 3**

1041 ## 6.2. Module reverse-proxy modèle portail à portail

1042 ### 6.2.1. Rôle du module

1043 Ce module reçoit les requêtes HTTP des Organismes Clients contenant un vecteur d'identification. Il
1044 valide le vecteur et permet la redirection de la requête vers le service visé.

1045 Le module reverse-proxy transcrit les adresses externes en adresses locales à réception de la requête. De
1046 même lorsqu'il transmet le résultat de la requête il transcrit les adresses locales en adresses externes.

1047 Le module reverse-proxy utilise les trois modules « Validation Assertion SAML », « Validation Vecteur
1048 d'identification » et « Transcription Identification Locale ».

1049 Le module reverse-proxy trace les assertions SAML dès réception d'une requête.

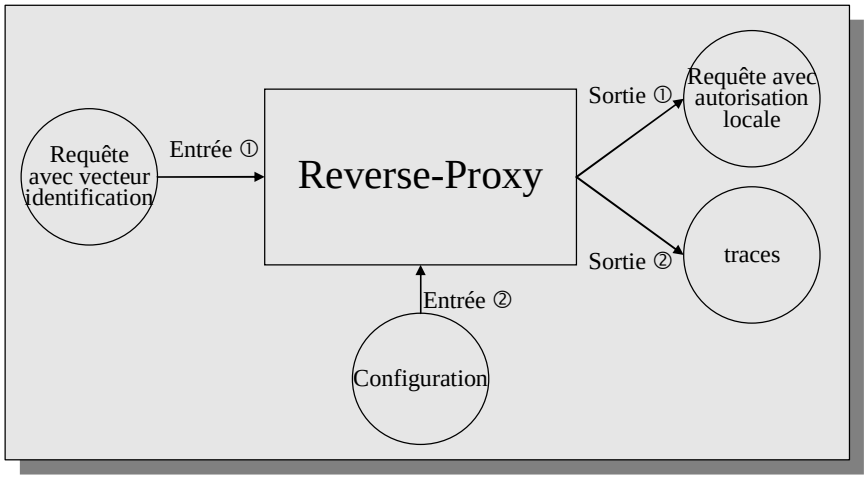


Figure 17 : Flux d'entrées/sorties du module Reverse-Proxy

6.2.2. Interface d'entrée

6.2.2.1. Flux numéro 1 : la requête

En provenance du proxy de l'Organisme Client, la requête contient en entête dans le champ **X-IOPS-Vecteur-identification** le vecteur d'identification en tant qu'assertion SAML signée. En outre, le champ **Host** contient le service visé, selon les règles de nommage indiquées au paragraphe *2.5.2 Dénomination de service*. Par exemple :

```
GET / HTTP/1.0
Host: rniam.cnav.fr
X-IOPS-Vecteur-Identification: assertion SAML
```

Si le champ X-IOPS-Vecteur-Identification n'est pas présent et que l'URL visée requiert un vecteur d'identification, le reverse-proxy renvoie le message d'erreur type *HTTP/1.0 401 Unauthorized*.

Le vecteur d'identification est transmis sous forme d'assertion SAML au module « Validation Assertion SAML » pour validation de la signature puis au module « Validation Vecteur d'Identification » pour validation du vecteur d'identification contenu dans l'assertion.

Le vecteur d'identification validé et la requête sont transmis au module « Transcription Identification locale » pour associer la requête à une identification/autorisation locale correspondant au service visé.

6.2.2.2. Flux numéro 2 : éléments de configuration

Les éléments à prendre en compte sont :

- La valeur du paramètre X-IOPS-Vecteur-Identification (par défaut : « X-IOPS-Vecteur-Identification,
- Les éléments permettant de transcrire les adresses locales en adresses externes et vice-versa.

6.2.3. Interface de sortie

6.2.3.1. Flux numéro 1 : la requête avec autorisation locale

Le module reverse-proxy permet la redirection de la requête vers le service visé. La requête contient les éléments d'identification/autorisation locaux nécessaires à l'accès au service visé.

6.2.3.2. Flux numéro 2 : le module de Traces

Le module Proxy trace les assertions SAML dès qu'elles sont reçues de l'Organisme Client.

6.2.4. Performance

De même que pour le proxy côté Organisme Client les éléments de performance concernent les points :

- La validation d'assertions SAML,
- Le traçage des assertions SAML.

Ici aussi un mécanisme de cache peut fortement accroître les performances du module. En particulier, pour la validation, une stratégie d'utilisation de cache peut être :

- Première requête : assertion SAML enregistrée,
- Requêtes suivantes (même Organisme Client), tant que la validité du premier vecteur n'est pas terminée, il est possible de comparer le nouveau vecteur avec le premier vecteur, au lieu d'effectuer une validation de la signature puis une validation des éléments du vecteur.

De manière similaire, pour le traçage une stratégie d'utilisation de cache peut être de n'enregistrer que la première assertion SAML d'un même Organisme Client tant que sa durée de validité n'est pas écoulée et les éléments des assertions suivantes sont identiques à celle-ci.

Il est à décider par chaque organisme si ce type de stratégie est applicable aux vecteurs d'identification dépersonnalisés.

