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 fonctionnelles et techniques pour la sphère de confiance Interops

Standard d'interopérabilité entre organismes de la sphère sociale

Réf. : Standard Interops-S2.0_SpécificationsDétaillées
Version 2.0 du 05/04/2012

Référence : **Version :** **Date de dernière mise à jour :**	Standard Interops-S2.0_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

1. INTRODUCTION

1.1. Objet du document

Ce document présente les spécifications fonctionnelles et techniques pour la gestion d'une **sphère de confiance** par le Standard d'Interopérabilité des Organismes de la Sphère Sociale [R1] dans un **mode « portail à portail »** : **Interops-S**.

Interops 2.0 spécifie le mode Interops-P « portail à portail » portant sur l'accès à un service par un utilisateur de manière sécurisée entre deux partenaires :

- Un organisme client chargé de réaliser l'authentification des utilisateurs
- Un organisme fournisseur hébergeant un service

Tous les échanges sont à l'initiative de l'organisme client.

Interops-S étend ces mécanismes pour :

- Mettre en relation plus de deux organismes
- Initier indifféremment des cinématiques depuis chaque organisme
- Déconnecter globalement un utilisateur

1.2. Relation avec d'autres documents

Ce document dérive et complète le standard [R1] et [R3].

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 « **Description des mécanismes** » présente les différents mécanismes mis en œuvre dans le cadre d'Interop-S ainsi que les cinématiques d'authentification et de déconnexion ;
- Le chapitre 3 « **Spécifications techniques** » précise pour l'architecture, les éléments de sécurité ainsi que le format des différents échanges à mettre en place dans le cadre d'Interops-S ;

  Le chapitre 4 « **Impacts sur les traces** » énumère les différents éléments à consolider lors de la génération des traces.

88 ## 1.4. Références

89 ### 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 Interops-P2.0_SpécificationsDétaillées	Spécifications détaillées du mode « portail à portail »	Groupe de travail Interops	2.0	05/04/2012
[R4]	Standard Interops2.0_FormatEchangeTraces	Format d'échange des traces	Groupe de travail Interops	2.0	05/04/2012
[R5]	Standard Interops2.0_ConventionTechnique	Convention technique	Groupe de travail Interops	2.0	05/04/2012
[R6]	Standard Interops2.0_GuideMiseEnOeuvre-TransfertDeContexteApplicatif	Guide de mise en œuvre pour le transfert de contexte applicatif	Groupe de travail Interops	1.0	05/04/2012
[R7]	Standard Interops2.0_Glossaire	Glossaire du standard Interops	Groupe de travail Interops	2.0	05/04/2012

90 ### 1.4.2. Documents externes

Réf	Titre	Auteur	Date
[HTTP1.0]	RFC 1945 - Hypertext Transfer Protocol -- HTTP/1.0	T. Berners-Lee, R. Fielding, H. Frystyk	Mai 1996
[HTTP1.1]	RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1	R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee	Juin 1999
[RFC1951]	RFC1951 - DEFLATE Compressed Data Format Specification version 1.3	P. Deutsch	Mai 1996
[RFC4122]	A Universally Unique IDentifier (UUID) URN Namespace	P. Leach, M. Mealling, R. Salz	07/2005
[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

[SAMLAuthnCxt]	Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0	J. Kemp et al.	15/03/2005
[SAMLBind]	Bindings for the OASIS Security Assertion Markup Language	S. Cantor et al.	15/03/2005
[SAMLCore2]	Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0	Cantor, Scott, Kemp, John, Philpott, Rob, Maler, Eve, eds.	15/03/2005
[SAMLDisco]	Identity Provider Discovery Service Protocol and Profile	H. Lockhart, B. Campbell, R. Widdowson, S. Cantor	27/03/2008
[SAMLProf]	Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0	S. Cantor et al.	15/03/2005
[TLS]	RFC 4346 - The Transport Layer Security (TLS) Protocol Version 1.1	T. Dierks, E. Rescorla	Avril 2006
[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. DESCRIPTION DES MECANISMES

Une sphère de confiance correspond à une relation entre n organismes et non uniquement bipartite entre un organisme clients et un organisme fournisseur, tel que défini dans Interops [R1] et [R3].

Interops-S permet de mettre en relation n organismes au sein d'une sphère de confiance. L'objectif est de pouvoir assurer une navigation à un utilisateur sans réauthentification au travers de son navigateur entre les différents opérateurs de service dès lors qu'il s'est authentifié auprès d'un opérateur d'authentification (mécanismes de Web SSO entre un opérateur d'authentification et des opérateurs de service).

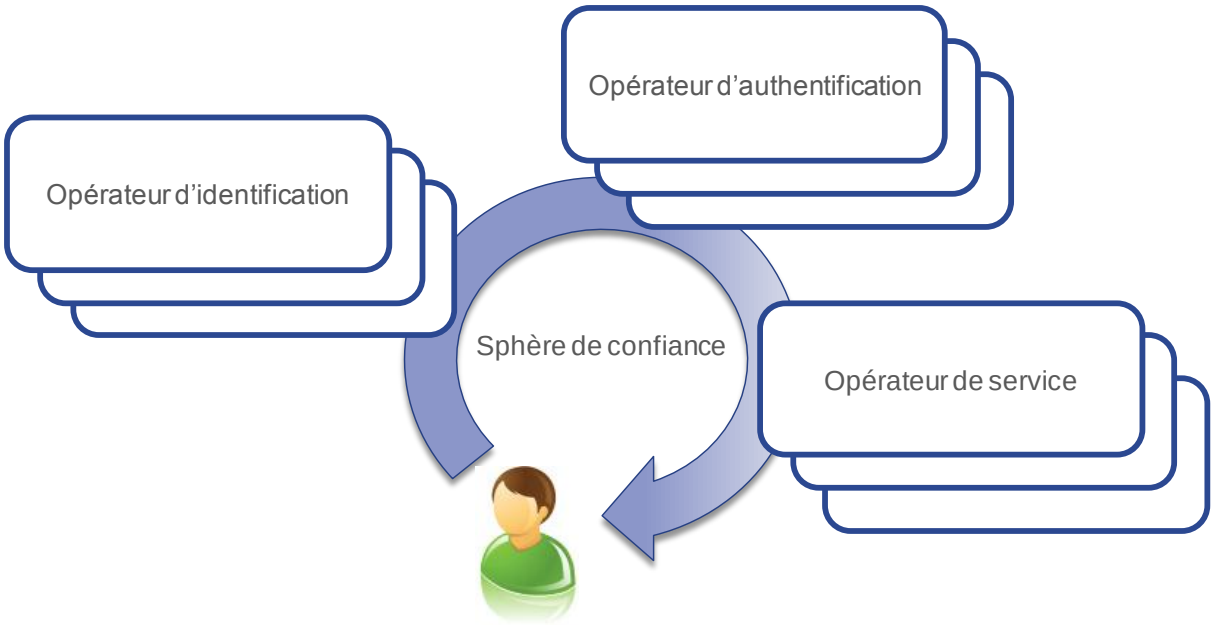
Interops-S définit les rôles d'organismes suivant :

- **Opérateur d'identification** : organisme chargé de réaliser l'identification de l'utilisateur.
- **Opérateur d'authentification** : organisme authentifiant l'utilisateur final. L'identifiant de l'utilisateur peut avoir été récupéré grâce à l'opérateur d'identification. Il est chargé de produire un VI.

  C'est l'équivalent de l'organisme client dans Interops-P ou du fournisseur d'identité dans SAML 2.0.
- **Opérateur de service** : organisme hébergeant un service offert aux utilisateurs. Il vérifie et consomme le VI pour contrôler l'accès au service.

  C'est l'équivalent de l'organisme fournisseur dans Interops-P ou du fournisseur de service dans SAML 2.0.

Chaque organisme peut jouer plusieurs rôles en même temps vis-à-vis des autres organismes. Ainsi, dans le cas le plus ordinaire, le couple opérateur d'identification et d'authentification sont confondus comme c'est le cas pour le fournisseur d'identité (IDP) dans [SAMLProf]. De même un opérateur de service peut être opérateur d'authentification.



L'opérateur d'identification et les échanges entre l'opérateur d'identification et d'authentification ne sont pas abordés dans ce document mais dans [R6]. [R6] précise également le format d'échange des traces propre à l'échange de contexte et donc aux échanges entre opérateur d'identification et d'authentification.

Les organismes de la sphère de confiance partagent un format de VI commun, dont le format d'identifiant et la liste d'attributs de l'utilisateur. L'identifiant échangé entre les organismes peut être différent de l'identifiant utilisé pour authentifier l'utilisateur. La valeur de cet identifiant est

partagé par tous les organismes de la sphère de confiance et établi de manière implicite : aucun processus impliquant l'utilisateur n'est nécessaire pour le définir. C'est par exemple le NIR qui est un identifiant d'un utilisateur entre plusieurs organismes. Il est appelé **clé de fédération**.

Alors qu'Interops-P 1.0 ne permettait qu'une cinématique d'authentification à partir de l'organisme client, Interops-S offre la possibilité de rediriger l'utilisateur vers son opérateur d'authentification à partir d'un opérateur de service. Interops-S permet de déterminer quel est l'opérateur d'authentification utilisé par un utilisateur à partir d'un opérateur de service de façon à initier une cinématique d'authentification. La cinématique d'authentification à partir de l'opérateur d'authentification suit la cinématique d'Interops-P avec quelques adaptations précisées au paragraphe 2.1 ci-dessous.

Interops-S propose également un mécanisme de déconnexion globale permettant de déconnecter un utilisateur sur l'opérateur d'authentification et l'ensemble des opérateurs de service sur lesquels il s'est connecté.

