prédice
MA SANTÉ
HAUTS•DE•FRANCE

Spécification d'interopérabilité

Single Sign-On (SSO)

ARCHITECTURE TECHNIQUE

Ce document fournit les informations nécessaires pour intégrer l'infrastructure d'authentification SSO de Maincare Solutions.

Référence	Emis le	Par	Visa
		Maincare	
Version	**Revu le**	**Par**	**Visa**
1.0		Maincare	
Qualification	**Approuvé le**	**Par**	**Visa**

Diffusion			
Organisme ou Entreprise	Destinataires	Nb Copies	Pour

A = Approbation C = Action I = Information D= Diffusion R = Revue V= Validation

Fiche de Mise à Jour		
Version	Date	Motifs - Détail des opérations

ars Agence Régionale de Santé Hauts-de-France

Sant& Numérique Hauts-de-France

l'Europe s'engage en Hauts-de-France avec le FEDER

Revue du document			
Validation interne			
Nom Prénom	Titre	Département	Validé
		Développement	
		Marketing	
		Services	
		CQL	
		Support	
		Architecture Sales	
Validation externe (groupe de travail)			
Nom Prénom	Titre	Centre Hospitalier	Validé

Date : 26/11/2019

SOMMAIRE

1. Généralités

1.1. Introduction

L'objectif du document est de décrire l'intégration des applications au serveur IdéoSSO. Les points abordés seront :

- Les concepts et les cas d'utilisations d'un serveur IdéoSSO
- L'intégration des clients CAS officiels compatibles avec IdéoSSO
- L'authentification déléguée au serveur IdéoSSO

1.2. Version

Ce document correspond aux versions **5.1.x d'IdéoSSO** et **4.5.x d'IdéoDirectory**.

1.3. Conventions typographiques

Un élément entouré des signes « plus petit » et « plus grand » représente un espace réservé à remplacer par une valeur spécifique. Exemple : 1.2.250.1.247.2.13.<FINESS> pourra représenter pour un établissement donné : 1.2.250.1.247.2.13.590050100.

1.4. Conventions sémantiques

Dans les tableaux de description de messages,

- « R » signifie « Requis » (valeur obligatoire)
- « O » signifie « Optionnel » (valeur possible mais pas obligatoire)
- « - » signifie « pas d'application » (la valeur ne doit pas être présente ; option seulement valable quand plusieurs cas différents sont présentés dans un tableau synthétique).

2. Authentification centralisée (Identity Provider)

IdéoSSO est un serveur d'authentification centralisée.

2.1. Mode d'authentification

Le mode d'authentification est le dispositif utilisé par l'utilisateur pour prouver son identité. On parle aussi de facteur d'authentification.

On distingue deux types d'authentification :

- L'authentification simple lorsque celle-ci ne repose que sur un seul facteur (exemple : l'utilisateur indique son mot de passe)
- L'authentification forte lorsque plusieurs facteurs différents sont combinés (par exemple, ce que je sais et ce que je possède : mot de passe saisi sur un terminal lui-même authentifié et enregistré comme appartenant à la personne authentifiée).

Les modes d'authentifications supportés sont décrits ci-après.

2.2. Authentification directe publique

2.2.1. Authentification forte par carte à puce (X509)

L'authentification par carte à puce utilise les mécanismes standards PKCS#11 des navigateurs Internet. L'authentification TLS mutuelle utilise les clés privées et les certificats de la carte à puce chargés dans les magasins du système d'exploitation. A l'exception de Firefox qui utilise ses propres périphériques PKCS#11 d'accès aux cartes à puce.