6.3. Module reverse-proxy modèle web service

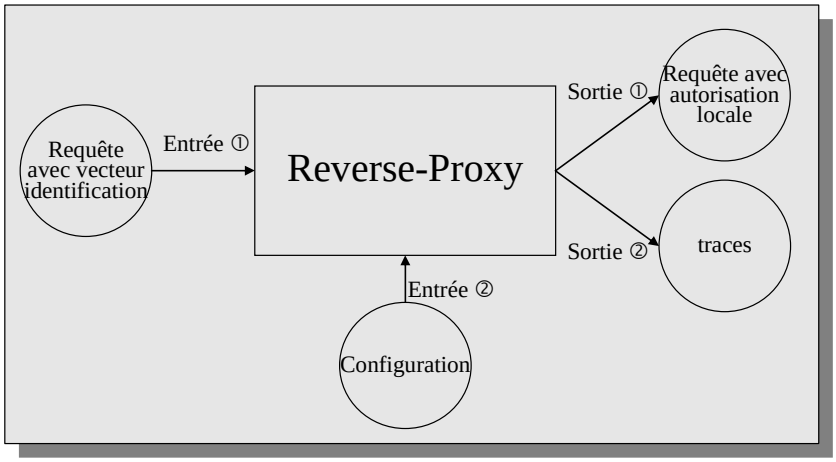
6.3.1. Rôle du module

Ce module reçoit les requêtes SOAP des Organismes Clients contenant un vecteur d'identification. Il valide le vecteur et permet la redirection de la requête vers le service visé. Il agit donc à la manière d'un relais Web Service.

Le module reverse-proxy transcrit les adresses externes en adresses locales à réception de la requête.

Le module reverse-proxy utilise les trois modules « Validation Assertion SAML », « Validation Vecteur d'identification » et « Transcription Identification Locale ».

Le module reverse-proxy trace les assertions SAML dès réception d'une requête.



1102 **Figure 18 : Flux d'entrées/sorties du module Reverse-Proxy**

1103 ### 6.3.2. Interface d'entrée

1104 #### 6.3.2.1. Flux numéro 1 : la requête

1105 En provenance du proxy de l'Organisme Client, la requête SOAP contient le vecteur d'identification en tant
1106 qu'assertion SAML signée. En outre, le champ **Host** de l'entête HTTP contient le service visé, selon les
1107 règles de nommage indiquées au paragraphe *2.5.2 Dénomination de service*. Par exemple :

```
1108 POST / HTTP/1.1
1109 Host: rniam.cnav.fr
1110 SOAPAction: "http://rniam.cnav.fr"
```

1111 ✍ *Noter que, vis-à-vis de l'exemple donné au paragraphe* 5.2.3.1 Flux numéro 1 : la requête*, la valeur*
1112 *de SOAPAction est ici bien celui du service publié.*

1113 Si le vecteur d'identification au format SAML n'est pas présent et que l'URL visée requiert un vecteur
1114 d'identification, le reverse-proxy renvoie le message d'erreur type *env:Sender*.

1115 Le vecteur d'identification est transmis sous forme d'assertion SAML au module « Validation Assertion
1116 SAML » pour validation de la signature puis au module « Validation Vecteur d'Identification » pour
1117 validation du vecteur d'identification contenu dans l'assertion.

1118 Le vecteur d'identification validé et la requête sont transmis au module « Transcription Identification
1119 locale » pour associer la requête à une identification/autorisation locale correspondant au service visé.

1120 #### 6.3.2.2. Flux numéro 2 : éléments de configuration

1121 Les éléments à prendre en compte sont :

1122 ❑ Les éléments permettant de transcrire les adresses locales en adresses externes et vice-versa.

6.3.3. Interface de sortie

6.3.3.1. Flux numéro 1 : la requête avec autorisation locale

Le module reverse-proxy permet la redirection de la requête vers le service visé. La requête contient les éléments d'identification/autorisation locaux nécessaires à l'accès au service visé.

6.3.3.2. Flux numéro 2 : le module de Traces

Le module Proxy trace les assertions SAML dès qu'elles sont reçues de l'Organisme Client.

6.3.4. Performance

Les remarques quant à la performance de ce module sont les mêmes que pour la version Portail-à-Portail, décrites au paragraphe *6.2.4* Performance.

6.4. Module validation de l'assertion SAML

6.4.1. Rôle du module

La validation de l'assertion SAML consiste à vérifier la signature apposée sur l'assertion et, si la vérification est correcte, à renvoyer le vecteur d'identification contenu dans cette assertion SAML.

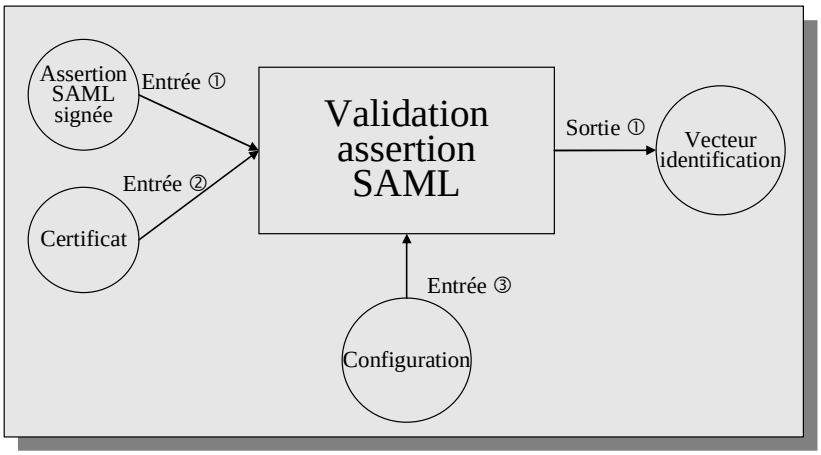


Figure 19 : Flux d'entrées/sorties du module Validation assertion SAML

6.4.2. Interface d'entrée

6.4.2.1. Flux numéro 1 : l'assertion SAML

Il s'agit du vecteur d'identification sous format d'assertion SAML signée.

6.4.2.2. Flux numéro 2 : le certificat

Le certificat servant à la signature provient des accords d'interopérabilité. Dans le cadre de ce module il est défini dans les éléments de configuration. Il sert à la vérification de la signature des vecteurs d'identifications. Si la vérification de la signature échoue le module renvoie une erreur d'authentification.

