Liberté • Égalité • Fraternité

RÉPUBLIQUE FRANÇAISE

MINISTÈRE DU TRAVAIL, DE l'EMPLOI ET DE LA SANTÉ

MINISTÈRE DES SOLIDARITÉ ET DE LA COHÉSION SOCIALE

MINISTÈRE DU BUDGET, DES COMPTES PUBLICS ET DE LA RÉFORME DE L'ÉTAT

Spécifications détaillées du mode « application à application »

Standard d'interopérabilité entre organismes de la sphère sociale

Réf. : Standard Interops-A2.0_SpécificationsDétaillées
Version 2.0 du 05/04/2012

Référence : **Version :** **Date de dernière mise à jour :**	Standard Interops-A2.0_SpécificationsDétaillées 2.0 05/04/2012
Niveau de confidentialité :	PUBLIC

Table des mises à jour du document

N° de version	Date	Auteur	Objet de la mise à jour
2.0	05/04/12	Groupe de travail Interops	Version pour diffusion

SOMMAIRE

SOMMAIRE 3
1. INTRODUCTION 6
1.1 Objet du document 6
1.2 Relation avec d'autres documents 6
1.3 Organisation et structure du document 6
1.4 Références 7
1.4.1 Documents internes 7
1.4.2 Documents externes 7
1.5 Conventions 8
2. PRINCIPES GENERAUX 9
2.1 Modélisation des échanges 9
2.2 Vecteur d'identification 10
3. FONCTIONNEMENT GENERAL 11
3.1 Architecture générale 11
3.1.1 Découpage fonctionnel modulaire 11
3.1.2 Eléments génériques et spécifiques 11
3.1.3 Boîtes à outils 12
3.1.4 Schéma d'architecture 12
3.1.5 Description des éléments d'architecture 13
3.2 Sécurité des échanges 16
3.2.1 Filtrage TCP/IP 17
3.2.2 Utilisation des bi-clés / certificats 17
3.2.3 Protection du vecteur d'identification 17
3.2.4 Authentification et confidentialité des échanges 18
3.2.5 Protection contre le rejeu 18
3.3 Web service sécurisé et SOAP 19
3.4 Structures applicatives Web Service 19
3.5 Eléments techniques représentant les accords 20
3.6 Administration 20
3.7 Interconnexion réseau, adressage et présentation de service 21
3.7.1 Interconnexion réseau 21
3.7.2 Dénomination de service 22
3.8 Traces 23

3.8.1 Traces d'audit 24
3.8.2 Traces techniques 24
3.9 Gestion des erreurs 25
3.9.1 Utilisation de soap Fault 25
3.9.2 Normes de référencement des erreurs 26
3.9.3 Filtrage des erreurs 27
3.9.4 Lien avec les traces 28
3.10 Synchronisation temporelle 28

4. LOTS A DEVELOPPER 29
4.1 Lot 1 : Administration des accords 29
4.2 Lot 2 : Vecteur et proxy organisme client 29
4.3 Lot 3 : Vecteur et reverse proxy organisme fournisseur 29
4.4 Lot 4 : Traces 29

5. LOT 1 : OUTILS D'ADMINISTRATION DES ACCORDS 30
5.1 Outil de création des accords 30
5.1.1 Rôle de l'outil 30
5.1.2 Cinématique générique 30
5.1.3 Interface d'entrée 30
5.1.4 Interface de sortie 31
5.2 Outil de mise en œuvre des accords 31
5.2.1 Rôle de l'outil 31
5.2.2 Interface d'entrée 31
5.2.3 Interface de sortie 32

6. LOT 2 : VECTEUR ET PROXY ORGANISME CLIENT 33
6.1 Intégration du VI par l'application cliente 33
6.1.1 Description du scénario 33
6.1.2 Composants utilisés 33
6.1.3 Diagramme de séquence nominal 33
6.2 Intégration du VI par le proxy 35
6.2.1 Description du scénario 35
6.2.2 Composants utilisés 35
6.2.3 Diagramme de séquence nominal 36

7. LOT 3 : VECTEUR ET REVERSE-PROXY ORGANISME FOURNISSEUR 38
7.1 Validation du VI et réponse du service visé 38
7.1.1 Composants utilisés 38
7.1.2 Diagramme de séquence nominal 38

8. LOT 4 : TRACES 41

8.1 Présentation générale 41
8.1.1 Eléments de traçage côté organisme client 41
8.1.2 Eléments de traçage côté organisme fournisseur 42
8.1.3 Sécurisation des traces 43
8.1.4 Processus de consolidation 43
8.2 Le module d'enregistrement des traces 43
8.3 L'outil de gestion des traces 44
9. ANNEXES 45
9.1 Acronymes 45
9.2 Glossaire 45

1. INTRODUCTION

1.1 Objet du document

Ce document présente les spécifications détaillées du Standard d'Interopérabilité des Organismes de la Sphère Sociale [R1] pour le mode « application à application ».

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 documents que d'applications du standard.

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 « **Principes généraux** » présente macroscopiquement le mode « application à application » du Standard d'Interopérabilité des Organismes de la Sphère Sociale
- Le chapitre 3 « **Fonctionnement général** » définit le périmètre des spécifications et apporte des éclairages sur les contraintes d'implémentation du standard,
- Le chapitre 4 « **Lots à développer** » présente les blocs fonctionnels à développer
- Le chapitre 5 « **Lot 1 : Outils d'administration des accords** » décrit les spécifications détaillées du lot concernant les accords d'interopérabilité
- Le chapitre 6 « **Lot 2 : Vecteur et proxy organisme client** » décrit les spécifications détaillées du lot côté organisme client concernant la création du vecteur d'identification, sa propagation du et la propagation des requêtes des applications clientes
- Le chapitre 7 « **Lot 3 : Vecteur et reverse-proxy organisme fournisseur** » présente les spécifications détaillées du lot concernant la réception, la manipulation du vecteur d'identification du côté de l'Organisme Fournisseur et le traitement des requêtes provenant des organismes clients
- Le chapitre 8 « **Lot 4 : Traces** » représente les spécifications détaillées du lot concernant l'enregistrement et l'analyse des traces.
- Le chapitre 9 « **Annexes** » rassemble les annexes de ce document

121 ## 1.4 Références

122 ### 1.4.1 Documents internes

Référence		Titre	Auteur	Ver.	Date
[R1]	Standard Interops2.0_SpecificationsFonctionnelles	Spécifications fonctionnelles	Groupe de travail Interops	2.0	05/04/2012
[R2]	Standard Interops2.0_SpecificationsVI	Spécifications du Vecteur d'Identification	Groupe de travail Interops	2.0	05/04/2012
[R3]	Standard Interops2.0_ConventionTechnique	Convention technique	Groupe de travail Interops	2.0	05/04/2012
[R4]	Standard Interops2.0_Glossaire	Glossaire du standard Interops	Groupe de travail Interops	2.0	05/04/2012

123 ### 1.4.2 Documents externes

	Titre	Auteur	Date
[RGS]	Référentiel Général de Sécurité version 1.0	ANSSI/DGME	06/05/2010
[RGS_A_14]	Référentiel Général de Sécurité version 1.0 Annexe A14 : Profils de Certificats / LCR / OCSP et Algorithmes Cryptographiques	ANSSI/DGME	11/02/2010
[RGS_B_1]	Référentiel Général de Sécurité version 1.0 Annexe B1 : Mécanismes cryptographiques	ANSSI/DGME	26/01/2010
[SOAP]	Simple Object Access Protocol (SOAP) 1.1	Andrew, Mendelsohn, Noah, Nielsen, HenrikFrystyk, Winer, Dave, eds.	08/03/2000
[TLS]	RFC 4346 - The Transport Layer Security (TLS) Protocol Version 1.1	T. Dierks, E. Rescorla	Avril 2006
[WSI]	Basic Profile Version 1.1	Keith Ballinger, David Ehnebuske, Christopher Ferris, Martin Gudgin, Canyang Kevin Liu, Mark Nottingham, Prasacd Yendluri	10/04/2006
[WSS]	Web Services Security: SOAP	Hallam-Baker, Phillip,	Janvier

	Message Security," OASIS Standard V1.0	Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds	2004
[XMLDsig]	XML-Signature Syntax and Processing	Eastlake, Donald, Reagle, Joseph, Solo, David, eds.	12/02/2002

1.5 Conventions

Sauf indication contraire, toutes les spécifications précisées par ce document sont OBLIGATOIRES (« MUST »).

2. PRINCIPES GENERAUX

Le standard d'interopérabilité entre les organismes pour le mode « application à application » repose sur deux principes :

- L'utilisation des technologies Web Services sécurisés basées sur SOAP et WS-Security
- La mise en coupure d'un proxy et d'un reverse-proxy pour la sécurisation des flux entre les deux organismes

Cette solution permet de répondre aux exigences émises par les OPS :

- Le modèle repose sur la confiance entre les organismes
- L'authentification de l'utilisateur ou de l'application n'est pas effectuée de bout en bout mais est réalisée par l'organisme client
- L'habilitation est attribuée par l'organisme client à ses utilisateurs ou applications clientes en respectant les règles établies avec l'organisme fournisseur (Convention)
- L'habilitation est transmise à l'organisme fournisseur de manière sécurisée (par un Vecteur d'identification)
- Toute création de vecteur d'identification est auditable afin d'en permettre le contrôle « a posteriori »

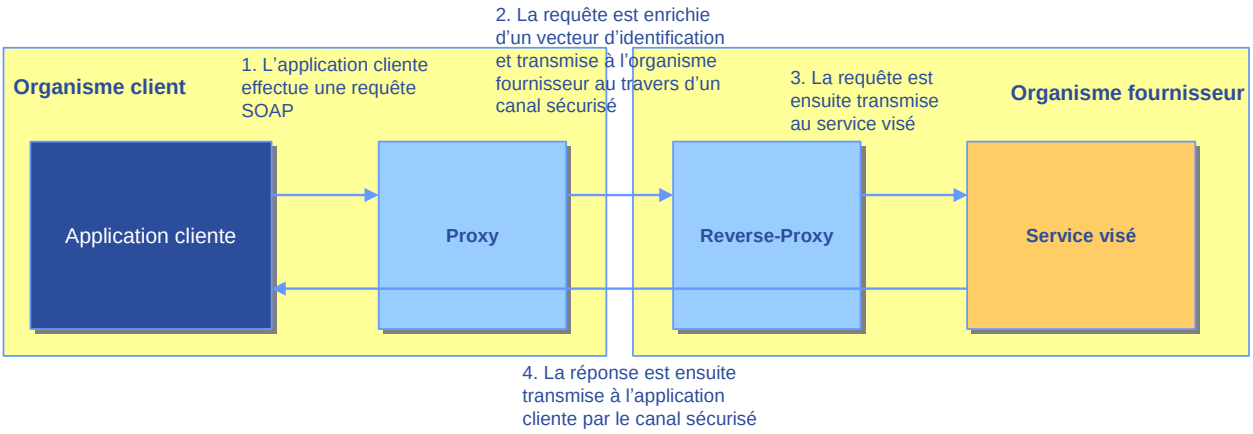
2.1 Modélisation des échanges

Le standard permet la communication entre une application cliente et un service visé, sur un serveur applicatif, par l'utilisation conjuguée de SOAP [SOAP] et de WS-Security [WSS] pour attacher un VI aux requêtes.