Les mécanismes suivants sont mis en œuvre pour offrir les nouvelles cinématiques :

- Service de découverte : permet de déterminer l'opérateur d'authentification d'un utilisateur auprès duquel il s'authentifiera ;
- Cinématique d'authentification à partir d'un opérateur de service : un utilisateur pourra se connecter à une partie publique d'un opérateur de service, puis s'authentifier sur son opérateur d'authentification afin d'accéder à une partie privée de l'opérateur de service ;
- Déconnexion globale : un utilisateur pourra se déconnecter en une seule action sur tous les opérateurs de service sur lesquels il s'est connecté et son opérateur d'authentification.

Ces nouveaux échanges s'appuient sur les grands principes édictés pour l'élaboration d'Interops-P [R1] et [R3] :

- Le modèle repose sur la confiance entre les organismes
- L'authentification de l'utilisateur n'est pas effectuée de bout en bout mais est réalisée par l'opérateur d'authentification
- Les habilitations sont attribuées par l'opérateur d'authentification en respectant les règles établies avec l'opérateur de service (Convention)
- L'habilitation est transmise à l'opérateur de service de manière sécurisée (par un Vecteur d'identification)
- Toute création de vecteur d'identification est auditable afin d'en permettre le contrôle « a posteriori »

En outre, les principes de sécurité et d'auditabilité sont élargis aux nouveaux mécanismes.

2.1. Adaptation d'Interops-P dans le cadre de la sphère de confiance

Le proxy côté organisme client tel que défini dans Interops-P est supprimé des échanges Interops-S.

Ainsi, l'authentification mutuelle entre le proxy client et le reverse-proxy fournisseur est supprimée. Seule est conservée une authentification serveur pour sécuriser les flux entre le navigateur et l'opérateur de service :

- Authentification de l'opérateur de service
- Intégrité des échanges
- Confidentialité des échanges

Les noms de domaines (au sens DNS) entre les opérateurs d'authentification et de service sont exposés au navigateur : **le poste de l'utilisateur voit donc et doit résoudre le nom de domaine des opérateurs d'authentification, de service et du service de découverte de la sphère de confiance**.

Les nouvelles cinématiques d'authentification et de connexion à l'application sont décrites au paragraphe 2.3 p13.

2.2. Service de découverte

Le service de découverte doit permettre de :

- Sauvegarder l'opérateur d'authentification de l'utilisateur
- Déterminer l'opérateur d'authentification de l'utilisateur pour tout opérateur de service de la sphère de confiance.

La sauvegarde de l'opérateur d'authentification se fait au travers d'un cookie sur le navigateur de l'utilisateur contenant un identifiant du ou des opérateurs d'authentification ayant procédé à l'authentification.

Il est également possible de supprimer un opérateur d'authentification du cookie lors d'une déconnexion. Egalement, pour permettre à l'utilisateur de choisir à nouveau son opérateur d'authentification, il est possible de supprimer la valeur du cookie et de rediriger l'utilisateur vers le service de découverte en mode actif.

Un cookie est attaché à un nom de domaine et n'est accessible en lecture et en écriture qu'à partir de ce nom de domaine. Le cookie doit rester visible par tous les services de découverte de la sphère de confiance. De ce fait, le service de découverte n'est pas nécessairement dans le même domaine DNS que l'opérateur d'authentification ou de service. Il utilise un nom de domaine commun aux opérateurs de la sphère de confiance. Chaque opérateur peut ainsi héberger son propre service de découverte et en limiter les fonctions (lecture ou écriture) par rapport au rôle joué dans la sphère de confiance.

La durée de vie du cookie est paramétrable :

- Cookie de session : à la fermeture du navigateur, le cookie est détruit ;
- Cookie permanent : le cookie est conservé sur le navigateur pendant une durée prédéterminée.

L'opérateur d'authentification de l'utilisateur peut être retrouvé à l'aide de l'un des modes décrit au §2.2.2 p11:

- Un mode passif qui ne sollicite pas d'action de l'utilisateur en s'appuyant sur la valeur positionnée dans le cookie ;
- Un mode actif qui offre la possibilité à l'utilisateur de sélectionner son opérateur d'authentification parmi une liste proposée.

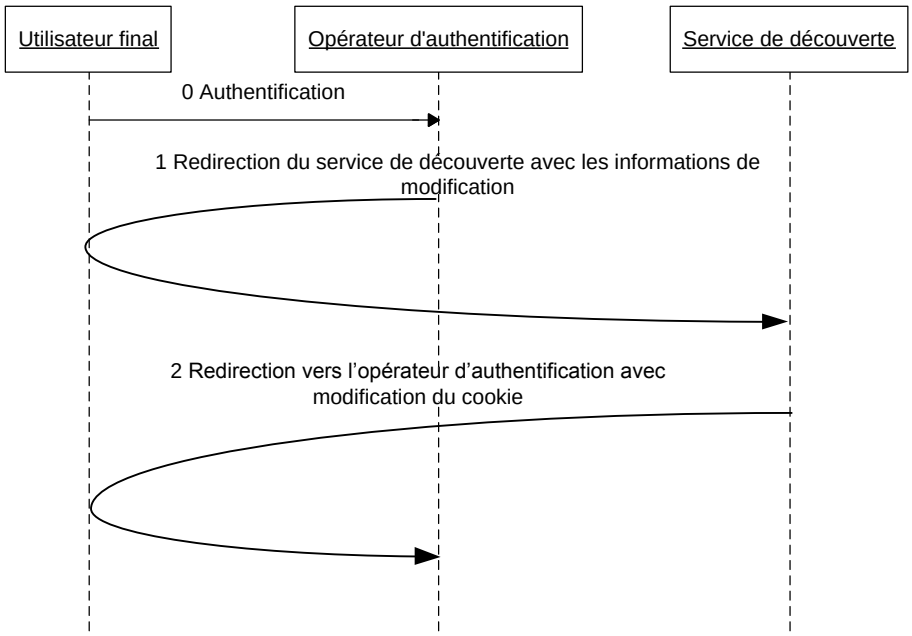
Le format du cookie suivra les recommandations de SAML 2.0 [SAMLProf]

La cinématique de détermination de l'opérateur d'authentification s'appuie sur les travaux de [SAMLDisco].

2.2.1. Modification de l'opérateur d'authentification

L'opérateur d'authentification de l'utilisateur peut faire une demande explicite au service de découverte pour sauvegarder l'information. Ce cookie est lu ultérieurement pour connaître l'opérateur d'authentification (cf. 2.2.2 p11). Il est également possible pour l'opérateur d'authentification de l'utilisateur de supprimer cette information du cookie.

La cinématique est décrite sur le schéma ci-dessous :



0. L'utilisateur s'authentifie auprès de l'opérateur d'authentification.

1. L'opérateur d'authentification redirige l'utilisateur vers le service de découverte en précisant l'identifiant de l'opérateur et le comportement attendu par le service de découverte (ajout, écrasement, suppression, etc.). La liste des paramètres de la requête est définie au paragraphe 3.3.2.1 p21.

2. Le service de découverte redirige l'utilisateur vers l'opérateur d'authentification en prenant soin de modifier le cookie permettant le stockage de l'opérateur d'authentification en fonction du comportement attendu :

- Ajout,
- Ecrasement,
- Suppression,
- Etc.

Le format de la réponse est défini au paragraphe 3.3.2.1 p21. Le format du cookie écrit par le service de découverte est défini au paragraphe 3.3.1 p21.

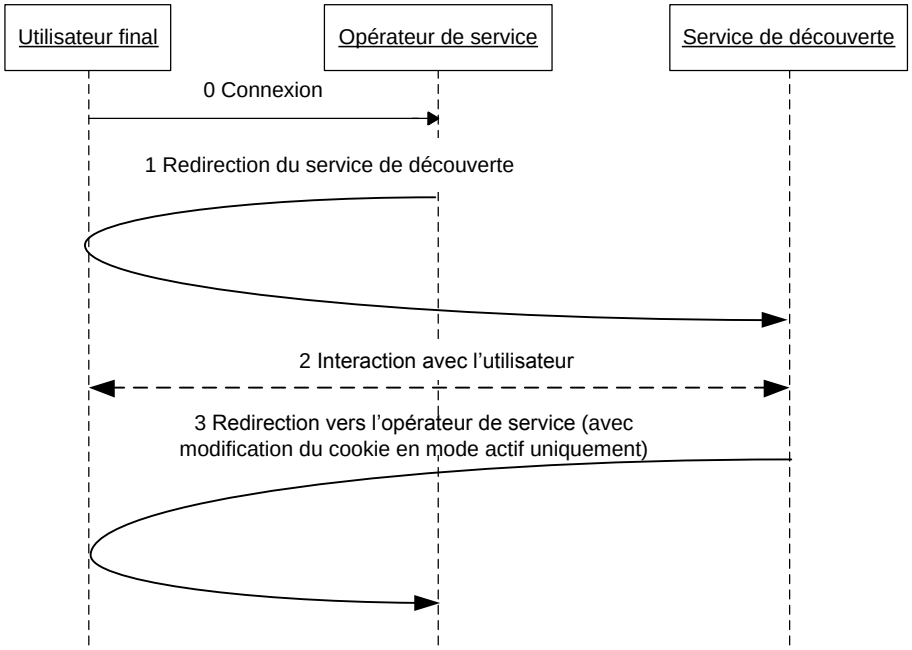
Le service de découverte devrait être configuré pour pouvoir vérifier que l'identifiant de l'opérateur présent dans une demande de modification fait partie des identifiants attendus d'opérateur d'authentification.

Le service de découverte devrait également limiter les URL de retours possibles en se limitant aux noms de domaine présents au sein des conventions applicables.

2.2.2.Récupération de l'opérateur d'authentification

L'opérateur de service peut chercher à connaître l'opérateur d'authentification auquel est affilié l'utilisateur afin de faire une demande d'authentification (cf. §2.3 p13).

La cinématique est décrite sur le schéma ci-dessous :



0. L'utilisateur se connecte à l'opérateur de service. L'utilisateur n'est alors pas authentifié. L'opérateur de service cherche à connaître l'opérateur d'authentification de l'utilisateur.