6.4.2.3. Flux numéro 3 : éléments de configuration

Les éléments de configuration sont :

- Le certificat servant à la vérification de signature des assertions SAML.

6.4.3. Interface de sortie

6.4.3.1. Flux numéro 1 : le vecteur d'identification

Si la vérification de la signature de l'assertion a échoué le module renvoie une erreur d'authentification. Autrement, le vecteur d'identification contenu dans l'assertion SAML est renvoyé sous forme d'une structure de données.

6.4.4. Performance

Les performances de ce module sont uniquement liées à la vérification de la signature. L'amélioration de performance peut reposer sur l'utilisation de moyen matériel de vérification de signature.

6.5. Module validation du vecteur d'identification

6.5.1. Rôle du module

Le but de ce module est de valider un vecteur d'identification. La validation du vecteur permet de s'assurer que l'organisme d'origine de la requête a bien le droit d'affecter les PAGM fournis pour l'URL visée. Si tel est le cas alors la requête respecte les accords d'interopérabilité.

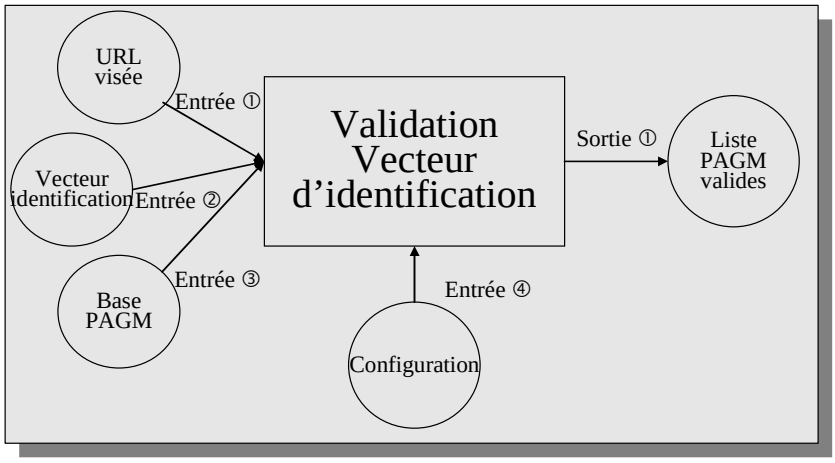


Figure 20 : Flux d'entrées/sorties du module de Validation Vecteur d'Identification

6.5.2. Interface d'entrée

6.5.2.1. Flux numéro 1 : l'URL visée

Les PAGM requis ne sont pas nécessairement les mêmes pour un service donné selon les fonctions de ce service. L'URL précise permet de valider chaque cas.

6.5.2.2. Flux numéro 2 : le vecteur d'identification

La validation du vecteur suit les règles ci-dessous :

- Le numéro de version : il doit être conforme aux spécifications de l'accord,
- L'identifiant de l'Organisme Fournisseur : il doit être égal à l'identifiant de l'organisme,
- La durée de validité du vecteur : il doit être en cours de validité,
- L'identifiant de l'Organisme Client, la liste des PAGM et l'URL visée doivent être associés dans la base des PAGM. A défaut, le test est répété en remplaçant l'URL visée avec le service visé,
- Selon les accords : les attributs (indication géographique, localisation,…) sont aussi vérifiés en combinaison avec le triplet valide {identifiant client, PAGM, URL} ou {identifiant client, PAGM, service visé}.

6.5.2.3. Flux numéro 3 : la base des PAGM

La définition de ce flux vient des éléments de la configuration du serveur, comprenant une liste d'associations PAGM/Services.

6.5.2.4. Flux numéro 4 : éléments de configuration

Les éléments de configuration de ce module sont la liste des associations de PAGM entre les applications locales et les PAGM en fonction des identifiants d'organismes (ces données viennent des accords d'interopérabilité).

6.5.3. Interface de sortie

6.5.3.1. Flux numéro 1 : liste des PAGM valides

Ce module retourne, en cas de succès de la validation, la liste exacte (ce qui est nécessaire et suffisant) des PAGM nécessaires pour l'URL visée. En cas d'échec une erreur est renvoyée.

Le résultat de la validation se décline de la façon suivante :

- Succès : le vecteur est correct,
- Echec d'identification : le numéro de version, l'identifiant de l'Organisme Fournisseur ou l'identifiant de l'Organisme Client n'est pas correct,
- Echec d'authentification : le vecteur n'est plus en cours de validité,
- Echec d'autorisation : tous les autres cas d'échec de validation.

6.5.4. Performance

Les performances de ce module dépendent de l'accès à la base PAGM.

6.6. Module transcription du vecteur d'identification en identification locale

6.6.1. Rôle du module

Ce module est nécessairement spécifique à chaque service de chaque organisme. Son rôle est d'associer une requête entrante (dont le vecteur d'identification est dûment validé) avec une identification/autorisation valide pour l'URL visée, en fonction des données fournies par le vecteur d'identification.

> ✍ *L'association faite entre le vecteur d'identification et l'identification/autorisation locale devrait être tracée (le flux numéro 4 vers la base d'habilitation spécifique service).*

6.6.2. Interface d'entrée

6.6.2.1. Flux numéro 1 : la requête

L'environnement de la requête est modifié pour incorporer les éléments d'identification/d'autorisation que le service visé, à l'URL visée, requiert. La liste des PAGM est utilisée pour générer ces éléments.

6.6.2.2. Flux numéro 2 : l'URL visée

En conjonction avec les PAGM, l'URL visée permet de déterminer le choix d'identification/autorisation.

6.6.2.3. Flux numéro 3 : la liste des PAGM

Les PAGM donnés sont suffisants pour décider de l'identification/autorisation valide pour l'URL visée.