L'application cliente peut envoyer des requêtes à un service externe à l'organisme client pour son compte ou pour le compte d'un utilisateur.

En guise d'exemple, l'application peut correspondre à un batch, une application lourde sur le poste d'un agent, ou encore un portail Web.

La cinématique d'échange est décrite sur le schéma ci-dessous :



1. L'application cliente effectue une requête à destination d'un service offert par un organisme fournisseur avec lequel il existe un accord.
2. Toute requête sortante de l'organisme client possède un entête WS-Security incluant un vecteur d'identification. L'enrichissement de la requête par le VI peut être le fait de

l'application cliente ou du proxy. La communication entre le proxy de l'organisme client et le reverse-proxy de l'organisme fournisseur est sécurisée par une authentification mutuelle et un chiffrement des données.

3. La requête est transmise au service visé.
4. Le service visé traite la requête et génère une réponse qui est transmise jusqu'à l'application cliente en traversant le canal sécurisé formé entre le proxy de l'organisme client et le reverse-proxy de l'organisme fournisseur.

2.2 Vecteur d'identification

Les spécifications du vecteur d'identification pour le mode « application à application » sont décrites dans le document [R2].

3. FONCTIONNEMENT GENERAL

3.1 Architecture générale

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.

3.1.1 Découpage fonctionnel modulaire

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

- L'administration des accords d'interopérabilité
- La manipulation des vecteurs d'identification côté organisme client
- La fonction de proxy côté organisme client
- La manipulation des vecteurs d'identification côté organisme fournisseur
- La fonction de reverse-proxy côté organisme fournisseur
- La gestion des traces

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

3.1.2 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 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

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

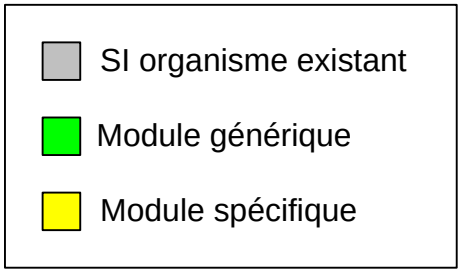


Figure 1 : Code couleur des schémas

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.

3.1.3 Boîtes à outils

La mise en œuvre des blocs fonctionnels décrits dans les spécifications détaillées doit répondre à une logique de boîte 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.

3.1.4 Schéma d'architecture

L'architecture d'un organisme client est représentée sur la Figure 2.

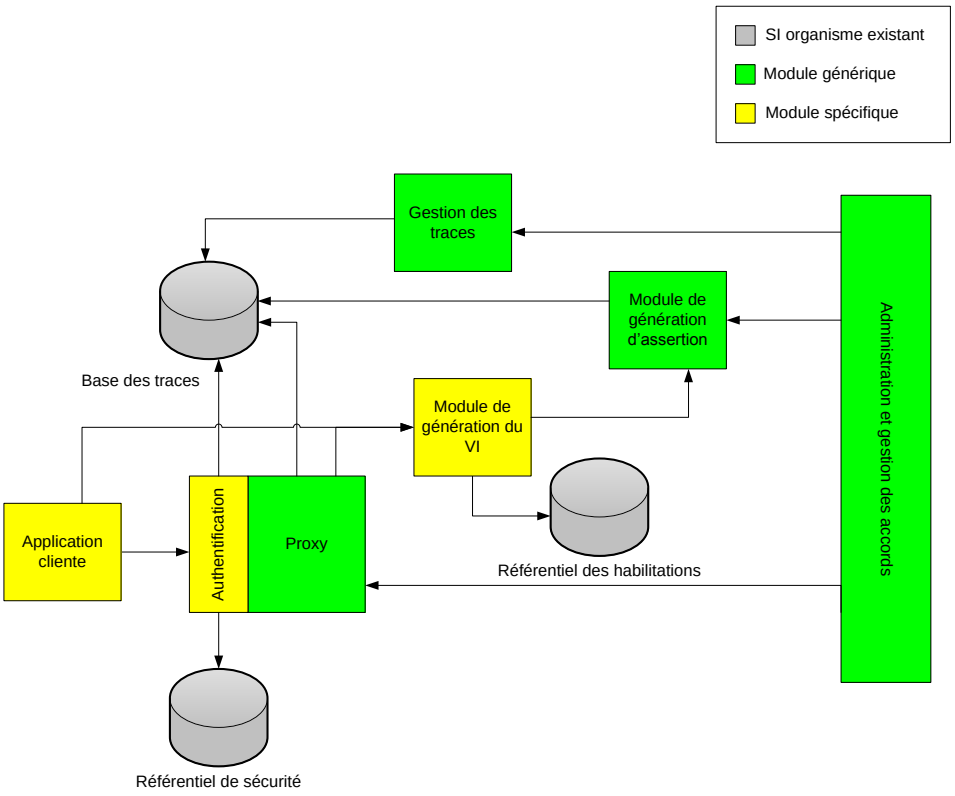


Figure 2 : Architecture générale d'un organisme client

L'architecture d'un organisme fournisseur est représentée sur la Figure 3.

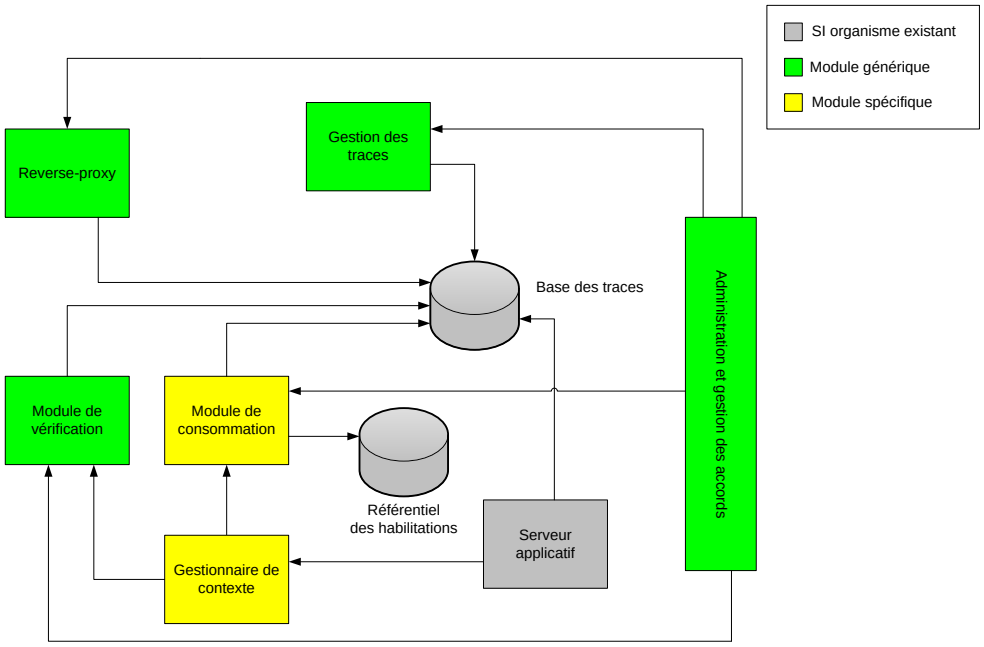


Figure 3 : Architecture générale d'un organisme fournisseur

Une description des éléments d'architecture est faite au paragraphe 3.1.5 *Description des éléments d'architecture*.

3.1.5 Description des éléments d'architecture

3.1.5.1 Référentiel de sécurité

Le référentiel de sécurité contient les utilisateurs (dont les applications clientes) du système :

- Leurs authentifiants (login et mot de passe par exemple)
- Leurs informations relatives (nom, prénom, etc.)

Il est utilisé par le module d'authentification pour authentifier les différents utilisateurs.

3.1.5.2 Référentiel des habilitations

Le référentiel des habilitations contient les droits des utilisateurs (dont les applications clientes) sur des ressources.

Il est utilisé par le module de génération du vecteur d'identification pour calculer les droits d'accès d'un utilisateur ou d'une application à une application distante, hébergée par un organisme fournisseur, et déterminer ses PAGM.

Il est utilisé par le module de consommation pour déterminer les droits d'accès d'un utilisateur ou d'une application cliente à une application locale à partir des PAGM présentés.

3.1.5.3 Module de génération du VI

Le module de génération du vecteur d'identification peut être appelé en fonction des choix d'implémentation fait par les organismes clients par :

- L'application cliente elle-même
- Le proxy

A partir des informations passées, le module de génération du VI peut élaborer les éléments constituant le VI. Il est responsable de la détermination des PAGM de l'utilisateur ou de l'application cliente à partir du référentiel d'habilitation.

3.1.5.4 Module de génération d'assertion

Le module de génération d'assertion permet de construire l'assertion conforme aux spécifications du standard et à l'accord passé entre l'organisme client et l'organisme fournisseur.

Chaque génération d'assertion est tracée conformément aux accords inter-organismes.

3.1.5.5 Module d'authentification

Le module d'authentification s'appuie sur le référentiel de sécurité pour authentifier les applications appelantes. Il transmet l'identité de l'utilisateur (application ou utilisateur)aux modules qui en dépendent, l'identification des utilisateurs étant nécessaire pour la traçabilité et la génération des VI appropriés.