1. L'opérateur de service redirige l'utilisateur vers le service de découverte pour demander l'identifiant de l'opérateur d'authentification en précisant dans la requête :

- L'identifiant de l'opérateur de service
- Le comportement attendu par le service de découverte (mode passif/actif, ne retourne qu'un seul identifiant ou tous, etc.)
- L'URL de retour

La liste des paramètres est définie au paragraphe 3.3.2.2 p22.

2. Le service de découverte lit le cookie partagé et récupère l'identifiant de l'opérateur d'authentification. Si le cookie est inexistant ou si aucun opérateur d'authentification n'est présent, deux comportements sont possibles en fonction du mode choisi :

- Passif : il n'existe aucune interaction entre le service de découverte et l'utilisateur en dehors des redirections. Le service de découverte redirige l'utilisateur vers l'opérateur de service avec en réponse un opérateur d'authentification vide
- Actif : si aucun cookie n'existe ou si aucun identifiant correspondant n'est trouvé, le service de découverte présente une page avec une liste d'opérateurs d'authentification avec lesquels il existe une convention et pour lesquels l'opérateur joue le rôle d'opérateur de service. Ainsi, l'utilisateur peut préciser son opérateur d'authentification. Le service de découverte sauvegarde le choix de l'utilisateur dans le cookie avant de rediriger l'utilisateur vers l'opérateur de service avec cet identifiant d'opérateur d'authentification en paramètre

Le format du cookie lu et écrit par le service de découverte est défini au paragraphe 3.3.1 p21.

3. Le service de découverte redirige l'utilisateur vers l'opérateur de service en retournant dans l'URL les informations de l'opérateur d'authentification si elles sont connues. Le format de la réponse est défini au paragraphe 3.3.2.2 p22.

2.3. Cinématique d'authentification

2.3.1. Cinématique globale

Des opérateurs de service peuvent fournir des services avec une partie publique et une partie sécurisée nécessitant une authentification d'un des opérateurs d'authentification de la sphère de confiance.

Les utilisateurs peuvent alors naviguer sur la partie publique et décider d'accéder à des services sécurisés. L'opérateur de service doit alors :

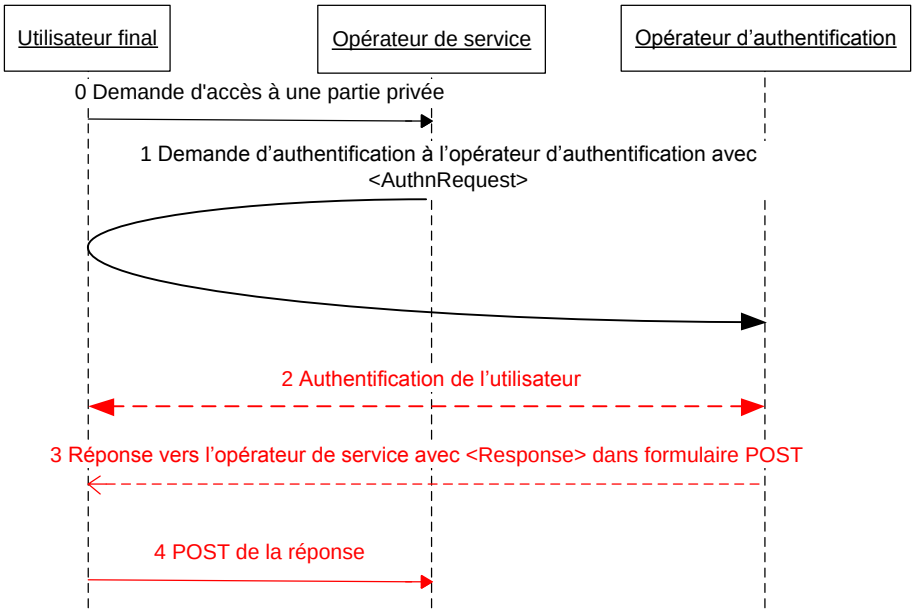
- Déterminer l'opérateur d'authentification de l'utilisateur grâce au service de découverte (cf. §2.2.2 p11)
- Faire une demande d'authentification de l'utilisateur auprès de l'opérateur d'authentification

La demande d'authentification se fait grâce à l'élément <AuthnRequest> de SAML 2.0 ([SAMLCore2])

Le diagramme ci-dessous présente une cinématique complète d'authentification à partir de l'opérateur de service, correspondant à une cinématique « WebSSO Profile SP-Initiated » au sens SAML 2.0.

Les étapes 2 à 4 en rouge correspondent à un échange d'un VI dans le cadre d'une cinématique Interops-P décrit au paragraphe 2.3.2 p14.

> *Dans le cas où seules les étapes 2 à 4 sont mise en œuvre, la cinématique d'authentification est dite initiée au niveau de l'opérateur d'authentification, correspondant à une cinématique « WebSSO Profile IdP-Initiated » au sens SAML 2.0.*



0. L'utilisateur se connecte à l'opérateur de service et accède à une zone sécurisée nécessitant d'être authentifié. Si l'opérateur d'authentification est inconnu, alors il est possible de le déterminer grâce à la cinématique de récupération de l'opérateur d'authentification (cf. §2.2.2 p11). Si l'opérateur d'authentification reste indéterminé, alors l'opérateur de service peut remonter une erreur à l'utilisateur ou authentifier localement l'utilisateur.

1. L'opérateur de service émet une demande d'authentification signée à destination de l'opérateur d'authentification en précisant le service visé. L'opérateur de service transmet également dans le RelayState toute information utile au suivi de la requête comme URL de l'application visée, etc.

Le format de la requête est défini au paragraphe 3.5.1 p24. La méthode de transmission de la requête d'authentification entre l'opérateur de service et l'opérateur d'authentification est décrite au paragraphe 3.4 p22.

2. L'opérateur d'authentification vérifie :

- Le format de la requête d'authentification
- La période de validité de la requête connaissant la date et l'heure de création de la requête et la durée de validité autorisée
- L'identifiant de la requête pour éviter un rejeu
- La signature de la requête d'authentification

Si l'utilisateur n'est pas déjà authentifié, l'opérateur d'authentification authentifie l'utilisateur.

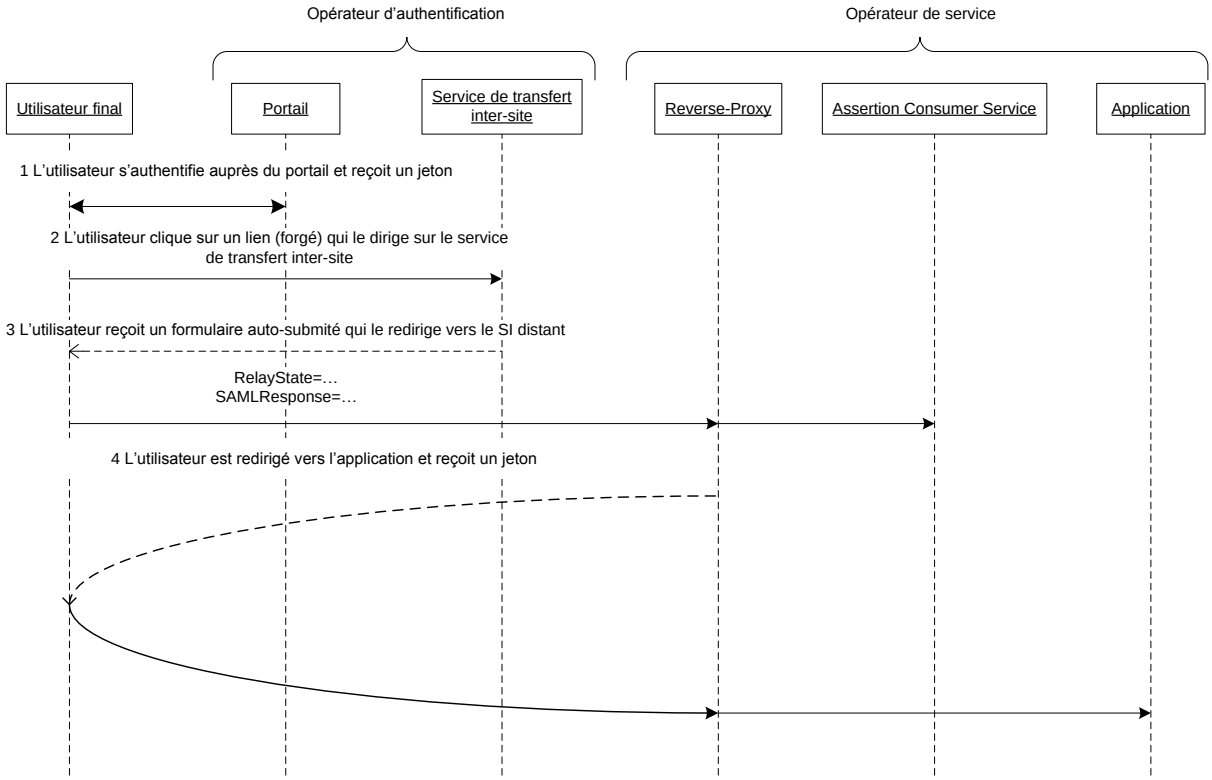
3. L'opérateur d'authentification retourne dans un formulaire auto-soumis à destination de l'opérateur de service une réponse contenant les informations d'authentification, c'est-à-dire le VI. Le RelayState transmis à l'étape 1 est également transmis sans modification en paramètre du formulaire dans le champ RelayState.

4. Le navigateur transmet la réponse contenant le VI.

2.3.2. Cinématique détaillée de réponse et de transfert du VI

2.3.2.1. Cinématique d'authentification lors de la première connexion

Les échanges à la première connexion à l'application de l'utilisateur respectent le profil Web SSO/POST de SAML 2.0 et sont représentés sur la figure ci-dessous.