2.2.1.1 Fournisseurs de cartes à puces supportées

Les cartes à puces supportées sont :

- Les cartes CPS Françaises : http://esante.gouv.fr/services/espace-cps/
- Les cartes LuxTrust Luxembourgeoises : https://www.luxtrust.lu/
- Les cartes FMH Suisses : https://www.fmh.ch/fr/index.html
- Les cartes SuisseID Suisses : https://www.suisseid.ch/fr

2.2.1.2 Systèmes d'exploitation supportés

Les systèmes d'exploitation supportés par l'authentification X509 sont :

- Windows 8 64-bit
- Windows 8.1 32-bit et 64-bit
- Windows 10 32-bit et 64-bit
- macOS 10.12

Cette liste est valable sous condition que les fournisseurs de cartes à puce supportent aussi ces systèmes d'exploitation.

2.2.1.3 Navigateurs Internet supportés

Les navigateurs supportés par l'authentification X509 sont :

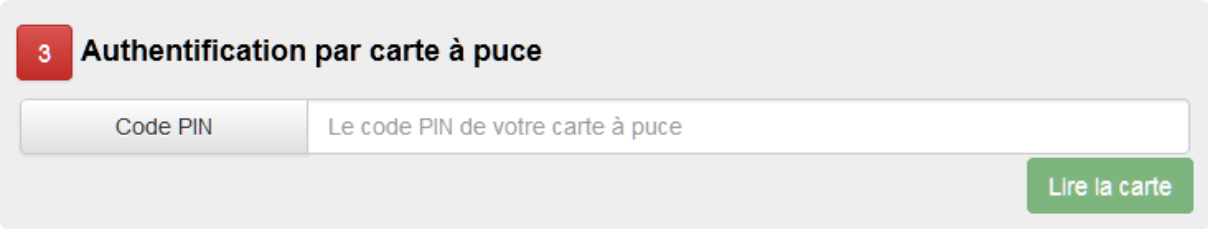
- Firefox ESR 52+

- Firefox 55+
- Windows Edge 40+
- Internet Explorer 11
- Google Chrome 60+

Cette liste est valable sous condition que les fournisseurs de cartes à puce supportent aussi ces navigateurs.

2.2.2. Authentification forte par carte à puce (LecteurCPS)

L'authentification par carte à puce utilise une applet Java pour accéder à la carte à puce. Suite à la politique de sécurité des navigateurs Internet, ce mode d'authentification est devenu obsolète. Ce mode d'authentification est remplacé par l'authentification X509 décrite précédemment.



2.2.3.1 Navigateurs Internet supportés

Les navigateurs supportés par l'authentification par carte à puce sont :

- Firefox ESR 52

2.2.3. Prérequis Java

L'installation de Java JRE 32-bit est obligatoire. Les versions de Java supportées sont :

- Java 8 Update 25 32-bit
- Java 7 Update 71 32-bit

2.3. Authentification directe privée

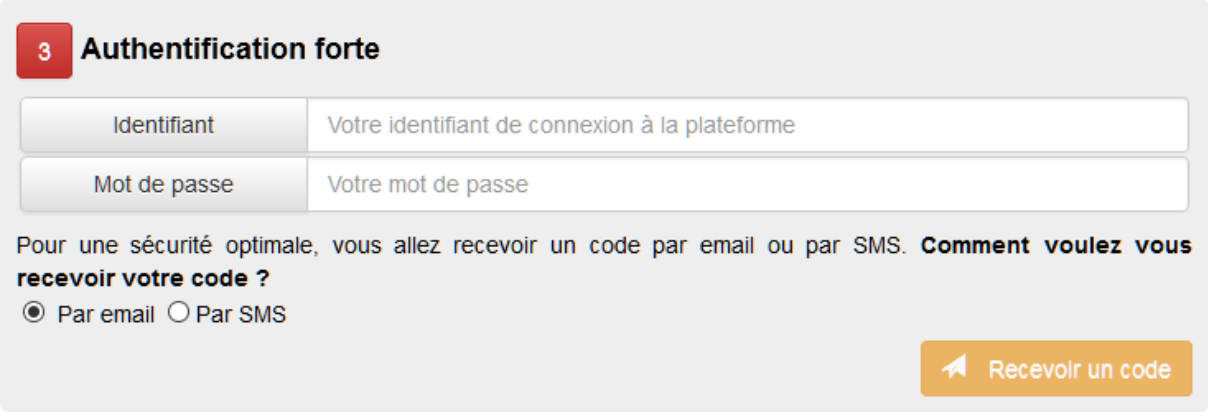
2.3.1. Authentification Login / Mot de passe (Classique)

L'authentification s'effectue en utilisant le login et le mot de passe de l'utilisateur.