Le module d'authentification est indispensable lorsque le proxy est chargé de la génération du VI, puisque l'identité de l'application appelante doit être connue pour générer le VI. Le module d'authentification devient également nécessaire en fonction de la traçabilité désirée.

Dans tous les cas, il devra au minimum tracer les authentifications réussies et échouées et les éléments associés :

- Identifiant de l'utilisateur ou de l'application cliente
- Date
- Méthode d'authentification

3.1.5.6 Proxy

Le proxy permet aux applications clientes, après authentification, de communiquer avec l'application hébergée par l'organisme fournisseur. Il se place ainsi en coupure de la communication entre l'application cliente et l'organisme fournisseur et peut tracer les URL d'accès des applications clientes.

Il réalise une authentification mutuelle avec le reverse-proxy de l'organisme fournisseur (cf. paragraphe 3.1.5.11 *Reverse-proxy*) de manière à certifier l'origine du flux HTTP et permet une traçabilité des échanges côté organisme client, puisqu'il se situe en coupure de tous les échanges.

3.1.5.7 Administration et gestion des accords

L'administration doit permettre d'appliquer les accords passés entre les organismes :

- Configuration des certificats de signature
- Configuration des certificats d'authentification client et serveur
- Configuration du format du VI pour un service donné :
  - Configuration de la version de l'accord
  - Configuration des identifiants des organismes
  - Configuration des PAGM possibles
  - Configuration de la durée de validité des assertions
  - Configuration des attributs supplémentaires nécessaires

- Déclaration des services visés
- Configuration de la politique de traces
- Etc.

Bien qu'ici toutes les fonctions soient regroupées, pour des raisons techniques, l'interface d'administration pourra être découpée par module à administrer (les traces, le proxy, etc.).

Ce module ne permet cependant pas d'attribuer les habilitations aux utilisateurs finaux ou aux applications clientes.

3.1.5.8 Gestion des traces

Le module de gestion des traces permet d'administrer la politique de trace conformément aux accords inter-organismes et doit permettre de :

- Appliquer la politique de trace conformément aux accords inter-organismes.
- Consulter les traces / effectuer des recherches multicritères
- Archiver les traces
- Effacer les traces expirées (automatiquement ou manuellement)
- Consolider des traces suites à une demande
- Préparer une demande de rapprochement en fournissant une liste d'identifiants de PAGM
- Etc.

3.1.5.9 Module de vérification

Le module de vérification permet de vérifier la conformité d'une assertion SAML aux accords :

- Identifiants utilisés
- PAGM employés et autres attributs présents
- Version de l'accord
- Certificat de signature employé
- Validité de la signature
- Etc.

3.1.5.10 Module de consommation

Le module de consommation permet de traduire un VI transmis par un organisme client en un contexte de sécurité utilisable par le service visé.

Par exemple, Il est chargé de

- Identifier l'application cliente ou l'utilisateur
- Traduire les PAGM contenus dans l'assertion en un profil applicatif avec les habilitations correspondantes.

Le module de consommation doit tracer la traduction des informations contenues dans une assertion SAML dans le contexte de sécurité afin de faire le lien entre le VI et l'identité locale à l'organisme fournisseur.

3.1.5.11 Reverse-proxy

En plus d'une fonction de sécurité évidente de protection du SI, le reverse-proxy authentifie et déchiffre le flux entrant et vérifie les habilitations d'accès d'un organisme client.

Puisqu'il se situe en coupure de tous les échanges, il permet une traçabilité technique des échanges côté organisme fournisseur.

3.1.5.12 Base des traces

La base des traces contient les traces. Elle est alimentée par les différents composants intervenant dans les échanges inter-organismes :

- Proxy
- Reverse-proxy
- Module de génération d'assertion
- Module de vérification
- Module de consommation
- Serveur applicatif

Elle est accédée en lecture par le gestionnaire de trace.

3.1.5.13 Serveur applicatif et service visé

Le serveur applicatif héberge le service visé.

Il est capable en sus des traces d'audit de tracer les tentatives d'accès d'une identité, c'est-à-dire possédant un contexte de sécurité.

3.1.5.14 Gestionnaire de contexte

Le gestionnaire de contexte permet de :

- Vérifier les éléments du VI
- Créer un contexte de sécurité à partir des informations du VI
- Associer le contexte de sécurité à une identité pour le serveur applicatif

3.2 Sécurité des échanges

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-clés numériques {clé publique, clé privée}
- Par ailleurs, les communications entre organismes sont chiffrées et authentifiées par la technique TLS v1.0 ou SSLv3.

Les échanges doivent être sécurisés par des moyens classiques tels qu'un filtrage au niveau TCP/IP.

> ***Dans le cas où les échanges devront être sécurisés en utilisant des mécanismes conformes au Référentiel Général de Sécurité, les moyens cryptographiques utilisés devront suivre les préconisations contenues dans le***

[RGS]. En particulier, les tailles de clés et algorithmes utilisés devront respecter [RGS_B_1] et les profils de certificats devront s'appuyer sur [RGS_A_14].

3.2.1 Filtrage TCP/IP

La vérification d'un certificat (cf. paragraphe 3.2.2 *Utilisation des bi-clés / certificats*) est une opération qui peut être lourde en termes de calcul. Les infrastructures ne mettant en œuvre que cette protection pourraient donc être peu résistantes à des attaques de type « déni de services » qui tenteraient des connexions avec des certificats clients invalides.

Toutes les communications entre l'organisme client et l'organisme fournisseur sortent de l'organisme client par son ou ses proxys et rentrent par le ou les reverse-proxys de l'organisme fournisseur.

La liste de ces composants avec leur adressage IP et la liste des ports mis en jeu doivent être définies afin d'établir des règles de filtrage. Ces règles pourront être appliquées sur les composants de sécurité périphériques des organismes clients et des organismes fournisseurs, tels que les firewalls.

3.2.2 Utilisation des bi-clés / certificats

Dans le standard, chaque organisme client devra posséder au moins un certificat d'authentification SSL client et un certificat de signature, et chaque organisme fournisseur un certificat d'authentification SSL serveur.

Les scénarios de distribution des certificats n'entrent pas dans les spécifications du standard.

Néanmoins, chaque organisme devra être à même de vérifier la validité du ou des certificats de son partenaire.

La vérification d'un certificat comprend la validation des points suivants :

- La date de validité du certificat est correcte
- Le certificat a été émis par une chaîne de certification de confiance
- Le certificat n'a pas été révoqué
- L'usage du certificat correspond bien à l'emploi qui en est fait

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.

Concernant la conformité au RGS, se reporter à la note du paragraphe 3.2.

3.2.3 Protection du vecteur d'identification

Comme spécifié dans [R2], le vecteur d'identification sera signé numériquement afin d'assurer :

- Un contrôle d'intégrité au moment de la transmission
- Une authentification de l'organisme client
- Une non-répudiation de l'organisme client
- Une valeur probante après archivage

Les organismes doivent donc disposer au moins d'un certificat numérique X509 de signature à cette fin.

> ***La signature « cachet serveur » est une fonction de sécurité faisant appel à des mécanismes cryptographiques qui peut nécessiter d'être conforme au RGS. Concernant cette conformité, se reporter à la note du paragraphe 3.2.***

3.2.4 Authentification et confidentialité des échanges

Dans le cas où la liaison entre les organismes client et fournisseur n'est pas une liaison sécurisée de type « point à point », un canal sécurisé doit être mis en place. Dans ce cas, l'authentification mutuelle et la confidentialité des échanges entre les organismes client et fournisseur doit s'appuyer sur les éléments suivants :

- Le protocole TLS ou SSL v3 (avec une méthode de chiffrement AES 128 bits).
- Le proxy de l'organisme client
- Le reverse-proxy de l'organisme fournisseur

Pour une authentification mutuelle de serveur et de client chaque partenaire doit disposer d'au moins un certificat numérique X509 d'authentification.

Pour garantir un niveau de sécurité suffisant, les implémentations doivent supporter au minimum (cf. [TLS]) :

- TLS 1.1
- AES 128 bits ou 256 bits
- SHA-1

Pour des clés RSA, ceci correspond aux « ciphersuites » suivants :

- `TLS_RSA_WITH_AES_128_CBC_SHA`
- `TLS_RSA_WITH_AES_256_CBC_SHA`

> ***L'authentification serveur est une fonction de sécurité faisant appel à des mécanismes cryptographiques qui peut nécessiter d'être conforme au RGS. Concernant cette conformité, se reporter à la note du paragraphe 3.2.***

3.2.5 Protection contre le rejeu

Le standard Interops est soumis aux mêmes menaces de rejeu que les standards WSS et SAML.

Les risques associés au rejeu sont :

- Le déni de service
- La connexion frauduleuse

Il est à noter que seules les applications d'un organisme client sont susceptibles de se connecter à un organisme fournisseur ce qui limite les risques de déni de service.

Les mêmes mécanismes mis en place actuellement contre les dénis de service pour protéger les applications peuvent être mis en œuvre pour protéger les éléments liés à Interops-A.

Concernant la connexion frauduleuse, en premier lieu, il est nécessaire d'empêcher le vol des assertions et des authentifiants. Dans le contexte d'Interops, les connexions entre l'organisme client et l'organisme fournisseur sont sécurisés par une authentification mutuelle sur TLS/SSL permettant une protection en confidentialité et en intégrité.

Pour empêcher le vol d'assertion ou d'authentifiants, en fonction des risques identifiés, les connexions internes de l'organisme client entre l'application et le proxy doivent être sécurisées. Le standard Interops recommande une sécurisation par TLS/SSL avec authentification serveur

des connexions internes de l'organisme client afin d'assurer l'authentification des composants et la confidentialité des communications. Cette recommandation adresse le problème des attaques de type « man in the middle » à l'intérieur de l'organisme client.