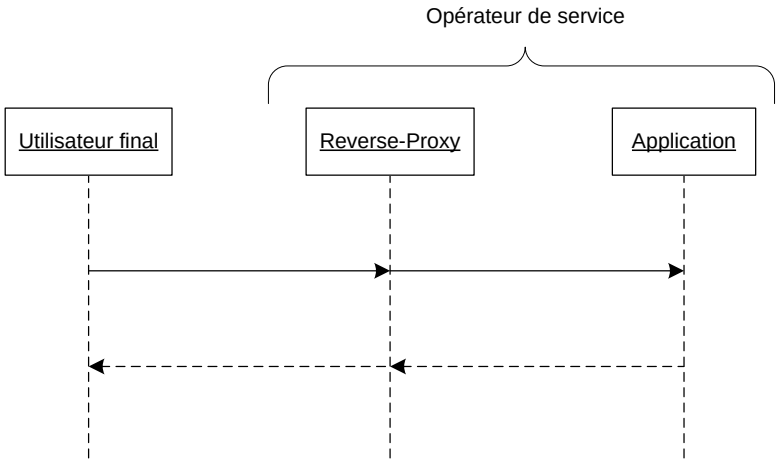
1. L'utilisateur s'authentifie sur un portail situé dans l'opérateur d'authentification et obtient un jeton d'authentification afin de ne pas se réauthentifier à chaque échange avec le portail. Un lien vers l'application lui est présenté de façon à le rediriger vers le service de transfert inter-site.

2. L'utilisateur clique sur le lien et est dirigé vers le service de transfert inter-site avec en paramètre le service visé.
3. Le service de transfert inter-site génère le VI et retourne un formulaire Web au navigateur, auto-soumis par JavaScript, avec :
   - Le paramètre « action » du formulaire contenant l'URL de l'Assertion Consumer Service de l'opérateur de service
   - Le champ caché « SAMLResponse » contenant une réponse SAML (qui joue le rôle de vecteur d'identification) encodée en base64 (cf. [R3])
   - Le champ caché « RelayState » contenant la même valeur que celle transmise avec la requête d'authentification par l'opérateur de service

   La communication entre le navigateur de l'utilisateur final et l'Assertion Consumer Service traverse le reverse-proxy de l'opérateur de service, de manière transparente pour l'utilisateur.
4. L'Assertion Consumer Service intègre toute la logique SAML côté opérateur de service. Il vérifie les données contenues dans la réponse SAML et crée un contexte de sécurité pour l'utilisateur. Finalement, l'utilisateur est redirigé vers l'application et reçoit un jeton d'authentification propre à l'opérateur de service.

2.3.2.2. Cinématiques des échanges après la première connexion

Les échanges de transaction effectués après la première connexion de l'utilisateur à l'application sont représentés sur la figure ci-dessous :



L'utilisateur est déjà reconnu par l'application. Toutes les requêtes et les réponses de l'application se font sans intervention de l'opérateur d'authentification. Ils sont simplement protégés par le reverse-proxy de l'opérateur de service. De ce fait, il n'existe pas de communication triangulaire entre les flux Utilisateur – Opérateur d'authentification – Opérateur de service si l'utilisateur est sur Internet.

Les échanges de transaction sont identiques que la cinématique d'authentification ait démarré à partir de l'opérateur d'authentification ou de l'opérateur de service.

2.4. Déconnexion globale

La déconnexion globale doit permettre à un utilisateur de se déconnecter, c'est-à-dire d'invalider toute session en cours, avec tous les opérateurs de service auprès desquels il s'est connecté ainsi qu'auprès de son opérateur d'authentification.

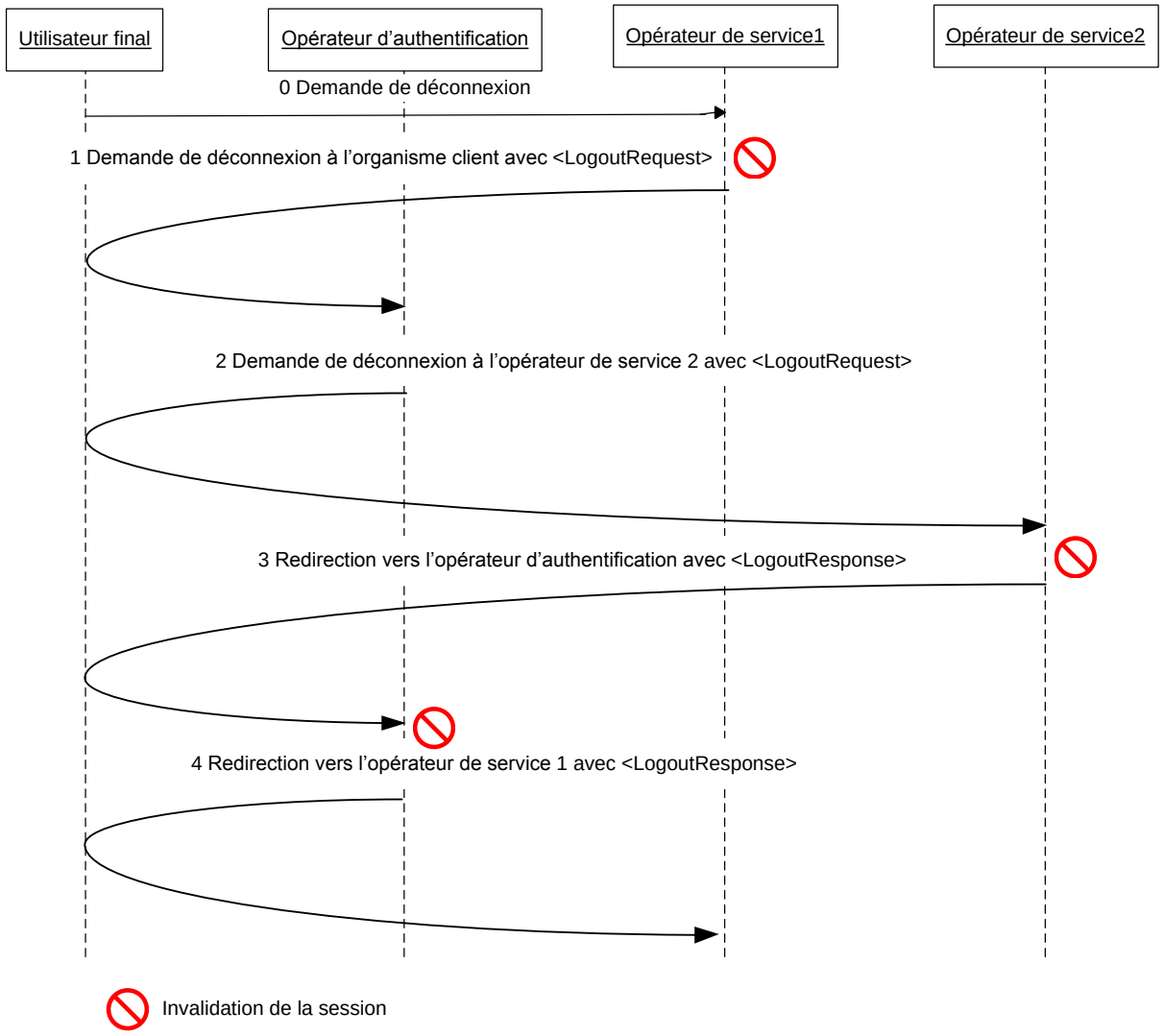
L'opérateur d'authentification a la charge de maintenir la liste des opérateurs de service auprès duquel l'utilisateur s'est connecté.

L'utilisateur peut faire une demande de déconnexion globale auprès d'un opérateur de service sur lequel il s'est connecté ou auprès de son opérateur d'authentification.

Les cinématiques de déconnexion globale s'appuient sur SAML 2.0.

La cinématique suivante présente une déconnexion globale initiée à partir de l'opérateur de service1 pour un utilisateur s'étant authentifié et s'étant aussi connecté sur un opérateur de service2.

Dans le cas où la demande est initiée à partir de l'opérateur d'authentification, l'étape 1 ci-dessous est omise.



0. L'utilisateur s'est authentifié sur l'opérateur d'authentification et connecté auprès des opérateurs de service 1 et 2. Il demande alors une déconnexion globale.

1. L'opérateur de service 1 invalide toute session ouverte pour le compte de l'utilisateur. Il génère une demande de déconnexion sous la forme d'une <LogoutRequest>, la signe et la transmet à l'opérateur d'authentification. **Cette étape est optionnelle si l'utilisateur fait la demande de déconnexion globale directement auprès de l'opérateur d'authentification**.

Le format de la demande de déconnexion est décrit au paragraphe 3.7.1 p30.

Le mécanisme de transmission de la requête est décrit au paragraphe 3.4 p22.

2. L'opérateur d'authentification vérifie :

- Le format de la demande de déconnexion

- L'adéquation de la demande à l'utilisateur et à l'identifiant de session (égal à l'identifiant du VI)
- La période de validité de la requête connaissant la date et l'heure de création de la requête et la durée de validité autorisée
- L'identifiant de la requête pour éviter un rejeu
- La signature de la requête de déconnexion

L'opérateur d'authentification propage la demande de déconnexion globale en générant des <LogoutRequest> signées pour chacun des opérateurs de service auprès desquels l'utilisateur s'est authentifié, excepté l'opérateur de service émetteur de la demande. Dans notre cas, la demande est transmise uniquement à l'opérateur de service2.

Le formalisme et le mécanisme de transmission de la requête sont identiques à l'étape précédente.

3. L'opérateur de service 2 vérifie la demande de déconnexion comme l'opérateur d'authentification à l'étape 2 et invalide toute session encore ouverte pour le compte de l'utilisateur et transmet une réponse à la demande de déconnexion sous forme de <LogoutResponse> signée. Si la session est déjà expirée, l'opérateur de service ne doit pas retourner d'erreur.