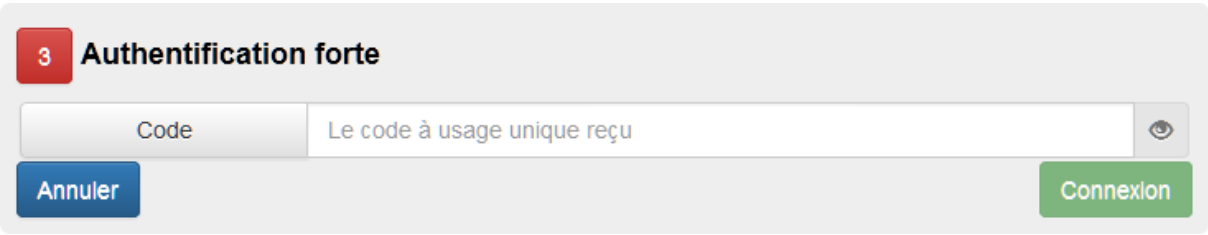
Suite à plusieurs erreurs d'authentification, le serveur IdéoSSO verrouille temporairement le compte utilisateur.

2.3.2. Authentification forte par mail (mailOttFort)

L'authentification s'effectue en utilisant le login et le mot de passe de l'utilisateur. Suite à cette première authentification, l'utilisateur reçoit un code sur son adresse email d'authentification.



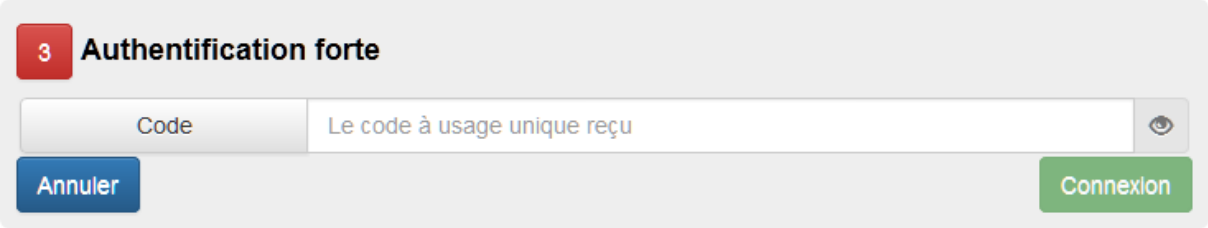
Par défaut, le code reçu est à usage unique et doit être utilisé avec le navigateur ayant fait la demande.



2.3.3. Authentification forte par téléphone (smsOttFort)

L'authentification s'effectue en utilisant le login et le mot de passe de l'utilisateur. Suite à cette première authentification, l'utilisateur reçoit un code sur son numéro de téléphone mobile d'authentification.

Par défaut, le code reçu est à usage unique et doit être utilisé avec le navigateur ayant fait la demande.



2.3.4. Vérification CAPTCHA

Suite à plusieurs erreurs d'authentification et avant le verrouillage d'un compte utilisateur, le serveur IdéoSSO propose à l'utilisateur la validation d'un CAPTCHA.

2.4. Niveau d'authentification

2.4.1. Localisation IP

Le réseau privé virtuel précise la provenance de l'utilisateur. Le serveur IdéoSSO détecte l'adresse IP du poste utilisé lors de l'authentification. Les contrôles d'accès aux applications peuvent être fonction de la localisation de l'utilisateur.