Côté organisme fournisseur, un certain nombre de contrôle du doivent être mis en œuvre :

- L'organisme fournisseur doit vérifier que le service visé est bien celui inclus dans le VI pour éviter de rejouer l'application sur une autre application du domaine.
- L'organisme fournisseur doit vérifier la validité temporelle (attributs `NotBefore`, `NotOnOrAfter`) des assertions. Cette validité doit être la plus courte possible pour minimiser la fenêtre de rejeu d'une assertion
- L'organisme fournisseur peut vérifier qu'un VI n'est pas retransmis en se basant sur son identifiant
- La signature du VI doit évidemment être vérifiée son intégrité et son authenticité. Ceci prévient de toute modification du VI et être rejoué.

Dans le but de faciliter pour le fournisseur la mise en œuvre des contrôles précédents, les organismes clients doivent s'assurer que les identifiants de VI utilisés sont le plus aléatoires possibles.

3.3 Web service sécurisé et SOAP

Le mode « application à application » du standard d'interopérabilité s'appuie sur les spécifications de WS-Security, utilisé pour sécuriser les Web Services.

WS-Security permet entre autres d'inclure des jetons de sécurité formatés en SAML 1.1 ou 2.0 tels que définis dans [R2].

3.4 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.

Le cas des structures complexes ou asynchrones ne sont pas aujourd'hui pris en charge par le standard INTEROPS-A.

De même, les problématiques liées à la transitivité - c'est-à-dire quand un organisme A qui a accès à l'application X d'un organisme B qui accède elle-même à l'application Y d'un organisme C – ne seront pas supportés dans un premier temps du fait de l'absence de cas d'usage envisagés.

Dans le cadre de structure applicative Web Service avec résultat par demande, 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.

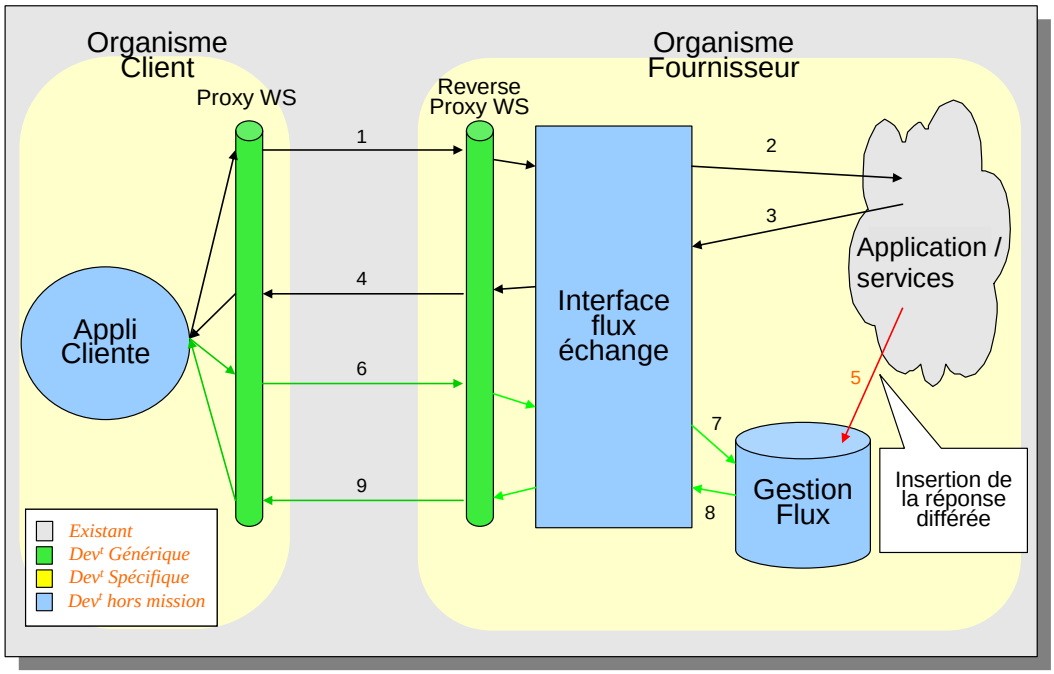


Figure 4 : Echange web service type messagerie

3.5 Eléments techniques représentant les accords

La mise en place d'échanges de données entre deux organismes fait l'objet 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.

Le standard défini par ailleurs un schéma XML pour l'échange des éléments techniques de l'accord.

La liste des paramètres techniques de l'accord et la description du schéma XML sont disponibles dans le document « Convention technique » ([R3]).

3.6 Administration

Ce document ne spécifie pas l'administration des éléments qui ne sont pas liés à l'accord d'interopérabilité tels que les serveurs applicatifs, les habilitations, etc.

L'outil d'administration des accords est décrit dans le chapitre *Lot 1 : Outils d'administration des accords*.

3.7 Interconnexion réseau, adressage et présentation de service

L'accès 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.

3.7.1 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 se fait en trois grandes zones :

- Adressage du service par le client selon le plan d'adressage interne à l'organisme client,
- Adressage du service après, si nécessaire, 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, si nécessaire, 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.

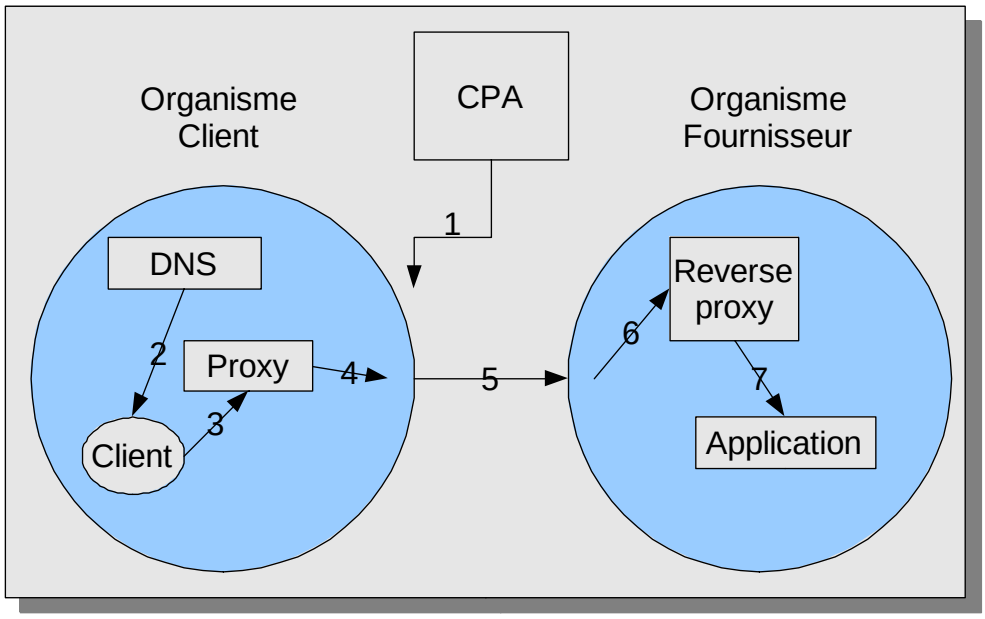


Figure 5 : Principe de communication entre services

Selon cette figure, l'adressage d'un service chez l'organisme fournisseur par un client peut suivre 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.

Ce principe permet de montrer qu'une adresse unique est suffisante au client pour atteindre le service de l'organisme fournisseur.

3.7.2 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 faciliter 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**,

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 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.cnamts.fr**, le DNS 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 services 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 distinction 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 n'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.

3.8 Traces

Les traces peuvent être de deux types :

- Les traces d'audit, qui permettent d'archiver les actions des applications clientes ou des utilisateurs finaux et de fournir une trace opposable en cas de litige ou contentieux
- Les traces techniques, relatives à chaque composant et permettant de relater les événements techniques

Le standard impose les traces d'audit afin de pouvoir effectuer des contrôles à posteriori.

Par la suite, le terme « trace » se référera aux traces d'audit.

3.8.1 Traces d'audit

Etant donné la responsabilité partagée entre l'organisme client et fournisseur, l'accord impose aux deux parties la configuration associée aux traces :

- Les événements à tracer et les éléments constituant les traces propres au standard, et donc communs à tous les accords
- Les événements à tracer et les éléments constituant les traces propres à chaque accord, jugés nécessaires, par exemple en fonction de contraintes légales particulières, ainsi que le cadre d'utilisation de ces traces.
- La durée de conservation des traces

Les événements supplémentaires à tracer, propre à chaque accord peuvent provenir :

- Des modules génériques
- Des blocs techniques propres à chaque organisme

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 de l'utilisateur ou de l'application cliente, qui est une donnée relative (sur le long terme cet identifiant peut ne plus exister ou être modifié ou affecté à un autre utilisateur ou à une autre application cliente), peut être rapproché d'autres traces d'audit du système d'information indiquant la signification de cet identifiant. De même, un vecteur d'identification contient les habilitations sous forme de PAGM d'un utilisateur ou d'une application cliente à un instant donné. D'autres traces d'audit du système peuvent être rapprochées pour déterminer l'historique des habilitations d'un utilisateur ou d'une application cliente.

La durée de conservation étant conventionnelle, elle peut varier entre les accords. Les traces propres à chaque accord doivent être séparées ou marquées de manière à gérer des cycles de vie hétérogènes entre les différents accords.

Les présentes spécifications détaillées décrivent :

- La nature des traces de journalisation propres aux modules génériques et spécifiques objet des présentes spécifications détaillées
- Les processus de consolidation des traces
- Un outil de gestion des traces pour l'analyse des traces

> *Les traces d'audit 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.*

3.8.2 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.

Les traces techniques devront notamment remonter les informations lorsqu'une anomalie survient. Par exemple, les traces techniques disponibles doivent être suffisantes pour déterminer :

- La nature et la gravité d'une anomalie

- Le composant qui a présenté l'anomalie
- Les impacts de l'anomalie (traitements en erreur, messages corrompus et / ou perdus, etc.)
- La date et l'heure de l'anomalie

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

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.

3.9 Gestion des erreurs

La gestion des erreurs liées au standard IOPS doit suivre les recommandations des différents standards sous-jacents.

Comme recommandé dans [WSI] et [WSS], en cas d'erreur lié au traitement de l'entête WS Security et donc du VI, l'organisme fournisseur doit générer une erreur HTTP avec le code « 500 Internal Server Error » ainsi qu'une erreur SOAP (SOAP fault).

D'une manière générale, le proxy doit toujours retourner une erreur sous forme SoapFault à l'entité à l'origine de la requête.