Le format de la réponse de déconnexion est décrit au paragraphe 3.7.2 p31.

Le mécanisme de transmission de la réponse est identique à la requête et est décrit au paragraphe 3.4 p22.

4. L'opérateur d'authentification vérifie :

- Le format de la réponse de déconnexion
- L'identifiant de la réponse pour éviter un rejeu
- La période de validité de la réponse connaissant la date et l'heure de création de la réponse et la durée de validité autorisée
- La signature de la réponse

L'opérateur d'authentification invalide la session et redirige l'utilisateur vers l'opérateur de service1 avec un <LogoutResponse>.

Note : La transmission des requêtes et des réponses de déconnexion décrites aux étapes 2 et 3 peuvent être conduites en parallèle sur l'ensemble des opérateurs de service en fonction des implémentations.

2.5. Maintien de session

Pour éviter l'expiration de la session de l'utilisateur sur son opérateur d'authentification, un mécanisme de maintien de session doit être mis en œuvre entre l'opérateur de service et l'opérateur d'authentification. Ceci permettra à l'utilisateur de rafraîchir sa session sur l'opérateur d'authentification lors de sa navigation sur l'opérateur de service.

Chaque opérateur d'authentification de la sphère de confiance pourra proposer l'URL d'un composant dynamique retournant une image transparente d'un pixel. Cette URL est directement intégrée à l'application de l'opérateur de service de façon à ce qu'un échange existe entre le navigateur de l'utilisateur pendant la navigation de l'utilisateur sur l'application et l'opérateur d'authentification. Ceci aura pour effet de maintenir la session sur l'opérateur d'authentification.

L'utilisation du pixel transparent facilite l'intégration dans les pages Web de l'application de l'opérateur de service.

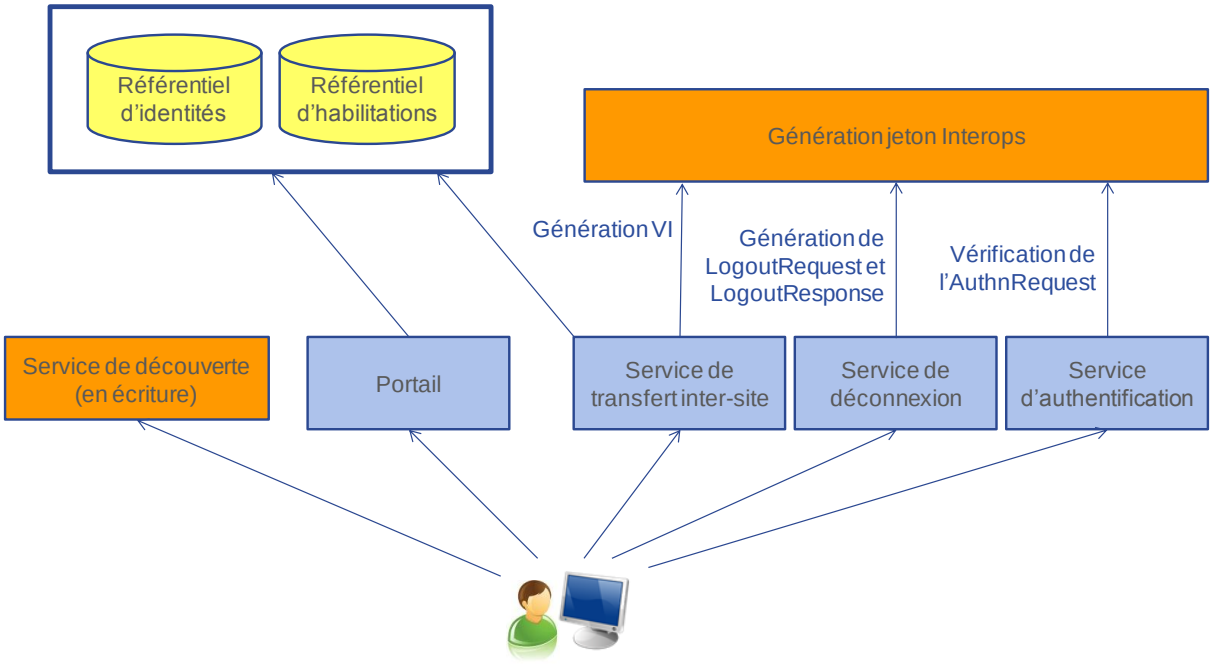
Les durées de session et les délais d'inactivité entre les opérateurs devraient cependant être harmonisés pour éviter au maximum les problèmes.

3. SPECIFICATIONS TECHNIQUES

3.1. Architecture fonctionnelle

3.1.1. Opérateur d'authentification

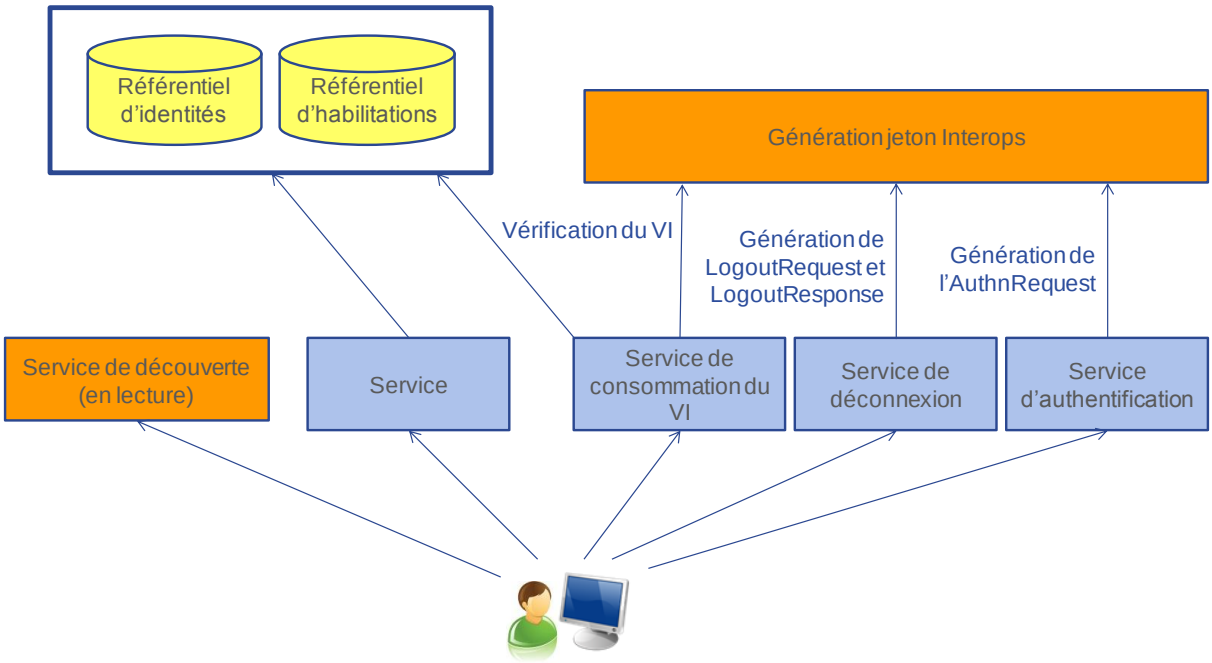
Le schéma ci-dessous présente l'architecture fonctionnelle d'un opérateur d'authentification :



Le portail et le service de transfert inter-site sont hérités de l'architecture Interops-P 1.0.

3.1.2. Opérateur de service

Le schéma ci-dessous présente l'architecture fonctionnelle d'un opérateur de service :



Le service et le service de consommation du VI sont hérités de l'architecture Interops-P 1.0.

3.2. Sécurité des échanges

> ***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. Certificats X509

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

> ***Dans le cas où les certificats devront être conformes au Référentiel Général de Sécurité, se reporter à la note du paragraphe 3.2 p19.***

3.2.2. Echanges entre l'utilisateur et les différents composants

Toutes les communications entre le navigateur et les différents composants devront être protégées par TLS ([TLS]) avec authentification serveur. Les communications seront alors protégées en confidentialité et en intégrité.

De ce fait, la vérification du certificat X.509 présenté par les opérateurs de service est de la responsabilité du navigateur de l'utilisateur. Afin que les données transmises et qui transitent par le navigateur ne soient facilement interceptées, il est nécessaire que le navigateur puisse automatiquement authentifier les serveurs mis en œuvre en se basant sur une autorité de certification reconnue. Il est donc recommandé d'avoir un certificat émis par une autorité de certification reconnue automatiquement par les principaux navigateurs.

- ***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 p19.***

- ***Une des autorités reconnues par le principaux navigateurs et conforme au RGS est l'IGC/A***

3.2.3. Echanges entre les différents composants serveurs

Les communications entre les différents composants serveurs (et particulièrement entre le portail de l'organisme destinataire et le gestionnaire de contexte applicatif) sont sécurisées par TLS ([TLS]) avec authentification mutuelle par certificat X509.

Les communications seront alors protégées en confidentialité et en intégrité.

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 p19.***

3.3. Service de découverte de l'opérateur d'authentification

3.3.1. Format du cookie

Le cookie partagé du service de découverte doit avoir les caractéristiques suivantes :

- Nom : _saml_idp
- Domaine : .<nom de domaine commun>
- Path : /
- Flag secure (Note : pour pouvoir positionner ce flag, le cookie doit être transmis sur HTTPS)

Le domaine correspond au nom de domaine commun entre les opérateurs de la sphère de confiance. Il doit nécessairement être précédé d'un point.

La valeur du cookie correspond à la concaténation de l'encodage en base64 de chaque URI d'opérateur d'authentification séparé par un espace. La valeur finale est URL-encodée pour être transportable en tant que cookie.

3.3.2. Paramètres des requêtes

Les requêtes HTTP faites au service de découverte sont des requêtes HTTP GET.

Le service de découverte répondra par une redirection HTTP (code 302) et retournera les informations dans les paramètres de la requête.