2.4.2. Palier

Le niveau d'authentification est la combinaison d'un mode d'authentification et d'une localisation IP. Un indice numérique (le palier) est associé au niveau d'authentification. Les valeurs prédéfinies des paliers sont :

Identifiantd u niveau	Nom complet	Mode d'authentification	Indice
login	Login/mot de passe	Classique	1
~~**103**~~	~~Authentification par mail~~	~~Obsolète~~	~~1~~
~~**104**~~	~~Authentification par SMS~~	~~Obsolète~~	~~1~~
101	Authentification forte par mail	mailOttFort	3
102	Authentification forte par SMS	smsOttFort	3
cps	Lecteur CPS	LecteurCPS	5
X509	Authentification X509	X509	5

Les indices numériques permettent de spécifier aux applications le palier obtenu lors de l'authentification.

2.5. Fournisseur d'identité

IdéoSSO est un fournisseur d'identité.

2.5.1. Protocole CAS 2.0 et 3.0

En tant que fournisseur d'identité, IdéoSSO supporte les protocoles CAS 2.0 et CAS 3.0. Une description est disponible à l'URL suivante : https://apereo.github.io/cas/5.0.x/protocol/CAS-Protocol.html

La documentation officielle est disponible sur le site de l'APEREO : https://www.apereo.org/projects/cas

2.5.2. Protocole SAML 1.1

En tant que fournisseur d'identité, IdéoSSO supporte le profile *Browser/Artifact Profile* du standard SAML 1.1. Une description est disponible à l'URL suivante : https://en.wikipedia.org/wiki/SAML_1.1#Browser.2FArtifact_Profile

La documentation officielle est disponible sur le site de l'OASIS : https://www.oasis-open.org/

2.5.3. Identification de l'utilisateur

L'authentification d'un utilisateur est le fait qu'il apporte la preuve de son identité et qu'il soit reconnu par le serveur IdéoSSO. Dans l'architecture mise en œuvre, c'est le serveur IdéoSSO qui permet l'authentification de l'utilisateur grâce aux différents modes d'authentifications proposés. Les clients IdéoSSO obtiennent l'identité de l'utilisateur en validant des jetons de service (ou ticket de service) auprès du serveur. Cette partie est l'identification de l'utilisateur. Ces jetons sont eux-mêmes distribués par le serveur IdéoSSO.

Depuis la version 2.0 d'IdéoDirectory, les comptes utilisateurs ne sont plus identifiés par leur login de connexion mais par un identifiant interne. Cet identifiant interne est une séquence technique gérée par l'application IdéoDirectory. Ce nouveau comportement est nécessaire afin de permettre la modification du login de connexion d'un compte utilisateur.

Suite à l'authentification de l'utilisateur, les applications clientes reçoivent donc l'identifiant interne du compte utilisateur. Elles peuvent récupérer les informations complémentaires sur l'utilisateur connecté en utilisant les webservices sécurisés publiés par IdéoDirectory. Ces webservices sont décrits dans un document annexe à cette documentation.

Depuis la version 3.0 d'IdéoSSO, en plus de l'identifiant interne du compte utilisateur, les protocoles CAS 3.0 et SAML 1.1 permettent, lors de la validation d'un jeton de service, d'obtenir les attributs supplémentaires sur l'authentification initiale. Ces attributs peuvent etre utilisés par l'application cliente pour identifier l'utilisateur.

2.5.4. Attributs d'authentification

Les protocoles CAS 3.0 et SAML 1.1 diffusent de l'information sur l'authentification initiale dès la validation du ticket de service. Les attributs suivants sont diffusés s'ils sont disponibles pour le compte utilisateur dans l'annuaire référentiel. Les attributs notés (*) sont toujours présents.