3.9.1 Utilisation de soap Fault

3.9.1.1 Composition

L'élément SOAP Fault est utilisé pour transporter les erreurs et les informations. Quand il existe, il doit apparaître, en tant qu'entrée dans le Body, une et une seule fois. Le SOAP Fault est composé des éléments suivants:

- *faultcode* : Code identifiant l'erreur
- *faultstring* : Le libellé de l'erreur
- *faultactor* : L'entité à l'origine de l'erreur
- *detail* : Information supplémentaire

3.9.1.2 Exemple

```
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
 <soapenv:Body>
    <soapenv:Fault>
       <faultcode xmlns:iops="urn:iops:faulcodes">iops:IOPSPagmInvalid</faultcode>
       <faultstring>Le ou les PAGM presents dans le VI sont invalides</faultstring>
       <faultactor>urn:iops:cnaf</faultactor>
       <detail>
          <string>Le ou les PAGM présents dans le VI ne correspondent pas à la
convention</string>
       </detail>
    </soapenv:Fault>
 </soapenv:Body>
</soapenv:Envelope>
```

3.9.1.3 Recommandations

Certaines règles doivent être respectées lors de la génération du soap fault :

- Le message Soap Fault doit obligatoirement et uniquement avoir les éléments `faultcode`, `faultstring`, `faultactor`, `detail` comme défini dans la spécification
- Le détail doit être représenté sous forme de chaine de caractères. Les extensions des messages soap fault au niveau *detailFault* sont à proscrire afin de minimiser les problèmes de portabilité
- Les éléments fils du message Soap Fault doivent être non qualifiés (sans Namespace)

L'ajout de codes personnalisés est autorisé à condition qu'il respecte la convention de nommage décrite au paragraphe 3.9.2 *Normes de référencement des erreurs*.

3.9.2 Normes de référencement des erreurs

3.9.2.1 Les Namespace utilisés

Quatre *namespaces* sont identifiés dans le cadre de la gestion des erreurs du standard INTEROPS :

- Namespace `urn:transport:faultcodes` pour les erreurs en rapport avec la couche de transport et négociation TLS. On utilisera dans ce document le préfix `transport`.
- Namespace du WSS pour les erreurs en rapport avec la gestion de l'assertion. On utilisera dans ce document le préfix `wsse`.
- Namespace `urn:iops:vi:faultcodes` pour les erreurs en rapport avec le VI. On utilisera dans ce document le préfix `vi`.
- Namespace `urn:iops:service:faultcodes` pour les erreurs en rapport avec le service fournisseur visé. On utilisera dans ce document le préfix `service`.

3.9.2.2 Les acteurs identifiés

Nous utiliserons comme identifiant *faultActor* pour décrire l'organisme à l'origine de l'erreur. Cet identifiant doit respecter la convention de nommage suivante : `urn:iops:<organisme>` (Exemple : urn:iops:cnaf).

Cela permet d'identifier rapidement l'entité à l'origine de l'erreur.

3.9.2.3 Cartographie des erreurs INTEROPS

➢ **Erreur au niveau assertion**

La spécification WSS (cf. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf) définit un certain nombre de raisons pour lequel il est utile de remonter une exception. Ces codes erreur sont à réutiliser dans le cadre d'INTEROPS :

FaultString	FaultCode
Le jeton présent dans l'assertion n'est pas supporté	wsse:UnsupportedSecurityToken
L'algorithme de signature ou de chiffrement utilisé n'est pas supporté	wsse:UnsupportedAlgorithm

Une erreur s'est produite lors du traitement de l'entête de sécurité	wsse:InvalidSecurity
Le jeton de sécurité fourni est invalide	wsse:InvalidSecurityToken
Le jeton de sécurité n'a pas pu être authentifié ou autorisé	wsse:FailedAuthentication
La signature ou le chiffrement n'est pas valide	wsse:FailedCheck
La référence au jeton de sécurité est introuvable	wsse:SecurityTokenUnavailable

➢ **Erreur au niveau du service fournisseur**

Le tableau suivant décrit les erreurs identifiées dans le cadre de la disponibilité du service fournisseur :

FaultString	FaultCode
Le service est injoignable	soap:Server
Le service ne peut être rendu	soap:Server

Dans le cas où le système de trace est défaillant pour l'un des composants de son architecture (par exemple la brique de vérification du VI), le service ne peut être rendu. Une erreur « soap:Server » doit alors être transmise.

Il en est de même pour tout composant sous-jacent à l'application (base de données, annuaire, etc.) empêchant de rendre le service.

➢ **Erreur lié au VI**

Le tableau suivant décrit les erreurs identifiées dans le cadre de la gestion du VI :

FaultString	FaultCode
Le ou les PAGM présents dans le VI sont invalides	vi:InvalidPagm
Le service visé par le VI n'existe pas ou est invalide	vi:InvalidService
L'identifiant de l'organisme client présent dans le VI est invalide ou inconnu	vi:InvalidIssuer
Le niveau d'authentification initial n'est pas conforme au contrat d'interopérabilité	vi:InvalidAuthLevel
Le VI est invalide	vi:InvalidVI

➢ **Erreur au niveau transport**

Le tableau suivant décrit les erreurs identifiées dans le cadre de la couche transport :

FaultString	FaultCode
Problème de connexion	transport:TimeOutConnexion
Problème lors de l'initialisation de la négociation TLS	transport:InvalidTLSNegociation

3.9.3 Filtrage des erreurs

La diffusion et l'exploitation de ces erreurs techniques doivent être limitées aux seuls éléments d'infrastructure du standard INTEROPS (proxy, reverse proxy, serveur de jeton). A ce titre un filtrage devra être effectué au niveau proxy et reverse proxy avant restitution de la réponse SOAP à l'entité à l'origine de la requête afin de :

- Convertir les erreurs techniques en erreurs génériques
- Masquer des détails pouvant donner des informations de nature à mettre à mal la sécurité du système d'information

3.9.4 Lien avec les traces

La gestion des erreurs doit être mise en relation avec le mécanisme de trace afin de pouvoir remonter les tentatives d'intrusion à travers le processus de supervision de chaque organisme partenaire.

Cette disposition contribue à assurer la sécurité aval de la sphère de confiance INTEROPS.

Il est également nécessaire d'enrichir les traces d'audit avec le code erreur (FaultCode) afin de pouvoir éventuellement faire des rapprochements sur des critères de transactions infructueuses.

3.10 Synchronisation temporelle

Pour faciliter le rapprochement des traces et limiter la déviation des horloges pour les périodes de validité des VI, les serveurs des organismes doivent se synchroniser sur un serveur NTP reconnu. Chacun communiquera alors le serveur NTP de référence choisi. S'il s'agit d'un serveur NTP interne, l'organisme devra préciser quelle méthode est utilisée pour synchroniser ce serveur (GPS, DCF77, etc.).

Dans certains cas, un serveur NTP pourra être mis à disposition par l'un ou l'autre des deux organismes.

Si la synchronisation temporelle des serveurs est obligatoire, le choix du serveur de temps est conventionnel.

4. LOTS A DEVELOPPER

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'utilisation des vecteurs d'identification et le traitement des requêtes sortantes
- Vecteurs et reverse-proxy organismes fournisseurs, qui concerne la vérification et la consommation des vecteurs d'identification et le traitement des 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.

4.1 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 création des accords, il permet de récapituler dans un format d'échange normalisé les besoins 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 organismes client et organismes fournisseur.

4.2 Lot 2 : Vecteur et proxy organisme client

Le lot 2 décrit les scénarii associés à la création des vecteurs d'identification signés et la transmission des requêtes sortantes :

- Intégration du VI au niveau de l'application cliente
- Intégration du VI au niveau du proxy

4.3 Lot 3 : Vecteur et reverse proxy organisme fournisseur

Le lot 3 décrit les scénarios associés à la vérification et la consommation des vecteurs d'identification, ainsi que le traitement de chaque requête entrante.

4.4 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 d'enregistrement des traces : il permet d'insérer des traces dans une base
- Outil de gestion de traces : il permet l'analyse des traces et le contrôle *a posteriori*.

5. 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.

5.1 Outil de création des accords

5.1.1 Rôle de l'outil

Cet outil a pour objet de créer et modifier des conventions techniques d'interopérabilité entre les organismes client et fournisseur.

Il propose une interface permettant à chaque organisme de déclarer les éléments conventionnels le concernant.

Il produit un fichier au format XML contenant les éléments de paramétrage de l'interopérabilité souhaités par les organismes conforme au schéma défini dans [R3] Convention technique Interops.

Cet outil doit permettre de signer ce document.

Cet outil doit pouvoir valider le document XML, du point de vue de la syntaxe XML et de la conformité au schéma.

5.1.2 Cinématique générique

Pour créer une convention technique, la cinématique générique est la suivante :

- Création d'un nouveau projet dans l'outil
- Remplissage des champs des formulaires de l'outil par un premier organisme à partir des informations en sa possession
- Exportation depuis l'outil de la convention partiellement remplie au format XML spécifié par le standard Interops
- Envoi (mail, etc.) de ce fichier au second organisme
- Importation dans l'outil du fichier de convention XML Interops par le second organisme
- Remplissage des champs des formulaires de l'outil par le second organisme à partir des informations en sa possession
- Exportation depuis l'outil de la convention complétée au format XML Interops
- Exportation éventuelle de la convention dans d'autres formats (HTML, etc.)

5.1.3 Interface d'entrée

5.1.3.1 Éléments constituants une convention technique Interops

Cet outil prend en entrée les informations constituant une convention technique Interops (cf. [R3] Convention technique Interops).

Ces informations peuvent éventuellement être sous la forme de fichiers techniques (certificats).

Ils sont fournis à l'outil par le biais d'une IHM (Interface Homme Machine).

5.1.3.2 Convention technique XML

L'outil peut également prendre en entrée un fichier de convention technique Interops au format XML complet ou partiel.

5.1.4 Interface de sortie

5.1.4.1 Convention technique XML

L'outil permet d'exporter un fichier de convention technique Interops au format XML. Ce fichier a vocation à être échangé et complété par les deux organismes. Il respecte le schéma des conventions Interops sauf dans le cas où le document est incomplet (les éléments obligatoires peuvent ne pas être renseignés par exemple).