6.6.2.4. Flux numéro 4 : la base d'habilitation spécifique service

La base d'habilitation spécifique service contient l'identification/autorisation valide pour l'URL visée correspondant aux PAGM donnés.

6.6.2.5. Flux numéro 5 : le vecteur d'identification

Le vecteur est passé uniquement à des fins de traçage.

6.6.3. Interface de sortie

6.6.3.1. Flux numéro 1 : la requête avec identification/autorisation locale

La requête est complétée avec les éléments requis d'identification/autorisation locale pour accéder à l'URL visée.

7. Lot 4 : Traces

7.1. Présentation générale

Les traces à collecter puis à conserver dans le cadre de la fourniture d'une solution pour le standard sont de deux ordres :

- les traces techniques, provenant des briques techniques utilisées par le "constructeur de la solution",
- les traces de journalisation, stockées sous 2 formes :
  - un fichier sécurisé des traces, destiné à des utilisations à postériori dans un but d'audit,
  - une base de données des traces, accessible en temps réel.

Par ailleurs, et de façon complémentaire, chaque organisme veillera, dans le cadre de la mise en application du standard, à garder des traces de journalisation de ses propres systèmes (mais hors champs de la construction de la solution). Ces traces ne sont imposées (partiellement ou dans leur totalité) que si elles sont nécessaires à la compréhension des traces du standard. C'est, par exemple, le cas pour un système qui efface les données d'un utilisateur : pour que les traces (du standard) des requêtes passées restent valides, l'organisme de cet utilisateur doit conserver suffisamment d'informations pour pouvoir identifier cet utilisateur après avoir effacé ses données.

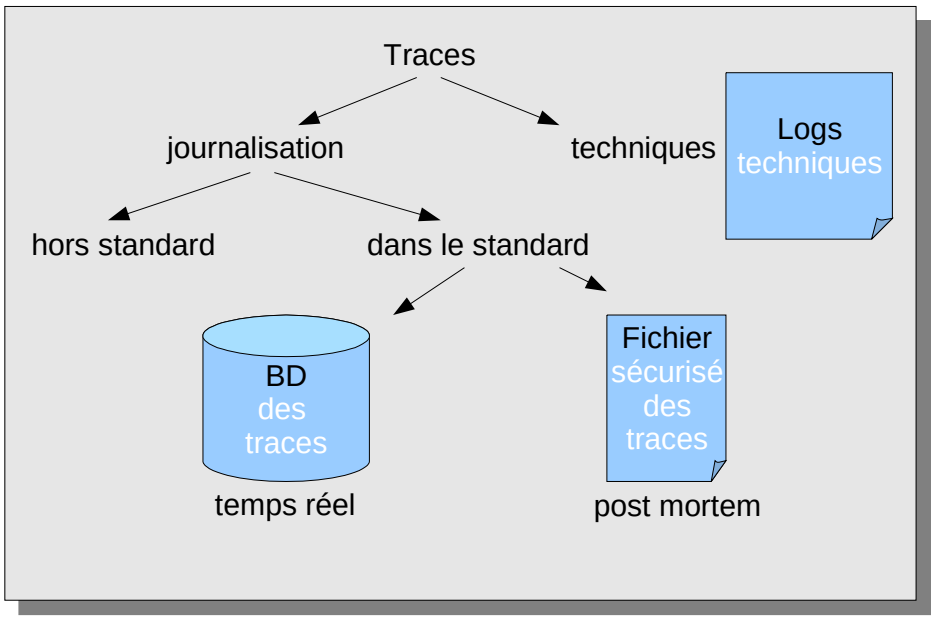


Figure 21 : Représentation de l'ensemble des traces

7.1.1. Eléments de traçage côté Organisme Client

Les éléments à tracer dans le cadre de la fourniture de la solution sont :

- le vecteur d'identification sous la forme d'assertion SAML signée. Cette trace est effectuée par le proxy après l'obtention du vecteur sous forme d'assertion SAML signée et avant l'envoi de la requête vers l'Organisme Fournisseur,

- l'identifiant réel du demandeur.

Les éléments à tracer par les organismes hors cadre de la fourniture de la solution sont ceux nécessaires à l'interprétation des éléments décrits ci-dessus. La liste ci-dessous donne l'ensemble des éléments possibles, à charge pour chaque organisme de définir ceux nécessaires à conserver pour l'interprétation :

- les mises à jour des définitions de services selon l'accord d'interopérabilité (URI des services, listes de PAGM associés ainsi que niveaux d'authentifications requis, dates d'application),

- dans le cadre de l'administration du système d'habilitation : les attributions de PAGM (identifiant local, niveau d'authentification, identifiant vecteur d'identification, identifiant dépersonnalisé, liste de PAGM attribués, date d'attribution, commentaire),

- dans le cadre opérationnel, lors de la création d'un vecteur d'identification : les attributions d'autorisations (identifiant local, niveau d'authentification, identifiant vecteur d'identification, identifiant dépersonnalisé, identifiant de l'organisme fournisseur, URI du service visé, liste de PAGM proposés, liste de PAGM retenus, date d'attribution, commentaire),

- l'identification de l'agent ou application requérant (identifiant local, date de connexion, date de requête, identifiant machine hôte, niveau d'authentification, information nominative),

- éventuellement, le traçage du système local concernant l'administration des agents et applications (ajout, modification, suppression d'identifiants agent/application, de même que rôles, niveaux d'authentification et autres informations qui seront utilisés lors de l'attribution des PAGM et autorisations, dates d'application).

7.1.2. Eléments de traçage côté Organisme Fournisseur

Les éléments à tracer dans le cadre de la fourniture de la solution sont :

- le vecteur d'identification sous forme d'assertion SAML signée. La trace est donc effectuée par le reverse proxy dès réception de la requête en provenance de l'Organisme Client,

- le nom réel du service.