Nom de l'attribut	Description	Provenance	Exemple
username (*)	**Identifiant interne du compte utilisateur**	**Annuaire Ref.**	**000000101**
firstname (*)	Prénom de la personne	Annuaire Ref.	AGENT
lastname (*)	Nom de la personne	Annuaire Ref.	IDO-IN

Nom de l'attribut	Description	Provenance	Exemple
authMode (*)	Mode d'authentification IdéoSSO	Auth. SSO	Classique LecteurCPS X509 SAML2WebSSO …
authLevel (*)	Niveau d'authentification IdéoSSO	Auth. SSO	login ou cps ou …
remoteAddress	Adresse IP	Auth. SSO	90.83.182.213
casUserPrincipalTimeout	Durée de connexion maximale	Auth. SSO	
uid (*)	**Login d'authentification**	**Auth. SSO**	**aidoin**
CompteUtilisateur.uid (*)	Login du compte utilisateur	Annuaire Ref.	aidoin
CompteUtilisateur.mailLocaux	Adresses Email locales du compte utilisateur	Annuaire Ref.	Tableau d'email
CompteUtilisateur.mailPrincipal	Adresse Email principal du compte utilisateur	Annuaire Ref.	aidoin@ido-in.com
cpsNumber	Numéro de carte CPS d'authentification CPS	Auth. CPS	2200467925
idNat	Identifiant GIP-CPS d'authentification CPS	Auth. CPS	00B1038344
cpsAuthCert	Certification X509 d'authentification CPS	Auth. CPS	Auth. Cert. Base64
Personne.id (*)	**Identifiant interne de la personne**	**Annuaire Ref.**	**000000101**
Personne.nom (*)	Nom du professionnel	Annuaire Ref.	IDOIN
Personne.prenom (*)	Prénom de la personne	Annuaire Ref.	AGENT
Personne.mail	Adresse Email du professionnel	Annuaire Ref.	aidoin@ido-in.com
Personne.mailPerso	Adresse Email personnel de la personne	Annuaire Ref.	perso@ido-in.com
Personne.mailPrincipal	Adresse Email principal de la personne	Annuaire Ref.	aidoin@ido-in.com
Personne.mailSecondaire	Adresse Email de notification de la personne	Annuaire Ref.	notification@ido-in.com
Personne.idNat (*)	**Identifiant de la personne**	**Annuaire Ref.**	**00B1038344**
Personne.civilite	Code civilité de la personne	Annuaire Ref.	Code Civilité
Personne.numAdeli	Numéro ADELI principal de la personne	Annuaire Ref.	0B1038344
Personne.numCarteCps	Numéros de carte CPS de la personne	Annuaire Ref.	
Personne.titre	Code titre de la personne	Annuaire Ref.	Code Titre
Personne.historiqueIdNat	**Historique des identifiant de la personne**	**Annuaire Ref.**	**Tableau des anciens idNat**
Personne.historiqueAdeli	Historique des numéros ADELI de la personne	Annuaire Ref.	Tableau des anciens ADELI
Personne.numRPPS	Numéro RPPS de la personne	Annuaire Ref.	00000000000
NiveauAuthentification.nomComplet	Nom du niveau d'authentification	Auth. SSO	Login/Mot de passe ou Lecteur CPS
NiveauAuthentification.authNiveauIndice	Indice du niveau d'authentification	Auth. SSO	Valeur de 1 à 10 (cps = 5)

De manière générale, il est préférable que les applications clientes identifient l'utilisateur connecté à partir d'attributs pérennes. Certains attributs sont pérennes car ils sont techniques et internes à l'annuaire référentiel. C'est le cas des attributs ***username et Personne.id***.

Toutefois, l'utilisation de ces attributs techniques n'est pas appropriée en cas de reprise de données suite à un dysfonctionnement d'infrastructure. Il est donc préférable d'utiliser des attributs fonctionnellement pérennes. C'est le cas des attributs : ***Personne.idNat et Personne.historiqueIdNat***.

Pour rappel, comme indiqué précédemment, dans l'annuaire référentiel, l'attribut ***uid*** d'un compte utilisateur est modifiable. Il n'est donc pas pérenne.