La convention au format XML est le fichier des éléments techniques des accords et, à ce titre, est annexée à la convention passée entre les deux organismes établissant les modalités d'interopérabilité.

Ce fichier peut être signé en utilisant le ou les bi-clés / certificats fournis du ou des auteurs. Les moyens utilisés pour générer les bi-clés ou distribuer les certificats de signature et de leurs chaînes de confiance sont hors-scope du standard.

5.1.4.2 Convention technique « lisible »

L'outil doit permettre de générer un fichier de convention technique Interops dans un format « lisible » par un utilisateur (HTML, PDF, etc.).

Cette version lisible des éléments techniques des accords peut également être annexée à la convention passée entre les deux organismes établissant les modalités d'interopérabilité.

5.2 Outil de mise en œuvre des accords

5.2.1 Rôle de l'outil

Cet outil a pour objet de générer les éléments de configuration des différentes briques techniques du système à partir du fichier XML de convention technique Interops.

Il peut également servir à déployer ces éléments dans chacun des deux systèmes d'information client et fournisseur.

5.2.2 Interface d'entrée

5.2.2.1 Convention technique XML

Cet outil prend en entrée un fichier de convention technique Interops au format d'échange XML.

Il s'agit ici d'un fichier « complet » conforme au schéma de donnée.

5.2.3 Interface de sortie

L'outil doit permettre de générer, à partir de la convention technique XML, les éléments de configuration des différents modules définis au paragraphe 3.1.5 « Description des éléments d'architecture » et en particulier :

- Module de génération du VI
- Module de génération d'assertion
- Module de vérification d'assertion
- Proxy
- Reverse-proxy
- Base des traces

> ✍ *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.*

6. LOT 2 : VECTEUR ET PROXY ORGANISME CLIENT

Le déploiement, côté organisme client, des éléments relatifs au vecteur d'identification est composé de deux modules :

- Module de génération du VI
- Module de génération d'assertion

Ils sont appelés par l'application cliente (cf. paragraphe 6.1 *Intégration du VI par l'application cliente*) ou le proxy (cf. paragraphe 6.2 *Intégration du VI par le proxy*) afin de générer le VI à incorporer dans la requête. Quelque soit l'approche choisie, les requêtes SOAP qui sortiront de l'organisme client posséderont un VI conforme au standard d'interopérabilité.

Dans tous les cas, le proxy authentifie l'application cliente et redirige toutes les requêtes vers l'organisme fournisseur approprié.

6.1 Intégration du VI par l'application cliente

6.1.1 Description du scénario

Dans ce scénario, l'**application cliente** est responsable de la constitution de certains éléments du VI et de l'intégration du VI dans **chaque requête SOAP soumise**.

Le module proxy conserve un rôle de sécurisation et de trace des flux.

Dans ce chapitre, nous ne décrirons que les échanges **internes à l'organisme client**.

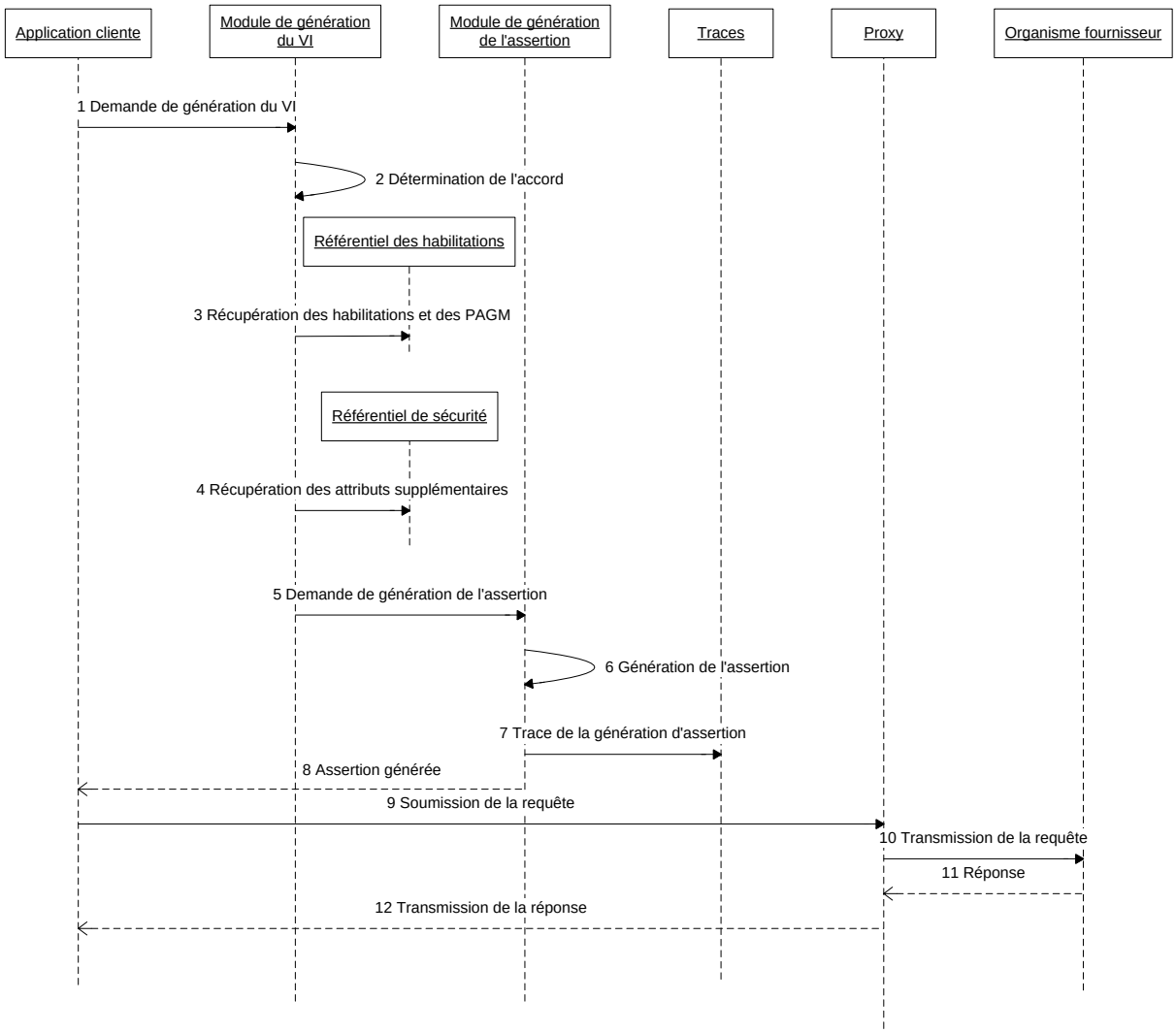
6.1.2 Composants utilisés

Les composants mis en œuvre dans ce scénario sont les suivants :

- Module d'authentification
- Module de génération du VI
- Module de génération de l'assertion
- Bases des traces
- Proxy

6.1.3 Diagramme de séquence nominal

Le diagramme de séquence entre les objets identifiés précédemment est le suivant :



1. L'application cliente fait la demande de génération du VI. Elle passe en paramètre les informations utiles à la génération du VI :
   - L'identifiant local de l'application cliente (ou de l'utilisateur)
   - L'identifiant du service visé
   - La méthode d'authentification utilisée par l'application pour obtenir le VI (login/mot de passe, certificat, etc.)
2. Le module de génération du VI détermine à partir du service visé l'accord et les informations techniques associées :
   - L'identifiant de l'accord et la version en cours
   - Le format de l'identifiant
   - Le niveau d'authentification acceptable
   - Les attributs supplémentaires nécessaires
   - L'identifiant de l'émetteur
   - La date d'authentification
3. Le module de génération du VI vérifie à partir du référentiel de sécurité les habilitations de l'application. Le ou les PAGM sont simplement récupérés ou déduits des habilitations de l'application.
4. Dans le cas où l'accord précise que des attributs supplémentaires doivent être contenus dans le VI, ils sont récupérés à partir du référentiel de sécurité.

5. Le module de génération du VI demande alors au module de génération d'assertion de générer une réponse SAML
6. Le module de génération d'assertion produit l'assertion SAML à partir des éléments fournis par le module de génération du VI et en générant à la volée :
   - Identifiant unique du VI
   - Date d'émission du VI
   - Date de validité du VI
   - Signature
7. Le module de génération de l'assertion trace l'événement.
8. Le VI ainsi généré est retourné à l'application cliente.
9. L'application cliente soumet alors au proxy une requête à destination du service visé intégrant dans l'entête WS-Security le VI.

   L'application cliente peut s'authentifier sur le proxy ou le proxy peut utiliser le VI déjà intégré dans la requête pour générée les traces associées au sujet du VI.
10. Après avoir réalisé une authentification mutuelle avec l'organisme fournisseur à l'aide de TLS [TLS], le proxy transmet la requête à l'organisme fournisseur dans le canal chiffré.

    Le proxy peut également tracer la requête de l'application pour déterminer les accès aux ressources externes.
11. Le service visé génère la réponse et la transmet au proxy dans le canal TLS sécurisé
12. Le proxy transmet la réponse à l'application cliente. La réponse peut éventuellement être tracée en fonction de l'accord passé avec l'organisme fournisseur.

6.2 Intégration du VI par le proxy

6.2.1 Description du scénario

Dans ce scénario, le **proxy** est responsable de la constitution de certains éléments du VI et de l'intégration du VI dans chaque requête SOAP transmise à l'organisme fournisseur. L'application cliente doit alors s'authentifier sur le proxy afin de générer un VI propre à l'application. Un format pivot peut être utilisé entre l'application cliente et le proxy afin de fournir des éléments supplémentaires pour la génération du VI et notamment spécifier l'utilisateur pour lequel l'appel est fait, si besoin est.

Le module proxy conserve un rôle de sécurisation et de trace des flux.

Dans ce chapitre, nous ne décrirons que les échanges **internes à l'organisme client**.