Les éléments à tracer par les organismes hors cadre de la fourniture de la solution sont :

- les mises à jour des associations rôles applicatifs/PAGM (URI service, rôles applicatifs, PAGM, date d'application) –par Organisme Client (c'est à dire par accord d'interopérabilité),

- les requêtes d'accès (le vecteur d'identification, code résultat des vérifications, code résultat de la requête, date de la requête),

- association entre les PAGM des requêtes d'accès et des rôles applicatifs,

- éventuellement, le traçage du système local (ajout, modification, suppression d'identifiant application, rôles applicatifs/niveaux d'authentification requis, dates d'application).

7.1.3. Sécurisation des traces

Un mécanisme de sécurisation des traces doit être incorporé au module d'enregistrement des traces. Il prémuni contre les risques liés aux modifications à posteriori (quelles que soient les raisons des modifications).

Le mécanisme est une sécurisation logicielle basée sur les listes à n hachages. Il est encapsulé dans le module de trace.

> ✍ *Ce mécanisme ne présume pas des autres sécurisations que chaque organisme voudrait mettre en place, que ce soit matériel ou logiciel.*
>
> ✍ *Si la liste de hachage simple (hash-linking) est efficace pour détecter une incohérence dans la liste des traces (perte ou modification de donnée), elle est insuffisante car elle ne permet pas de réparer ou au moins de contourner l'incohérence car dès que l'incohérence est présente, la somme de contrôle de la liste ne peut plus être valide.*

7.1.4. Le fichier sécurisé des traces

Le fichier sécurisé des traces doit être accessible par un protocole normalisé. L'application "traces" doit pouvoir stocker de manière non-répudiable les données suivantes :

1. Identifiant unique d'enregistrement de trace,
2. Date d'insertion en table,
3. Le vecteur d'identification sous forme d'assertion SAML signée,
4. Identifiant réel du demandeur en ce qui concerne les traces côté Organisme Client et Nom réel de service en ce qui concerne l'Organisme Fournisseur.

La non-répudiation des données est essentielle : cela détermine la possibilité qu'a un organisme de pouvoir conduire une analyse des traces.

7.1.5. La base de données des traces

La base de données des traces comprend les mêmes éléments que le fichier sécurisé des traces, mais permet une consultation "temps réel" des traces de journalisation.

7.2. Le module enregistrement des traces

7.2.1. Rôle du module

Le rôle de ce module est d'enregistrer d'une manière sécurisée les traces. Son fonctionnement doit intégrer de manière transparente un mécanisme de sécurisation des traces permettant de conserver la cohérence des traces sur le long terme. Le mécanisme de choix d'utiliser des techniques de hachage.

Il convient d'implémenter ce module par un service de sécurité indépendant des applications ou des proxys.

7.2.2. Interface d'entrée

7.2.2.1. Flux numéro 1 : l'assertion SAML

Il s'agit de l'assertion signée, sous forme de chaîne de caractère.

7.2.2.2. Flux numéro 2 : l'identifiant du demandeur ou du service réel

Pour un Organisme Client il s'agit de l'identifiant non dépersonnalisé. Puisqu'il est possible que l''assertion SAML ne contienne qu'un identifiant dépersonnalisé, il est indispensable de fournir cet élément en accompagnement de l'assertion SAML.

Pour un Organisme Fournisseur il s'agit de l'identifiant réel du service visé puisque, du point de vue externe, le service publié (et donc visé par la requête cliente) est une abstraction par le reverse-proxy d'un service réel.

7.2.3. Interface de sortie

7.2.3.1. Flux numéro 1 : le fichier sécurisé des traces

Le fichier des traces contient l'ensemble des éléments fournis à ce service sécurisé par les mécanismes de hachage.

Il peut exister de multiples copies de ce fichier pour des raisons de sécurité.

7.2.3.2. Flux numéro 2 : la base de données des traces

Une représentation sous forme de tables dans une base de données est fournie permettant une visualisation et recherche en temps réel.

7.2.4. Eléments de configuration

Les fichiers de configuration dépendent de l'implémentation.

7.2.5. Performance

Le point critique de la performance est de s'assurer que les éléments sont proprement stockés dans les fichiers.

7.3. L'outil analyse des traces "post-mortem"

7.3.1. Rôle de l'outil

L'outil d'analyse de traces "post-mortem" doit être développé pour permettre l'exploitation des traces notamment lors d'un audit approfondit.

Il permet de valider la cohérence interne des traces, il permet aussi d'extraire un historique des actions de sécurisation des échanges entre organismes.

Il a les fonctions suivantes :

- Accéder au fichier sécurisé des traces,

- Lister tout ou partie des traces en fonction de critères de date d'insertion dans le fichier des traces et/ou de critères basés sur les éléments du vecteur d'identification tels que le service visé, l'identifiant de requérant (utilisateur ou application), de PAGM, etc.

- Vérifier tout ou partie des sommes de contrôles et relever les incohérences,

- Rendu du résultat selon différents modes de sortie (texte, HTML) et selon différents médiums de sortie (serveur HTTP, fenêtre graphique, console, fichier, imprimante).

7.3.2. Interface d'entrée

7.3.2.1. Flux numéro 1 : le fichier des Traces

L'accès au fichier des traces permet de réaliser toutes les fonctions de recherche des traces.

7.3.2.2. Flux numéro 2 : critères de recherche

Les critères d'analyses sont entrés par l'utilisateur.

7.3.3. Interface de sortie

7.3.3.1. Flux numéro 1 : résultat

Le résultat de la recherche ou de la vérification est rendu en fonction du médium de sortie et du mode de sortie.

7.3.4. Eléments de configuration

Sans objet.