2.5.5. Domaine d'authentification

Depuis la version 4.0 d'IdéoSSO, en plus des attributs d'identification, le serveur IdéoSSO diffuse le domaine d'authentification de l'utilisateur. Le domaine par défaut « default » correspond au domaine d'authentification des professionnels de santé.

Nom du domaine	Description	Provenance	Exemple
default	**Domaine d'authentification par défaut**	**Annuaire Ref.**	**default**
patient	**Domaine d'authentification des patients**	**Annuaire Ref.**	**patient**

Le domaine est présent dans les attributs diffusés :

Nom de l'attribut	Description	Provenance	Exemple
ideoDirectoryDomain (*)	**Domaine de l'authentification**	**Annuaire Ref.**	**default**
CompteUtilisateur.ideoDirectoryDomain (*)	Domaine du compte utilisateur	Annuaire Ref.	patient
Personne.ideoDirectoryDomain (*)	Domaine de la personne	Annuaire Ref.	patient

Si présent, le domaine d'authentification **doit obligatoirement être pris en compte** afin d'identifier l'utilisateur connecté. L'unicité des attributs ainsi que des identifiants nationaux n'est assurée que dans le domaine indiqué. Dans deux domaines distincts, deux utilisateurs peuvent avoir les mêmes identifiants et les mêmes attributs.

2.5.6. Informations complémentaires sur l'utilisateur

Depuis la version 2.0, l'application IdéoDirectory propose un ensemble de webservices permettant de récupérer des informations d'un utilisateur. L'application cliente peut utiliser ces webservices afin de récupérer les informations suivantes :

- Informations sur le professionnel de santé, ses activités et ses fonctions localisées
- Informations sur le compte utilisateur.
- Information sur les communautés de pratique, les groupes fonctionnels.
- Informations sur la politique de sécurité.
- Informations sur les groupes de sécurité, les profils applicatifs, les applications autorisées.

Les interfaces de ces webservices sont décrites dans un document annexe à cette documentation. L'ensemble de ces webservices sont sécurisés (WS-Security) et sont autorisés qu'aux applications ayant un compte applicatif défini dans l'annuaire.

2.5.7. Recommandations pour l'identification CAS 3.0

Une application cliente se connectant via le serveur IdéoSSO doit utiliser et enregistrer l'Identifiant National (IdNat) de la personne **et le domaine d'authentification** pour identifier le compte utilisateur connecté. Toutefois, l'idNat principal peut être amené à être modifié (la personne change de département par exemple), cet identifiant passera donc dans l'historique des identifiants nationaux. Cet idNat pourra

toujours être utilisé pour rechercher la personne avec les webservices d'IdéoDirectory. Les recherches sont réalisées sur l'idNat et l'historique idNat.

C'est pour cela que les applications clientes doivent aussi vérifier l'historique idNat fourni dans l'authentification si elles ne connaissent pas l'idNat principal de la personne connectée.

Si une personne change d'idNat principal, il est conseillé de le mettre à jour dans la base locale de l'application cliente lors de la phase d'identification.

2.5.8. Recommandations pour l'identification SAML 1.1

Les remarques du paragraphe précédent sont valables pour le protocole SAML 1.1.

2.5.9. Recommandations pour l'identification CAS 2.0

Une application externe, compatible CAS 2.0, se connectant via le serveur IdéoSSO n'utilise pas les attributs diffusés par le serveur. Seul l'attribut *username* est diffusé, c'est-à-dire l'identifiant interne du compte utilisateur dans le cas d'IdéoSSO. Comme indiqué précédemment, il est préférable utiliser l'identifiant national de la personne pour identifier l'utilisateur connecté.

Depuis la version 3.7.2 d'IdéoSSO, il est possible de configurer l'attribut pivot qui sera utilisé dans l'attribut *username* par le protocole CAS 2.0 en fonction de l'application. Cet attribut pivot doit être mono-valué.