Les paramètres passés dans l'URL sont décrits ci-après. Chacun des paramètres doit être « URL-encodés ».

3.3.2.1. Modification de l'opérateur d'authentification

➢ **Requête**

Les paramètres d'entrée du service de découverte pour la modification de l'opérateur d'authentification sont :

- `entityID` : identifiant de l'opérateur d'authentification faisant la demande
- `return` : URL de redirection.
- `policy` (optionnel) : comportement attendu :
  - `append` (par défaut) : l'identifiant est ajouté à la liste des identifiants. Si l'identifiant était déjà présent, il peut être supprimé et ajouté à la fin pour stipuler le dernier opérateur d'authentification visité par l'utilisateur
  - `set` : le service de découverte écrase la valeur du cookie actuelle pour ne mettre que l'identifiant de l'opérateur d'authentification
  - `remove` : l'identifiant de l'opérateur d'authentification est supprimé du cookie

➢ **Réponse**

La réponse est retournée à l'URL spécifié par le paramètre `return`.

Si ce paramètre était absent, le service de découverte peut définir une URL de retour par défaut. Sinon, le service de découverte doit afficher une page d'erreur.

Le service de découverte ajoute pour une requête de modification de l'opérateur d'authentification le paramètre `status` à l'URL de retour. L'URL de retour peut déjà contenir des paramètres qui doivent être conservés en dehors du paramètre `status`.

Le paramètre `status` indique le résultat de l'opération. Il peut prendre les valeurs suivantes :

- `OK` : le traitement s'est déroulé correctement
- `UNKNOWN_ENTITYID` : Identifiant d'opérateur d'authentification inconnu
- `MISSING_ENTITYID` : Identifiant d'opérateur absent de la requête
- `UNKNOWN_POLICY` : Identifiant de politique non supporté

3.3.2.2. Récupération de l'opérateur d'authentification

➢ **Requête**

Les paramètres d'entrée du service de découverte pour la récupération de l'opérateur d'authentification sont :

- `entityID` : identifiant de l'opérateur de service faisant la demande
- `return` : URL de redirection
- `policy` (optionnel) : comportement attendu :
  - `urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol:single` : un seul identifiant d'opérateur est retourné
- `returnIDParam` (optionnel) : paramètre contenant le ou les identifiants de retour
- `isPassive` : mode attendu du service de découverte : passif si la valeur est `true`, actif si la valeur est `false` (par défaut). Si la valeur est incorrecte, la valeur par défaut est utilisée.

➢ **Réponse**

La réponse est retournée à l'URL spécifié par le paramètre `return`.

Si ce paramètre était absent, le service de découverte peut définir une URL de retour par défaut. Sinon, le service de découverte doit afficher une page d'erreur.

Le service de découverte ajoute pour une requête de récupération de l'opérateur d'authentification à l'URL de retour les paramètres suivants :

- `entityID` ou la valeur de `returnIDParam` : il contient l'identifiant de l'opérateur d'authentification associé à l'utilisateur ou une chaîne vide si aucun opérateur n'est trouvé.
- `status` : contient le résultat de l'opération. Les valeurs possibles sont :
  - `OK` : le traitement s'est déroulé correctement
  - `UNKNOWN_ENTITYID` : Identifiant d'opérateur d'authentification inconnu
  - `MISSING_ENTITYID` : Identifiant d'opérateur absent de la requête
  - `UNKNOWN_POLICY` : Identifiant de politique non supporté

L'URL de retour peut déjà contenir des paramètres qui doivent être conservés en dehors des paramètres `entityID` ou de la valeur de `returnIDParam` et `status`.

La réponse ne doit pas inclure le paramètre `entityID` ou la valeur de `returnIDParam` en cas d'erreur.

3.4. Binding HTTP-Redirect

3.4.1. Présentation

Le *binding* HTTP-Redirect permet la transmission d'un message entre deux organismes en passant par le navigateur de l'utilisateur.

Il s'appuie sur les requêtes HTTP GET et place les informations dans les paramètres de l'URL de l'organisme destinataire (opérateur d'authentification ou opérateur de service).

3.4.2.Encodage des messages

Le *binding* utilisé dans cette spécification suit l'encodage `urn:oasis:names:tc:SAML:2.0:bindings:URL-Encoding:DEFLATE` de la spécification de SAML 2.0 (cf. [SAMLBinding]).

La transformation à appliquer sur un message XML est la suivante :

1. Toute signature XML dont l'élément <ds:Signature> doit être supprimé
2. L'algorithme de compression DEFLATE (cf. [RFC1951]) est utilisé sur les données XML restantes
3. Le résultat doit être encodé en base64. Tout retour à la ligne ou espace doit être supprimé du résultat final
4. La valeur résultante doit être URL-encodée et placée dans le paramètre `SAMLRequest` ou `SAMLResponse` suivant qu'il s'agit d'une requête SAML ou une réponse. C'est cette valeur qui servira à la signature.
5. Le RelayState doit être URL-encodé et placé dans le paramètre `RelayState`

La requête ou la réponse sont signées en suivant la procédure suivante :

1. L'algorithme de signature utilisé doit être spécifié dans le paramètre d'URL `SigAlg`. La valeur doit être URL-encodée et correspondre à l'URI d'un algorithme tel que défini par [XMLDsig]. Toute implémentation doit supporter *a minima* RSAwithSHA1 (http://www.w3.org/2000/09/xmldsig#rsa-sha1)

> ***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 p19.***

2. Une chaîne de caractère est construite en concaténant le paramètre `SAMLRequest` ou `SAMLResponse` et sa valeur, le paramètre `RelayState` et sa valeur et le paramètre `SigAlg` et sa valeur comme suit :

```
SAMLRequest=value&RelayState=value&SigAlg=value
SAMLResponse=value&RelayState=value&SigAlg=value
```

3. La chaîne de caractères résultante est utilisée pour calculer la signature
4. Le résultat de la signature est ensuite encodé en base64 sans retour à la ligne et sans espace, puis URL-encodé et placé dans le paramètre `Signature` de l'URL.

Pour vérifier la signature, il est nécessaire de :

- Reconstruire la chaîne de caractère en respectant l'ordre des paramètres
- Conserver l'encodage URL initial pour vérifier la signature

3.4.3. En-tête HTTP

Pour éviter une mise en cache des messages échangés par des équipements intermédiaires, il est recommandé d'utiliser les entêtes suivants en HTTP 1.1 :

- Entête `Cache-Control` avec la valeur « `no-cache, no-store` ».
- Entête `Pragma` avec la valeur « `no-cache` ».

3.5. Initiation de la cinématique d'authentification à partir d'un opérateur de service

3.5.1. Format de la requête d'authentification

Le format d'une requête d'authentification est décrit ci-dessous. Il suit le format d'une requête d'authentification SAML 2.0. Cette requête est transmise en utilisant le binding HTTP-Redirect.

Chaque mot écrit entre crochets en gras rouge (ex : **[ID]**) est une variable paramétrée dont la valeur est définie ci-dessous.

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="[ID]" Version="2.0"
    IssueInstant="[IssueInstant]" Destination="[Destination]">
    <saml:Issuer>[Issuer]</saml:Issuer>
    <samlp:NameIDPolicy     Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:transient"/>
    <Conditions>
      <AudienceRestriction>
         <Audience>[Audience]<Audience>
      </AudienceRestriction>
    </Conditions>
</samlp:AuthnRequest>
```

Nom	Description	Format	Exemple
ID	Identifiant unique de la requête	Le format de l'identifiant doit suivre les recommandations de la RFC 4122 [RFC4122] afin d'assurer l'unicité de l'identifiant.	`uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
IssueInstant	Instant de génération de la requête	Valeur de type type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire.	`2003-04-17T00:46:02Z`
Issuer	Identifiant de l'émetteur de la requête (opérateur de service)	Le format de l'identifiant est une URI et suit les mêmes recommandations que pour le VI [R2].	`urn:interops:{SIREN|SIRET}:sp:{libre}`
Destination	URL identifiant l'adresse du service de réception des demandes d'authentification	Cette URL correspond à la page vers laquelle l'utilisateur est redirigé. L'opérateur d'authentification doit vérifier que la valeur de ce champ correspond bien à l'adresse à laquelle elle a été reçue, c'est-à-dire que la requête lui est bien destinée.	`https://www.cnav.fr/sso`
Audience	Identifiant du service visé	Identifiant sous forme d'URI décrivant le service visé. Il est recommandé de décrire le service à l'aide d'une URL, comprenant le nom de domaine de l'opérateur de service et le nom du service	`https://rniam.cnav.fr`

3.5.2. Gestion des erreurs

En cas d'erreur lors du traitement de la requête par l'opérateur d'authentification, une réponse d'erreur doit être retournée à l'opérateur de service (cf. § 3.6.2 p28).

3.6. Authentification et transfert du VI

> *Dans le cas d'une cinématique dite « IdP-Initiated » au sens SAML 2.0, la cinématique d'authentification est initiée au niveau de l'opérateur d'authentification et démarre à cette étape.*

3.6.1. Format d'une réponse valide

Le format d'une réponse d'authentification et de transfert du VI suit le format SAML 2.0 et les spécifications du VI [R2].

La réponse est obligatoirement signée.

Chaque mot écrit entre crochets en gras rouge (ex : **[ID]**) est une variable paramétrée dont la valeur est définie ci-dessous.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response Destination="[Destination]"
IssueInstant="[IssueInstant]" ID="[ID]" Version="2.0"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
InResponseTo="[SpRequestId]">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">[Issuer]</saml:Issu
er>
  <samlp:Status>
    <samlp:StatusCode
Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
<saml2:Assertion Version="2.0" IssueInstant="[IssueInstant]" ID="[ID]"
xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:SAML:2.0:assertion
http://docs.oasis-open.org/security/saml/v2.0/saml-schema-assertion-
2.0.xsd">
  <saml2:Issuer>[Issuer]</saml2:Issuer>
  <saml2:Subject>
    <saml2:NameID
Format="[SubjectFormat2]">[SubjectId2]</saml2:NameID>
    <saml2:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <saml2:SubjectConfirmationData NotOnOrAfter="[NotOnOrAfter]"
InResponseTo="[SpRequestId]" Recipient="[Recipient]"/>
    </saml2:SubjectConfirmation>
  </saml2:Subject>
  <saml2:Conditions NotOnOrAfter="[NotOnOrAfter]"
NotBefore="[NotBefore]">
    <saml2:AudienceRestriction>
      <saml2:Audience>[Audience]</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml2:Conditions>
  <saml2:AuthnStatement AuthnInstant="[AuthnInstant]" SessionIndex="[
AssertionId]">
    <saml2:AuthnContext>
```

```
<saml2:AuthnContextClassRef>[MethodAuthn2]</saml2:AuthnContextClassRef
>
    </saml2:AuthnContext>
  </saml2:AuthnStatement>
  <saml2:AttributeStatement>
    <saml2:Attribute Name="PAGM">
      <saml2:AttributeValue>[PAGM]</saml2:AttributeValue>
    <saml2:Attribute Name="[AttributeName]">
      <saml2:AttributeValue>[AttributeValue]</saml2:AttributeValue>
      <saml2:AttributeValue>[AttributeValue]</saml2:AttributeValue>
    </saml2:Attribute>
    <saml2:Attribute Name="[AttributeName]">
      <saml2:AttributeValue>[AttributeValue]</saml2:AttributeValue>
      <saml2:AttributeValue>[AttributeValue]</saml2:AttributeValue>
    </saml2:Attribute>
  </saml2:Attribute>
  </saml2:AttributeStatement>
</saml2:Assertion>
</samlp :response>
```