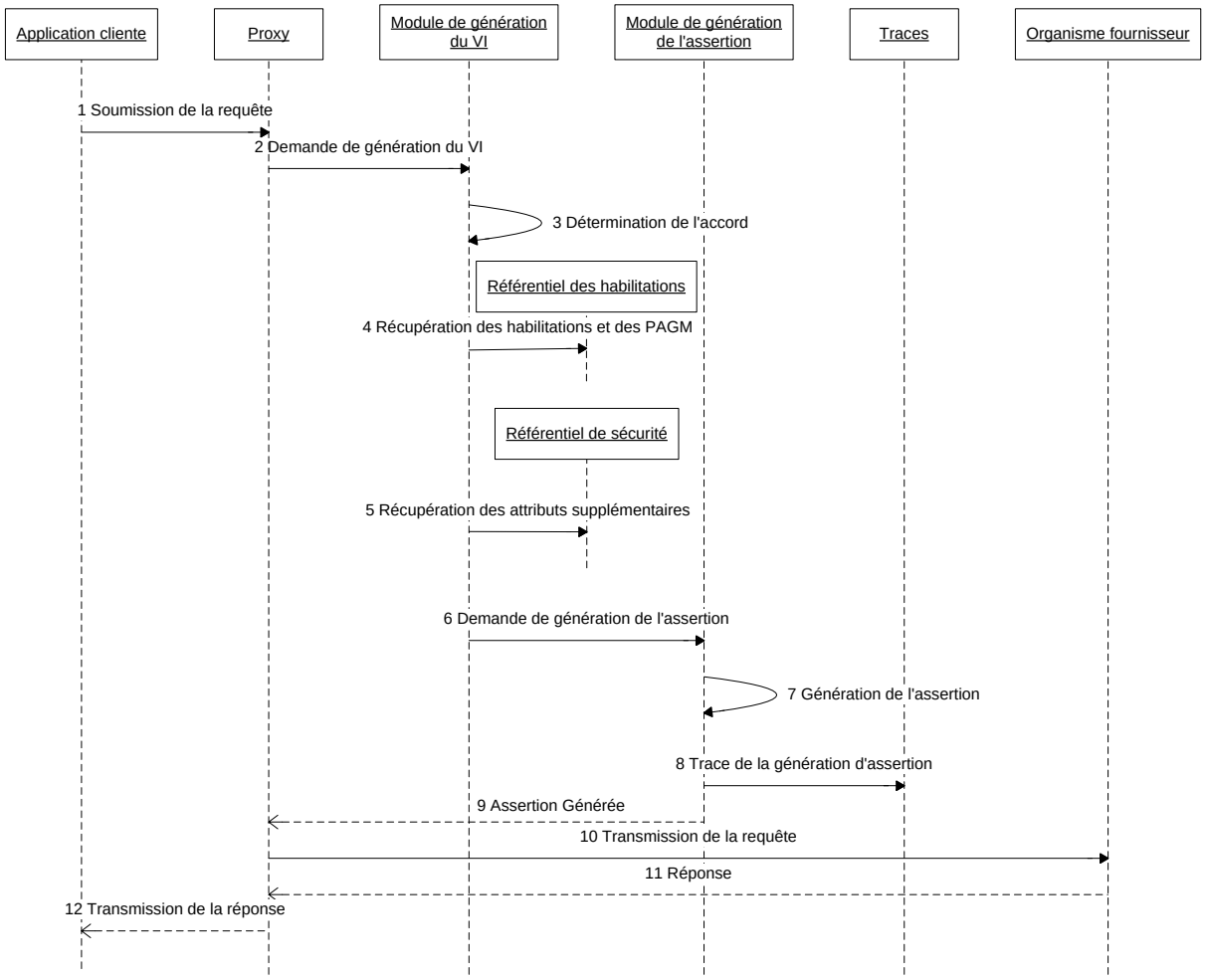
6.2.2 Composants utilisés

Les composants mis en œuvre dans ce scénario sont les suivants :

- Module d'authentification
- Module de génération du VI
- Module de génération de l'assertion
- Bases des traces
- Proxy

6.2.3 Diagramme de séquence nominal

Le diagramme de séquence entre les objets identifiés précédemment est le suivant :



1. L'application cliente génère et soumet au proxy une requête à destination du service visé.

   L'application cliente peut s'authentifier sur le proxy et/ou utiliser un format pivot propriétaire pour tracer l'activité de l'application cliente (ou de l'utilisateur appelant) ou transmettre des informations pertinentes pour la génération du VI à intégrer dans la requête.

2. Le proxy fait la demande de génération du VI. Il passe en paramètre les informations utiles à la génération du VI :
   - L'identifiant local de l'application cliente (ou de l'utilisateur)
   - L'identifiant du service visé
   - La méthode d'authentification utilisée par l'application pour obtenir le VI (login/mot de passe, certificat, etc.)
3. Le module de génération du VI détermine à partir du service visé l'accord et les informations techniques associées :
   - L'identifiant de l'accord et la version en cours
   - Le format de l'identifiant
   - Le niveau d'authentification acceptable
   - Les attributs supplémentaires nécessaires
   - L'identifiant de l'émetteur

- La date d'authentification

4. Le module de génération du VI vérifie à partir du référentiel de sécurité les habilitations de l'application. Le ou les PAGM sont simplement récupérés ou déduits des habilitations de l'application.
5. Dans le cas où l'accord précise que des attributs supplémentaires doivent être contenus dans le VI, ils sont récupérés à partir du référentiel de sécurité.
6. Le module de génération du VI demande alors au module de génération d'assertion de générer une réponse SAML
7. Le module de génération d'assertion produit l'assertion SAML à partir des éléments fournis par le module de génération du VI et en générant à la volée :
   - Identifiant unique du VI
   - Date d'émission du VI
   - Date de validité du VI
   - Signature
8. Le module de génération de l'assertion trace l'événement.

   En cas d'échec de génération de l'assertion, une réponse avec statut d'erreur doit être générée (cf. paragraphe 3.9 *Gestion des erreurs*) et retournée à l'application cliente
9. Le VI ainsi généré est retourné au proxy.
10. Le proxy ajoute l'entête WS-Security et le VI à la requête émise par l'application cliente et, après avoir réalisé une authentification mutuelle avec l'organisme fournisseur à l'aide de TLS [TLS], transmet la requête à l'organisme fournisseur dans le canal chiffré.

Le proxy peut également tracer la requête de l'application pour déterminer les accès aux ressources externes.

11. Le service visé génère la réponse et la transmet au proxy dans le canal TLS sécurisé
12. Le proxy transmet la réponse à l'application cliente. La réponse peut éventuellement être tracée en fonction de l'accord passé avec l'organisme fournisseur.

7. LOT 3 : VECTEUR ET REVERSE-PROXY ORGANISME FOURNISSEUR

Le déploiement, côté organisme fournisseur, des éléments relatifs au vecteur d'identification est composé de trois modules :

- Gestionnaire de contexte
- Module de vérification
- Module de consommation

Le module reverse-proxy authentifie le flux entrant et redirige toutes les requêtes vers les services visés.

7.1 Validation du VI et réponse du service visé

Ce scénario est joué pour chaque requête soumise à partir d'un organisme client partenaire au service visé (cf. les scénarios de génération des requêtes décrits aux paragraphes 6.1 *Intégration du VI par l'application cliente* et 6.2 *Intégration du VI par le proxy*).

La validation du VI entraîne la création d'un contexte de sécurité récupéré par le service visé afin d'élaborer une réponse appropriée.

Dans ce chapitre, nous ne décrirons que les échanges **internes à l'organisme fournisseur**.

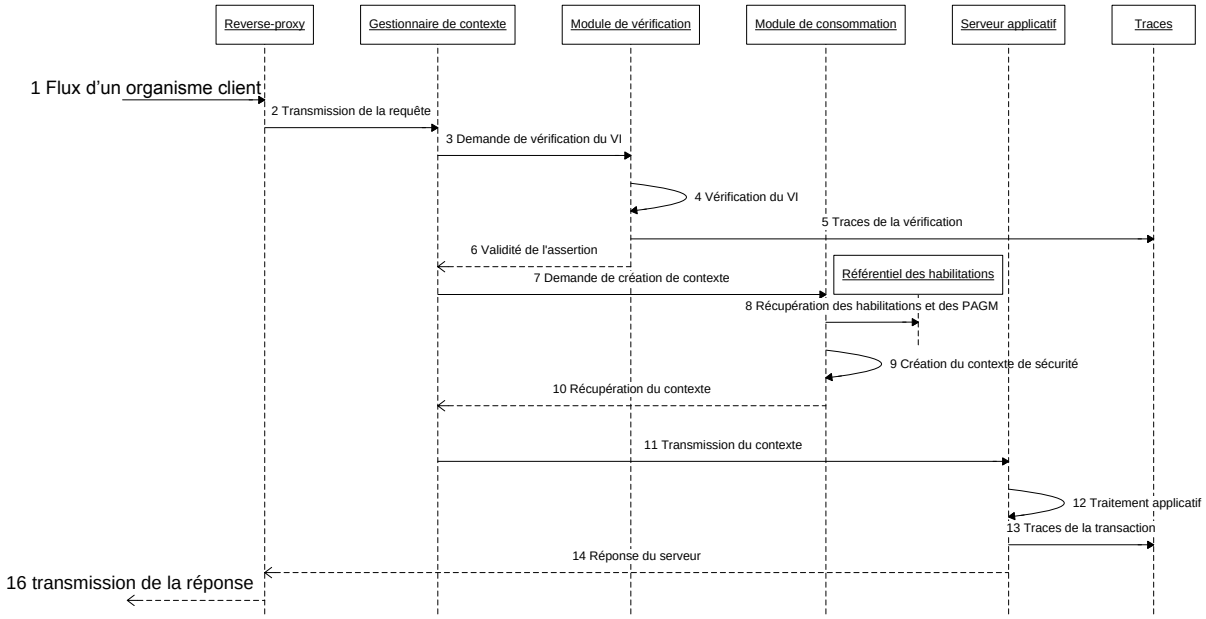
7.1.1 Composants utilisés

Les composants mis en œuvre dans ce scénario sont les suivants :

- Gestionnaire de contexte
- Module de vérification
- Module de consommation
- Bases des traces
- Reverse-proxy
- Serveur applicatif

7.1.2 Diagramme de séquence nominal

Le diagramme de séquence entre les objets identifiés précédemment est le suivant :



1. Le flux généré par l'organisme client est transmis au reverse-proxy de l'organisme fournisseur. Il est alors authentifié et déchiffré par le reverse-proxy grâce au protocole TLS.