7.3.5. Performance

Sans objet.

7.4. L'outil de visualisation de traces en temps réel

7.4.1. Rôle de l'outil

Il permet d'effectuer des recherches en temps réel dans la base de données des traces.

7.4.2. Interface d'entrée

7.4.2.1. Flux numéro 1 : la base de données des Traces

L'accès à la base de données des traces permet de réaliser toutes les fonctions de recherche des traces.

7.4.2.2. Flux numéro 2 : critères de recherche

Les critères d'analyses sont entrés par l'utilisateur avec une interface web conviviale.

7.4.3. Interface de sortie

7.4.3.1. Flux numéro 1 : résultat

Le résultat de la recherche ou de la vérification est rendu dans l'interface web.

7.4.4. Eléments de configuration

Sans objet.

7.4.5. Performance

Sans objet.

1370 # 8. GLOSSAIRE

1371

Nom	**Définition**
AAA	Authentication-Authorisation-Accounting (Authentification-Habilitation-Traçabilité).
AAS	Authentification-Autorisation-SSO.
Accord	Dans ce document le terme « accord » se réfère aux éléments techniques de la convention passée entre un Organisme Fournisseur et un Organisme Client. Ces éléments techniques servent à paramétrer les systèmes d'informations pour permettre l'accès au service mis à disposition par l'Organisme Fournisseur.
ADAE	Agence pour le développement de l'administration électronique cf. DGME
Administrateur privilégié	Dans ce document un administrateur privilégié est une personne désignée par son DN (lequel doit pouvoir être authentifié) et qui est habilité à créer des documents CPP à partir du système local et à configurer ce système local à partir de documents CPA.
Adresse locale	Une adresse locale est l'adresse qui permet de cibler une ressource à l'intérieur d'une zone d'adressage (par exemple un navigateur cible son proxy en utilisant une adresse locale).
Adresse externe	Une adresse externe est l'adresse qui permet de cibler une ressource à l'extérieur d'une zone d'adressage (par exemple un navigateur cible un site web externe avec une adresse externe). Dans le cadre du standard la différence entre l'adresse locale et l'adresse externe est importante du point de vue du proxy et du reverse-proxy.
AES	Advanced Encryption Standard (aussi nommé Rijndael) est un algorithme d'encryption basé sur des clefs symétriques.
Authentification	Processus visant à établir de manière formelle et intangible l'identification des parties à un échange ou une transaction électronique. Ce processus implique que les parties confirment et valident leur identification par des moyens techniques, tels que mot ou phrase de passe, un code secret, une réponse à un défi ou encore une signature numérique..
Autorisation	Mécanisme qui, à partir du vecteur d'autorisation, accorde ou non, à un utilisateur, l'accès à des applications, fonctions ou données spécifiques, en s'intéressant à des couples « objet, actions, conditions ».
Autorité de certification	Une AC est une entité qui délivre des certificats numériques pour utilisation par des tiers.

Nom	**Définition**
Base 64	Système d'encodage en ASCII (26 lettres minuscules + 26 lettres majuscules + 10 numériques + 2 caractères variables) de toute donnée numérique. Les deux caractères variables varient en fonction des systèmes. Ainsi pour le format MIME il s'agit de « + » et « / », pour les paramètres URL il s'agit de « * » et « - »,…
Bi-clé asymétrique	Ensemble des paramètres utilisés dans un algorithme cryptographique asymétrique. Une bi-clé asymétrique est composée d'un ensemble de paramètres rendus publics, globalement appelés la clé publique, et d'un ensemble de paramètres conservés secrets par le propriétaire de la bi-clé, et appelés la clé privée. Les deux ensembles de clés ont la propriété que, connaissant la clé publique, il est impossible par le calcul d'en déduire la clé privée.
Certificat numérique	De façon générique c'est un objet informatique logique qui permet de lier de façon intangible une identité d'entité à certaines caractéristiques de cette entité. Lorsqu'une des caractéristiques est une clé publique, on parlera de certificat de clé publique. Si ce n'est pas le cas on parlera de certificat d'attributs. Le lien est créé par la signature de l'ensemble des données du certificat par la clé privée de l'autorité qui émet le certificat Par extension on comprend que le certificat est l'ensemble formé par les données et par la signature de l'autorité sur ces données. La finalité première d'un certificat est de permettre à un utilisateur de vérifier l'authenticité (identité, caractéristique du propriétaire) de la clé publique qu'il va utiliser pour vérifier la signature produite par le signataire, en se basant sur la garantie apportée par l'autorité de certification. Par abus de langage, on utilise ce terme aussi pour l'ensemble d'un certificat d'identité et la clé privée
Certificat d'attribut	Un ensemble composé de l'identité d'une entité et d'attributs (caractéristiques) de cette entité, rendus indissociables par la signature du certificat d'attributs avec la clé privée de l'autorité de certification qui émet le certificat d'attributs.
Certificat d'identité	Un ensemble composé de l'identité d'une entité et d'une clé publique asymétrique (avec d'autres informations de gestion), rendus indissociables par la signature du certificat avec la clé privée de l'autorité de certification qui émet le certificat.
Convention	Dans le cadre de ce document il s'agit d'un ensemble d'éléments juridiques et techniques règlementant les échanges entre deux organismes.
CPA	Collaboration Protocol Agreement. Il s'agit d'un document de type ebXML résultant de la mise en commun de deux documents CPP et permettant de définir, entre les deux entités d'où sont issus les CPP, les modalités d'un échange de données numériques.