3.6.1.1. Eléments propres à la réponse

Les éléments propres à l'élément <Response> sont décrits dans le tableau ci-dessous :

Nom	Description	Format	Exemple
ID	Identifiant unique de la réponse	Le format de l'identifiant doit suivre les recommandations de la RFC 4122 [RFC4122] afin d'assurer l'unicité de l'identifiant	`f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
IssueInstant	Instant de génération de la réponse	Valeur de type type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire	`2003-04-17T00:46:02Z`
Issuer	Identification de l'émetteur de la réponse, et donc de l'opérateur d'authentification	Le format de l'identification est un URI. L'opérateur d'authentification doit être identifié par un URI, contenant le numéro de version de l'accord d'interopérabilité Cet identifiant est identique à l'élément `Issuer` de l'assertion	`urn:interops:idp:{SIREN|SIRET}:{libre}:version`
Destination	URI identifiant l'adresse du service de réception des assertions	Cette URL correspond à la valeur du paramètre « action » du formulaire utilisé pour soumettre la réponse SAML. L'opérateur de service doit vérifier que la valeur de ce champ correspond bien à l'adresse à laquelle elle a été reçue.	`https://www.exemple.com:9031/sp/ACS.saml2`
SpRequestId	Identifiant unique provenant de la requête d'authentification	Le format de l'identifiant doit suivre les recommandations de la RFC 4122 [RFC4122] afin d'assurer l'unicité de l'identifiant	`f81d4fae-7dec-11d0-a765-00a0c91e6bf7`

3.6.1.2. Eléments propres à l'assertion

Les éléments propres à l'élément <Assertion> sont décrits dans le tableau ci-dessous :

703

Nom	Description	Format	Exemple
ID	Identifiant unique de l'assertion	Le format de l'identifiant doit suivre les recommandations de la RFC 4122 [RFC4122] afin d'assurer l'unicité de l'identifiant	`uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
IssueInstant	Instant de génération de l'assertion	Valeur de type type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire	`2003-04-17T00:46:02Z`
Issuer	Identification de l'émetteur de l'assertion, et donc de l'opérateur d'authentification	Le format de l'identification est une URI. L'opérateur d'authentification doit être identifié par une URI contenant le numéro de version de l'accord d'interopérabilité	`urn:interops:{SIREN|SIRET}:idp:{libre}:version`
NotOnOrAfter	Date d'expiration de l'assertion La date d'expiration est dépendante de la durée de validité de l'assertion et doit prendre en compte une dérive des horloges des systèmes	Valeur de type type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire	`2003-04-17T00:46:02Z`
NotOnBefore	Date de début de validité de l'assertion La date de début de validité de l'assertion doit prendre en compte une dérive des horloges des systèmes. La date de début de validité doit donc être légèrement avancée par rapport à la date d'émission	Valeur de type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire	`2003-04-17T00:46:02Z`
Audience	Identifiant du service visé	URI décrivant le service visé. Il est recommandé de décrire le service à l'aide d'une URL, comprenant le nom de domaine de l'opérateur de service et le nom du service publics précisés dans l'accord	`http://rniam.cnav.fr`
AuthnInstant	Instant d'authentification de l'usager sur le SI. A moins de disposer de cette information, l'instant d'authentification peut être égal à l'instant de création	Valeur de type type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire	`2003-04-17T00:46:02Z`

	de l'assertion		
PAGM	Liste des PAGM	L'attribut listant les PAGM doit s'appeler `PAGM`. Une liste de PAGM peut être donnée en multipliant les éléments `<AttributeValue>` dans l'élément `<Attribute>` L'AttributeNamespace doit être `urn:iops:attributs:pagm`.	`<Attribute AttributeNamespace="urn:iops:attributs:pagm" AttributeName="PAGM"> <AttributeValue> PAGM1 </AttributeValue> <AttributeValue> PAGM2 </AttributeValue> </Attribute>`
AssertionId	Identifiant de session	L'identifiant de session est égal à l'identifiant de l'assertion, c'est-à-dire du VI	`uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
SubjectFormat2	Identifiant du format de l'identifiant du demandeur pour SAML 2.0	Le format de l'identifiant dépend de l'accord d'interopérabilité.	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
SubjectId2	Identifiant de l'utilisateur	Dépend SubjectFormat2	
MethodAuthn2	Méthode d'authentification de l'utilisateur sur le SI de l'opérateur d'authentification	La méthode d'authentification est une URI. Pour l'ensemble des valeurs normalisées, se reporter à [SAML2AuthnCxt]	`urn:oasis:names:tc:SAML:2.0:ac:classes:Password`
Recipient	Identifiant de l'opérateur de service	URI identifiant l'opérateur de service pouvant recevoir l'assertion	`urn:interops:sp:{SIREN|SIRET}:{libre}`

3.6.2. Format d'une réponse en cas d'erreur

En cas d'erreur suite à une sollicitation d'un opérateur de service, l'opérateur d'authentification doit retourner un élément <Response> avec un code d'erreur. Cet élément est retourné dans un formulaire POST comme le serait le VI.

L'élément <Response> est décrit ci-dessous. Il suit le format d'une réponse SAML 2.0.

Chaque mot écrit entre crochets en gras rouge (ex : **[ID]**) est une variable paramétrée dont la valeur est définie ci-dessous.

```
<?xml version="1.0" encoding="UTF-8"?>
<samlp:Response Destination="[Destination]"
IssueInstant="[IssueInstant]" ID="[ID]" InResponseTo="[SpRequestId]"
Version="2.0" xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol">
  <saml:Issuer
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">[Issuer]</saml:Issuer>
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
              <ds:CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
              <ds:SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
              <ds:Reference URI="#[ID]">
                    <ds:Transforms>
                          <ds:Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
```

```
                                <ds:Transform
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                        </ds:Transforms>
                        <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                        <ds:DigestValue>...</ds:DigestValue>
                </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>
        ...
        </ds:SignatureValue>
  </ds:Signature>
  <samlp:Status>
        <samlp:StatusCode Value="[STATUS]">
           <samlp:StatusCode Value="[STATUS2]"/>
        </samlp:StatusCode>
  </samlp:Status>
</samlp:Response>
```

Ces éléments sont identiques au format de la réponse SAML 2.0 tel que décrit dans [R2].

Le format de la signature suit également les recommandations de [R2].

Seuls varient les champs **[STATUS]** et **[STATUS2]**. L'élément <StatusCode> portant le champ **[STATUS2]** est optionnel.

[STATUS] peut prendre les valeurs :

- urn:oasis:names:tc:SAML:2.0:status:Requester : la requête comporte une erreur
- urn:oasis:names:tc:SAML:2.0:status:Responder : le traitement de la requête a entraîné une erreur côté opérateur d'authentification. Les deux sous-codes suivants peuvent être utilisés pour le champ **[STATUS2]** :
  - urn:oasis:names:tc:SAML:2.0:status:AuthnFailed : l'utilisateur n'a pu être authentifié (blocage du compte, etc.)

3.6.3.RelayState

Dans le cas d'une cinématique d'authentification à partir d'un opérateur de service, l'opérateur de service peut transmettre une chaîne de caractère « opaque ». Elle doit être retournée sans modification à l'opérateur de service par l'opérateur d'authentification.

Dans le cas d'une cinématique d'authentification à partir de l'opérateur d'authentification, l'URL du service visé est positionné dans le RelayState.

Le RelayState ne doit pas dépasser 80 caractères.

Des mécanismes de protection doivent être mis en place par l'opérateur de service pour assurer son intégrité.

3.6.4. Récapitulatif des différences par rapport au format d'une réponse Interops-P

3.6.4.1. Modification du VI

Dans le cas d'une cinématique d'authentification à partir d'un opérateur de service, l'opérateur de service génère une requête d'authentification possédant un identifiant unique (cf. §3.5.1 p24).

Cet identifiant unique doit être repris dans la réponse retournée (c'est-à-dire le VI) par l'opérateur d'authentification :

- Dans l'attribut `InResponseTo` de l'élément `Response`
- Dans l'attribut `InResponseTo` de l'élément `SubjectConfirmationData`

Dans le cas d'une cinématique d'authentification initiée à partir de l'opérateur d'authentification, ces attributs `InResponseTo` ne seront pas positionnés.