   Le reverse-proxy vérifie que l'organisme client est habilité à accéder à l'application, conformément aux accords signés par les organismes.

   En cas d'échec de négociation, une réponse avec statut d'erreur doit être générée (cf. paragraphe 3.9 *Gestion des erreurs*) et retournée à l'organisme client
2. La requête est transmise au gestionnaire de contexte, qui récupère le VI contenu dans la requête SOAP.
3. Le gestionnaire de contexte vérifie le VI en transmettant au module de vérification :
   - L'assertion SAML
   - Le service visé
   - L'identifiant de l'organisme client (déterminé à partir du certificat d'authentification)
4. Le module de vérification détermine l'accord correspondant à l'échange et vérifie la validité du VI, c'est-à-dire vérifie :
   - Le format de l'assertion (champs obligatoires et format des champs)
   - L'émetteur et de l'assertion (par rapport à l'organisme client)
   - La durée de validité de l'assertion
   - Le service visé et les restrictions de l'assertion
   - Le niveau d'authentification
   - La signature de la réponse
   - Etc.
5. L'assertion SAML et le résultat de la vérification sont tracés.
6. Le résultat de la vérification du VI est retourné au gestionnaire de contexte. Si l'assertion SAML est valide, le traitement peut continuer.

   En cas d'échec de vérification de l'assertion, une réponse avec statut d'erreur doit être générée (cf. paragraphe 3.9 *Gestion des erreurs*) et retournée à l'organisme client.
7. Le gestionnaire de contexte demande la création d'un contexte de sécurité en transmettant au module de consommation :

- L'assertion SAML vérifiée
- Le service visé
- L'identifiant de l'organisme client

8. A partir des informations contenues dans le VI et notamment les PAGM, le module de consommation détermine le profil avec les habilitations associées. S'il n'y a pas d'habilitation, une réponse avec statut d'erreur doit être générée (cf. paragraphe 3.9 *Gestion des erreurs*) et retournée à l'organisme client.
9. Le module de consommation génère un contexte de sécurité avec les informations d'identification de l'application cliente (ou de l'utilisateur) dans l'espace de confiance de l'organisme fournisseur.
10. Le module de consommation transmet le contexte de sécurité au gestionnaire de contexte. Le contexte de sécurité ainsi créé n'est valable que pour une requête et sa réponse. Aucun élément de contexte n'est donc transmis à l'organisme client et la gestion du contexte reste un processus interne à l'organisme fournisseur.
11. Le gestionnaire de contexte transmet le contexte de la requête au serveur applicatif.
12. Le service visé réalise le traitement applicatif en fonction des éléments de la requête et du contexte de sécurité.
13. Le service conserve une trace liée à la transaction effectuée suite à la requête, conformément à l'accord passé avec l'organisme client.
14. La réponse est générée et transmise au reverse-proxy.
15. Le reverse-proxy retourne la réponse du service dans le canal TLS sécurisé créé initialement.

8. LOT 4 : TRACES

8.1 Présentation générale

Dans ce chapitre, ne sont présentés que les éléments des traces relatifs au standard InterOPS. Les besoins de traces applicatives sont à étudier au niveau du projet fonctionnel, le format et le contenu des traces applicatives reste ouvert.

Les paragraphes 8.1.1 *Eléments de traçage côté organisme client* et 8.1.2 *Eléments de traçage côté organisme fournisseur* présentent les événements et les éléments constituant les traces.

8.1.1 Eléments de traçage côté organisme client

L'organisme client doit tracer dans le cadre de la fourniture de la solution :

- L'authentification de l'application cliente
- La génération d'un VI pour l'application cliente ou pour le compte d'un utilisateur

La trace d'une authentification de l'application doit comporter les éléments suivants :

- Date de l'événement
- Identifiant local à l'organisme client de l'application
- Méthode d'authentification
- Statut de l'authentification (succès et échec)

La trace de génération du VI doit comporter les éléments suivants :

- Date de l'événement
- Identifiant local à l'organisme client de l'application cliente ou de l'utilisateur
- Identifiant du service visé
- Identifiant « impersonnifié » de l'application cliente ou de l'utilisateur, contenu dans le sujet de l'assertion
- Identifiant du VI
- VI transmis, contenant la signature

Optionnellement, l'organisme pourra indiquer les conditions de génération du VI dans les traces : à partir de quel poste ou quel type de poste, de quelle entité de l'organisme, etc.

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 versions de l'accord
- Les mises à jour des définitions de services selon l'accord d'interopérabilité (URI des services, listes de PAGM associés ainsi que niveaux d'authentification 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)

- Eventuellement, le traçage du système local concernant l'administration des utilisateurs 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)

8.1.2 Eléments de traçage côté organisme fournisseur

L'organisme fournisseur doit tracer dans le cadre de la fourniture de la solution :

- La réception et la vérification du VI
- La transaction effectuée par l'application

La trace de réception et vérification du VI doit comporter les éléments suivants :

- Date de l'événement
- Identifiant « impersonnifié » de l'application cliente ou de l'utilisateur, contenu dans le sujet de l'assertion
- Identifiant du service visé
- Identifiant local à l'organisme fournisseur de l'application cliente ou de l'utilisateur
- Identifiant du VI
- VI reçu, contenant la signature
- Statut de la vérification (succès et échec)

La trace d'une transaction doit comporter les éléments suivants :

- Date de l'événement
- Identifiant local à l'organisme fournisseur de l'application cliente ou de l'utilisateur
- URL visée (suivant définition du paragraphe 3.7.2 *Dénomination de service*)
- Action réalisée
- Statut de l'action (succès et échec)

L'Identifiant local à l'organisme fournisseur peut être la représentation de l'application cliente ou de l'utilisateur dans l'espace de confiance de l'organisme fournisseur. Préférentiellement, l'Identifiant local à l'organisme fournisseur sera égal à l'identifiant « impersonnifié » de l'application cliente ou de l'utilisateur.

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).
- Les mises à jour des versions de l'accord

8.1.3 Sécurisation des traces

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

Etant donné les impacts induits par une signature de chaque élément de trace (ex : l'accès à une URL), en fonction des exigences, une protection physique et logicielle d'accès aux traces peut être suffisante.

Dans tous les cas, l'accès aux traces, même en lecture, devra être conservé à des fins d'audit.

La sécurité des traces sera conventionnelle et devra prendre en compte les contraintes opérationnelles de la chaîne complète :

- Gestion des traces sur les différents composants
- Performance des composants
- Etc.

8.1.4 Processus de consolidation

L'intégralité des traces concernant un service ne peut être obtenue que par la consolidation des traces des organismes client et fournisseur. En effet, l'authentification de l'utilisateur est réalisée par l'organisme client, alors que la transaction est effectuée chez l'organisme fournisseur.

Sans consolidation, chacun des organismes a donc une vue partielle des opérations :

- Un organisme client peut déterminer à quel service accède une application cliente ou un utilisateur
- Un organisme fournisseur peut déterminer quels organismes clients accèdent à ses applications sans connaître le nom l'utilisateur ou l'application cliente, ni l'usage qui a été fait du service rendu

La consolidation des traces d'un organisme client et d'un organisme fournisseur peut être à l'initiative d'un organisme client ou fournisseur et reste un événement exceptionnel.

Dans le respect de l'accord, la consolidation consiste en l'échange des traces d'un organisme liées à une assertion (en utilisant l'identifiant du VI et l'identité de l'organisme demandeur pour éviter le risque d'adresser des traces à un autre organisme).

Si un organisme fournisseur désire déclarer un comportement suspect, il fournit à l'organisme client une liste d'identifiants d'assertion à l'origine du comportement. L'organisme client peut alors déterminer le ou les identifiants des utilisateurs locaux ainsi que les conditions d'authentification.

Le processus de consolidation doit être facilité par l'outil de gestion des traces (cf. paragraphe 8.3 *L'outil de gestion des traces*), pour générer les demandes de consolidation ou y répondre.

8.2 Le module d'enregistrement des traces

Le module d'enregistrement des traces doit permettre aux différents modules de réaliser des traces d'audit. Chaque module doit cependant archiver différents événements avec un format également différent, rendant difficile la mutualisation de ce module au niveau des autres composants.

Les traces peuvent cependant être centralisées pour conservation et consultation ultérieure. Ce processus périodique peut alors collecter les différentes traces et les formater pour une exploitation postérieure. Sans contrainte de performance particulière, le module peut :

- Protéger l'intégrité et la cohérence des traces (par une signature)
- Formater les traces et les stocker dans une base commune
- Indexer les traces suivant les différents critères de recherche

8.3 L'outil de gestion des traces

L'outil d'analyse de traces doit être développé pour permettre l'exploitation des traces notamment lors d'un audit approfondi.

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

Il a les fonctions suivantes :

- Consulter les traces
- Rechercher dans les traces en fonction de critères temporels 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 l'intégrité de tout ou partie des traces
- Initier une demande de consolidation de traces en générant une liste d'identifiants d'assertion à partir de certains critères
- Réaliser la consolidation des traces à partir de la liste d'identifiants d'assertion

Les résultats des différentes opérations pourront être rendus selon différents modes de sortie (texte, HTML, PDF) et selon différents médiums de sortie (serveur HTTP, fenêtre graphique, console, fichier, imprimante).

9. ANNEXES

9.1 Acronymes

Sigles - abréviations	Définition
AAS	Authentification-Autorisation-SSO
ADAE	Agence pour le développement de l'administration électronique
CNIL	Commission Nationale de l'Informatique et des Libertés
CPA	Collaboration Protocol Agrement
CPP	Collaboration Protocol Profile
HTTP	Hypertext Transfer Protocol
LDAP	Lightweight Directory Access Protocol
MSP	Mon Service Public
PDA	Personal Digital Assistant
RGS	Référentiel Général de Sécurité
SAML	Security Assertion Markup Language
SI	Système d'Information
SOAP	Simple Object Access Protocol
SSO	Single Sign-On (équivalent français : authentification unique)
URI	Universal Ressource Information
URL	Universal Ressource Location
VI	Vecteur d'Identification
WAP	Wireless Application Protocol
X.509	Norme relative aux certificats à clé publique
XML	eXtented Markup Language

9.2 Glossaire

Le glossaire est contenu dans le document [R4]