Nom	**Définition**
CPP	Collaboration Protocol Profile. Il s'agit d'un document de type ebXML permettant à une entité de décrire son système. La mise en commun de deux documents CPP permet de créer un document CPA.
CPPA	Collaborative Partner Profile Agreement. Il s'agit du standard (ISO 15000-1) ebXML décrivant un protocole d'échange d'informations basé sur XML entre deux entités. Il utilise les documents de type CPP et CPA.
DGME	Direction générale pour la modernisation de l'Etat.
Distinguished Name (DN)	Terme de la norme X.500 concernant les annuaires Il s'agit d'un nom non-ambiguë) désignant une entrée dans l'arborescence de l'annuaire. Il est composé d'un RDN (Relative Distinguished Name) construit à partir d'attributs de l'entrée et du DN de l'entrée parente.
DNS	Domain Name Server. Serveur (application tournant sur un ordinateur) dont le rôle premier est de convertir les noms de domaines lisibles par l'homme par les adresses entières numériques IP auxquelles ils correspondent.
Droit	Un droit correspond à l'habilitation d'un métier dans une application et se compose d'un ou plusieurs groupes d'actions unitaires.
DSIG	Digital Signature. Ce terme est employé, dans le cadre XML, pour désigner les extensions de protocole permettant d'inclure des signatures numériques (par exemple l'extension SOAP-DSIG).
ebXML	Electronic Business eXtensible Markup Language. Il s'agit d'un standard créé par l'OASIS (norme ISO-15000) permettant de décrire des règles d'échange de données pour le commerce électronique.
Habilitation	Ensemble d'attributs attachés à une entité et autorisant cette entité à accéder à des ressources.
Hachage	Méthode mathématique permettant de générer une empreinte ou somme de contrôle à partir d'un ensemble de données. Une fonction de hachage H a les propriétés suivantes : si $H(x) \neq H(y)$ alors $x \neq y$ et si $H(x) = H(y)$ alors il est fortement probable que $x = y$.
Hash-linking	Méthode permettant de lier des éléments d'une liste à l'aide d'une fonction de hachage : la somme de contrôle d'un élément de la liste est calculée en fonction de cet élément et de la somme de contrôle de l'élément précédent.
HTTP	HyperText Transfer Protocol
Identification	Processus permettant d'affecter un identifiant à une entité.

Nom	**Définition**
IETF	Dans le cadre de l'Internet Society, Comité exécutif regroupant des ingénieurs et des chercheurs du monde entier, chargés de la normalisation des standards proposés par les utilisateurs et qui seront ultérieurement introduits sur Internet. Leurs buts sont de définir de nouveaux standards d'Internet et faire évoluer les réseaux internationaux et de comprendre les besoins futurs de l'utilisateur.
IGC	Infrastructure de Gestion de Clefs. Ensemble d'entités, de fonctions et procédures permettant de gérer des clefs et certificats numériques dans le cadre de services de sécurité à base de cryptographie à clef publique. Une IGC met notamment en œuvre les autorités de séquestre pour répondre aux exigences de la loi française.
Interconnexion	Dans le cadre de ce document il s'agit des règles permettant aux réseaux des organismes d'accéder les uns aux autres.
Interopérabilité	Faculté que possèdent des services ou des composants hétérogènes de fonctionner conjointement. L'une des conditions fondamentales permettant la communication entre ces services et ces composants est l'utilisation de langages et de protocoles communs. Par exemple, les protocoles SOAP ou XML sont normalisés et permettent aux différents services web d'échanger des informations selon les mêmes règles et les mêmes méthodes. Dans le cadre de ce document il s'agit des règles permettant aux utilisateurs et aux applications des organismes d'interagir, en particulier en respectant des contraintes de sécurité forte et de souplesse d'administration.
IOPS	Interopérabilité entre les Organismes de Protection Sociale.
Journal	Collection d'enregistrements chronologiques de l'activité d'un système qui est suffisante pour permettre la reconstitution et l'analyse de la séquence des états et activités entourant ou déterminant chaque événement dans le cheminement d'une transaction depuis son initialisation jusqu'à la sortie des résultats finaux.
LDAP	Lightweight Directory Access Protocol. Il s'agit d'un protocole réseau permettant de quérir et modifier des annuaires. Ces annuaires sont structurés en arborescences d'informations, chaque entrée étant composée d'un ensemble de paires attribut-valeur.

Nom	**Définition**
MIME	La norme MIME, extension de SMTP, permet d'inclure directement sous forme de “pièce-jointe” n'importe quel fichier binaire qui se trouve dans la messagerie e-mail d’Internet (texte, image, son, vidéo). Un logiciel de messagerie MIME permet d'envoyer et de recevoir des messages électroniques qui contiennent ces types de documents. Le récepteur peut ainsi les ouvrir sous forme de document ou les exécuter en tant que programmes. Ce protocole est indépendant du support de transmission et donc du réseau utilisé.
Nommage	Processus d’attribution d’un nom unique associé à l’abonné dans le processus d’enregistrement de celui-ci. Pour un domaine donné, c’est à dire pour l’ensemble des abonnés gérés par une AC, il ne doit pas y avoir d’ambiguïté. On parle souvent de nom distinctif (distinguished name) comme utilisé dans la norme X.500. D’autre type de nommage peut être utilisé comme le DNS
OASIS	Organization for the Advancement of Structured Information Standards. Cette organisation est un consortium international à but non lucratif et dont l’objet est de favoriser et mener le développement, la convergence et l’adoption de standards pour le commerce électronique. Cela inclut, par exemple, ebXML et SAML.
Open Source	Terme caractérisant des logiciels et leurs licences associées. Ces licences stipulent principalement que les logiciels licenciés doivent être redistribuables librement, le code source doit être inclus dans la distribution ou librement accessible et que les modifications ou travaux dérivés sont librement distribuables.
Opérateur de certification	Entité gérant des certificats d’autorité de certification en lieu et place d’autres entités. L’opérateur de certification génère alors des certificats au nom des autres entités.
Organisme Client	Dans le cadre de ce document il s’agit d’un organisme ayant passé une convention d’interopérabilité avec un autre organisme dans le but d’accéder à un service mis à disposition de cet autre organisme.
Organisme Fournisseur	Dans le cadre de ce document il s’agit d’un organisme ayant passé une convention d’interopérabilité avec un autre organisme dans le but de mettre un service à disposition de cet autre organisme.
PAGM	Dans le cadre de ce document le Profil Applicatif Générique Métier est défini: Il s’agit d’un profil spécifique à l’interopérabilité et dont l’attribution à un utilisateur d’un Organisme Client permet à cet utilisateur d’accéder à un service d’un Organisme Fournisseur, à la condition que les deux organismes aient passé une convention associant ce PAGM au service en question.