3.6.4.2. Modification du RelayState

Dans Interops-P 1.0, le RelayState est utilisé pour désigner l'URL cible. Il ne doit pas être modifié dans le cadre d'Interops-S pour une cinématique de connexion initiée à partir de l'opérateur de service.

3.7. Déconnexion globale

3.7.1. Format des requêtes

Le format d'une requête de déconnexion est décrit ci-dessous. Il suit le format d'une requête de déconnexion SAML 2.0. Cette requête est transmise en utilisant le binding HTTP-Redirect.

Chaque mot écrit entre crochets en gras rouge (ex : **[ID]**) est une variable paramétrée dont la valeur est définie ci-dessous.

```
<samlp:LogoutRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="[ID]" Version="2.0"
    Destination="[Destination]"
    IssueInstant="[IssueInstant]">
    <saml:Issuer NameQualifier="[NameQualifier]"
SPNameQualifier="[SPNameQualifier]">[Issuer]</saml:Issuer>
    <saml:NameID Format="[SubjectFormat]">[SubjectId]</saml:NameID>
    <samlp:SessionIndex>[AssertionId]</samlp:SessionIndex>
</samlp:LogoutRequest>
```

Nom	Description	Format	Exemple
ID	Identifiant unique de la requête	Le format de l'identifiant doit suivre les recommandations de la RFC 4122 [RFC4122] afin d'en assurer l'unicité.	`uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
IssueInstant	Instant de génération de la requête	Valeur de type `xs:dateTime` (type de donnée de schéma XML) et doit être	`2003-04-17T00:46:02Z`

		exprimée en UTC, sans fuseau horaire.	
NameQualifier	Identifiant de l'opérateur d'authentification	Le format de l'identifiant est une URI. Le format de l'identifiant de l'opérateur d'authentification suit les recommandations du VI [R2].	`urn:interops:{SIREN|SIRET}:idp:{libre}{:version}`
SPNameQualifier	Identifiant de l'opérateur de service	Le format de l'identifiant est une URI. Le format de l'identifiant de l'opérateur de service suit les recommandations du VI [R2].	`urn:interops:{SIREN|SIRET}:sp:{libre}{:version}`
Issuer	Identifiant de l'émetteur de la requête	Le format de l'identifiant est une URI. Le format de l'identifiant de l'opérateur d'authentification ou de service suit les recommandations du VI [R2].	`urn:interops:{SIREN|SIRET}:{idp|sp}:{libre}{:version}`
Destination	URL identifiant l'adresse du service de déconnexion en charge du traitement de demandes	Cette URL correspond à l'URL du service de déconnexion. Le service de déconnexion aura la charge de vérifier ce paramètre et s'assurer que la requête lui est bien destinée.	`https://rniam.cnav.fr/slo/request`
SubjectFormat	Identifiant du format de l'identifiant du demandeur pour SAML 2.0	Le format de l'identifiant dépend de l'accord d'interopérabilité. Il doit être identique au format utilisé pour le VI.	`urn:oasis:names:tc:SAML:2.0:nameid-format:transient`
SubjectId	Identifiant de l'utilisateur	L'identifiant de l'utilisateur doit être identique à celui du VI	
AssertionId	Identifiant de l'assertion	L'identifiant de session est égal à l'identifiant de l'assertion, c'est-à-dire du VI	`uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`

3.7.2. Format des réponses

Le format d'une réponse de déconnexion est décrit ci-dessous. Il suit le format d'une réponse de déconnexion SAML 2.0. Cette réponse est transmise en utilisant le binding HTTP-Redirect.

Chaque mot écrit entre crochets en gras rouge (ex : **[ID]**) est une variable paramétrée dont la valeur est définie ci-dessous.

```
<samlp:LogoutResponse
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="[ID]" Version="2.0" IssueInstant="[IssueInstant]"
    Destination="[Destination]" InResponseTo="[InResponseTo]">
    <saml:Issuer NameQualifier="[NameQualifier]"
SPNameQualifier="[SPNameQualifier]">[Issuer]</saml:Issuer>
    <samlp:Status>
        <samlp:StatusCode Value="[Status]">
            <samlp:StatusCode Value="[Status2]"/>
        </samlp:StatusCode>
    <samlp:Status>
</samlp:LogoutResponse>
```

Nom	Description	Format	Exemple
ID	Identifiant unique de la réponse	Le format de l'identifiant doit suivre les recommandations de la RFC 4122 [RFC4122] afin d'en assurer l'unicité.	`uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6`
IssueInstant	Instant de génération de la réponse	Valeur de type `xs:dateTime` (type de donnée de schéma XML) et doit être exprimée en UTC, sans fuseau horaire.	`2003-04-17T00:46:02Z`

NameQualifier	Identifiant de l'opérateur d'authentification	Le format de l'identifiant est une URI. Le format de l'identifiant de l'opérateur d'authentification suit les recommandations du VI [R2].	`urn:interops:{SIREN|SIRET}:idp:{libre}{:version}`
SPNameQualifier	Identifiant de l'opérateur de service	Le format de l'identifiant est une URI. Le format de l'identifiant de l'opérateur de service suit les recommandations du VI [R2].	`urn:interops:{SIREN|SIRET}:sp:{libre}{:version}`
Issuer	Identifiant de l'émetteur de la réponse	Le format de l'identifiant est une URI. Le format de l'identifiant de l'opérateur d'authentification ou de service suit les recommandations du VI [R2].	`urn:interops:{SIREN|SIRET}:{idp|sp}:{libre}:version`
Destination	URL identifiant l'adresse du service de réception des réponses de déconnexion	Cette URL correspond à l'URL du service de déconnexion. Le service de déconnexion aura la charge de vérifier ce paramètre et s'assurer que la réponse lui est bien destinée.	`https://rniam.cnaf.fr/slo/response`
InResponseTo	Identifiant de la requête	Doit être égal à l'identifiant de la requête associée.	`uuid:f83d4fae-7dec-11d0-a765-00a0c91e34f6`
Status et **Status2**	Code de retour	En cas de succès la valeur de `Status` doit être `urn:oasis:names:tc:SAML:2.0:status:Success` et `Status2` doit être omis. En cas d'erreur, se reporter au paragraphe 3.7.3.	`urn:oasis:names:tc:SAML:2.0:status:Success`

3.7.3. Gestion des erreurs

En cas d'erreur, Seuls varient les champs **`[Status]`** et **`[Status2]`** de la réponse. L'élément <`StatusCode`> portant le champ **`[Status2]`** est optionnel.

`[Status]` peut prendre les valeurs :

- `urn:oasis:names:tc:SAML:2.0:status:Requester` : la requête comporte une erreur coté serveur
- `urn:oasis:names:tc:SAML:2.0:status:Responder` : le traitement de la requête a entraîné une erreur côté. Les sous-codes suivants peuvent être utilisés pour le champ **`[Status2]`** :
  - `urn:oasis:names:tc:SAML:2.0:status:PartialLogout` : l'utilisateur n'a pu être déconnecté

3.8. Récapitulatif des échanges en fonction des cinématiques

3.8.1. Echanges liés à l'authentification et à la déconnexion

Le tableau récapitulant les types de requêtes et de réponses intervenants pour l'authentification ou la déconnexion, au sens [SAMLCore2], est le suivant :

Cinématique	Type des requêtes / Binding utilisé	Type des réponses / Binding utilisé
Authentification	AuthnRequest / HTTP-Redirect	Response / HTTP-POST
Déconnexion	LogoutRequest / HTTP-Redirect	LogoutResponse / HTTP-Redirect

3.8.2. Echanges liés au service de découverte

Les cinématiques liées au service de découverte, permettant de modifier ou récupérer un opérateur d'authentification, s'appuient sur un mécanisme de redirection HTTP dont les paramètres sont passés dans les URL. Le format de ces échanges est décrit au §3.3.2 p21 et s'appuie notamment sur [SAMLDisco].

4. IMPACTS SUR LES TRACES

Les événements à tracer sont définis conventionnellement entre les opérateurs d'authentification et de service.

Les événements liés à la génération et vérification d'un VI doivent être tracés respectivement par l'opérateur d'authentification et l'opérateur de service conformément au format d'échange des traces [R4].

4.1. Opérateur d'authentification

Afin de tracer l'ensemble des éléments relatifs à la connexion de l'utilisateur, l'opérateur doit tracer a minima :

- L'authentification de l'utilisateur
- La génération du VI

La trace d'une authentification de l'application cliente ou de l'utilisateur final doit comporter les éléments suivants :

- Date de l'événement
- Identifiant local à l'opérateur d'authentification de l'utilisateur
- Méthode d'authentification
- Statut de l'authentification (succès et échec)

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

- Date et heure de l'événement
- Identifiant unique VI correspond à l'attribut ID de l'élément Assertion et l'identifiant de l'opérateur d'authentification
- Identifiant du service visé
- Statut de la génération (échec ou réussite)

4.2. Opérateur de service

Afin de tracer l'ensemble des éléments relatifs à la connexion de l'utilisateur, l'opérateur doit tracer à minima :

- La réception et vérification du VI
- La connexion à l'application cible

La trace de vérification du contexte applicatif doit comporter les éléments suivants :

- Date et heure de l'événement
- Identifiant du contexte correspondant à l'attribut ID de l'élément Assertion et l'identifiant de l'organisme émetteur
- Identifiant du service visé
- VI vérifié, contenant la signature
- Statut de la vérification (échec ou réussite) avec éventuellement un détail en cas d'échec

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

- Date de l'événement
- Identifiant local à l'opérateur de service de l'utilisateur
- URL de l'application cible à laquelle tente de se connecter l'utilisateur
- Statut de l'action réalisée (succès et échec)