Nom	**Définition**
Passerelle	Système consistant à relier deux réseaux et donc deux systèmes d'adressage entre eux.
Proxy	Dispositif informatique associé à un serveur et réalisant, pour des applications autorisées, des fonctions de médiation, telle que le stockage des documents fréquemment demandés ou l'établissement de passerelles. Il a généralement un rôle de sécurité et de filtrage, et d'antémémoire / mémoire cache (optimise les performances d'accès à des pages Internet fréquemment consultées). Dans le cadre de ce document le proxy est un dispositif fonctionnant dans l'organisation de l'Organisme Client et assurant les fonctions de passerelle en ce qui concerne l'adressage de service en local et en externe ainsi que les fonctions d'habilitation pour demander l'accès aux services des Organismes Fournisseurs.
Publication	Dans le cadre de ce document action pour un Organisme Client ou un Organisme Fournisseur de délivrer des informations techniques sur son SI dans le but de former, avec un organisme partenaire, une convention d'échange de données.
Reverse Proxy	Dans le cadre de ce document le reverse proxy est un dispositif fonctionnant dans l'organisation de l'Organisme Fournisseur et assurant les fonctions de passerelle en ce qui concerne l'adressage de service en local et en externe ainsi que les fonctions de vérification d'habilitation pour permettre l'accès aux services de cet Organisme Fournisseur.
SAML	Security Assertion Markup Language. Il s'agit d'un format en XML d'échange d'informations liées à la sécurité. SAML a été créé par l'OASIS.
Service	Dans le cadre de ce document il s'agit d'un ensemble de fonctions regroupées et mise à disposition d'Organismes Clients par un Organisme Fournisseur. Un service est publié sous une dénomination type DNS (nom-de-service.organisme.domaine).
Service visé	Dans le cadre de ce document il s'agit d'un service ou d'un sous-groupe de fonctions de ce service que l'utilisateur d'un Organisme Client cherche à obtenir auprès d'un Organisme Fournisseur. Un service visé est nommé soit par la dénomination type DNS (nom-de-service.organisme.domaine) ou par la dénomination type DNS suivi d'un préfixe de chemin (nom-de-service.organisme.domaine/préfixe) permettant de spécifier le sous-groupe de fonctions ciblé.
SI	Système d'Information.
Signature numérique	Transformation cryptographique de données déterminée par avec une clé privée afin de fournir les services d'authentification de l'origine, d'intégrité des données et, sous certaines conditions, pouvant garantir la non répudiation par le propriétaire de la clé privée.

Nom	**Définition**
SOAP	Simple Object Access Protocol. SOAP est un protocole d'échange de messages sur un réseau, au format XML. SOAP permet de faciliter les architectures orientées services à travers un réseau.
SSL	Secured Socket Layer. Protocole de sécurisation d'une session de présentation au niveau de TCP. SSL permet, au minimum, l'authentification de l'une des parties. Il permet éventuellement l'authentification mutuelle ainsi que le chiffrement du canal de communication entre les deux parties. La version 3 de SSL normalisée par l'IETF s'appelle TLS.
SSO	Single Sign-On (authentification unique). Technique d'authentification permettant à une entité de s'authentifier une seule fois pour l'accès à un ou plusieurs services.
Standard	Dans le cadre de ce document il s'agit des règles définies dans le document [R1] pour permettre l'interopérabilité entre les organismes de protection sociale.
TLS	Transport Layer Security. Il s'agit de la normalisation de la version 3 de SSL.
Transcription de vecteur	Dans le cadre de ce document il s'agit du processus, chez l'Organisme Fournisseur, permettant de transcrire les informations d'habilitation contenues dans le vecteur d'identification en informations d'habilitation locales à l'Organisme Fournisseur.
URI	Uniform Resource Identifier. Il s'agit d'une courte chaine de caractères identifiant de façon unique une ressource physique ou abstraite. l'URI est spécifiée par la RFC 3986 de l'IETF.
URL	Uniform Resource Locator. Il s'agit d'un sous-ensemble d'URI permettant d'identifier et de localiser une ressource. L'URL est spécifiée par la RFC 1738 de l'IETF.
Vecteur d'Identification (VI)	De manière générale il s'agit d'un ensemble d'éléments caractérisant une entité dont un élément d'identification, un élément d'authentification et un ensemble d'attribut. Dans le cadre de ce document, le vecteur d'identification sert à transporter les PAGM entre les organismes en tant qu'habilitation d'accès aux services de l'Organisme Fournisseur permettant de réaliser l'interopérabilité.
X509	Norme relative aux certificats d'identité et d'attribut, issue de la Recommandation X509 de l'ITU-T (International Telecommunication Union Telecomminucation standardization sector).
XML	eXtented Markup Language est un langage général de type Markup permettant de créer des langages Markup spécialisés.

Nom	**Définition**
Web Service	Système logiciel permettant le support d'interaction machine-à-machine à travers un réseau et utilisant des messages. Typiquement le protocole SOAP permet de réaliser cette interaction.

FIN DU DOCUMENT