Voici la liste possible :

Nom de l'attribut	Description	Provenance	Exemple
username (*)	**Identifiant interne du compte utilisateur**	**Annuaire Ref.**	**000000101**
uid (*)	**Login d'authentification**	**Auth. SSO**	**aidoin**
CompteUtilisateur.uid (*)	Login du compte utilisateur	Annuaire Ref.	aidoin
idNat	Identifiant GIP-CPS d'authentification CPS	Auth. CPS	00B1038344
Personne.id (*)	**Identifiant interne de la personne**	**Annuaire Ref.**	**000000101**
Personne.mailPrincipal	Adresse Email principal de la personne	Annuaire Ref.	aidoin@ido-in.com
Personne.mailSecondaire	Adresse Email de notification de la personne	Annuaire Ref.	notification@ido-in.com
Personne.idNat (*)	**Identifiant de la personne**	**Annuaire Ref.**	**00B1038344**
Personne.numAdeli	Numéro ADELI principal de la personne	Annuaire Ref.	0B1038344
Personne.numRPPS	Numéro RPPS de la personne	Annuaire Ref.	00000000000

2.5.10. Mandataire CAS 2.0 et 3.0

2.5.10.1 Identification de l'utilisateur pour un mandataire

Un mandataire (ou proxy) est une application cliente IdéoSSO qui peut elle-même effectuer une authentification auprès d'une autre application cliente IdéoSSO. C'est-à-dire qu'il est capable de proposer aux applications des jetons qui serviront à la validation auprès du serveur IdéoSSO.

Pour qu'une application soit mandataire, il faut que le serveur IdéoSSO lui délivre un jeton de proxy. Pour cela, il faudra configurer les applications afin qu'elle puisse recevoir ce jeton.

2.5.10.2 Utilisation du mode mandataire

La validation des jetons fournis par un mandataire retournera l'identité de l'utilisateur authentifié aux applications. Il faut bien sûr s'être connecté au mandataire avec le processus d'authentification IdéoSSO.

L'avantage de ce fonctionnement est qu'il n'y a pas de rejoue de formulaire, le mot de passe de l'utilisateur ne transite pas entre les applications.

2.5.11. Déconnexion de l'utilisateur

Lorsque l'utilisateur souhaite se déconnecter d'une application cliente, 2 cas de figures sont possibles :

- **Déconnexion totale :** La demande fait en sorte que l'on soit totalement déconnecté du serveur IdéoSSO. L'utilisateur devra alors s'authentifier à nouveau pour accéder aux autres applications clientes.
- **Déconnexion partielle :** On se déconnecte de l'application cliente seulement. C'est-à-dire que l'application cliente prend en compte le fait que le délai de validité est bien expiré, néanmoins si l'utilisateur souhaite se reconnecter à l'application, c'est le mécanisme IdéoSSO qui sera lancé car le navigateur Internet possède un TGC. Il n'aura plus besoin de s'authentifier à nouveau mais son environnement de travail dans l'application cliente sera vierge comme une première connexion.

Depuis la version 3.0 d'IdéoSSO, le serveur diffuse aux applications clientes la déconnexion du compte utilisateur.

La requête de déconnexion SAML est décrite à l'URL suivante :
https://apereo.github.io/cas/5.0.x/installation/Logout-Single-Signout.html

2.6. Intégration d'une application WEB PHP (CAS 3.0)

2.6.1. La documentation officielle

La documentation officielle du client CAS pour l'environnement PHP se trouve à l'URL suivante : https://wiki.jasig.org/display/CASC/phpCAS

2.6.2. Téléchargement

L'archive contenant le client CAS PHP se trouve à l'URL suivante : http://downloads.jasig.org/cas-clients/php/current/

Actuellement, la version stable est la version 1.3.5.

2.6.3. Système requis

L'environnement requis pour le bon fonctionnement du client CAS PHP est :

- CURL 7.5+ compilé avec le support SSL
- PHP 5.4+ ou PHP 4.2.2+ pour la version 1.1.x
- Apache 2.0.44+

2.6.4. Installation et configuration

L'installation est simple. Il suffit d'extraire les fichiers de l'archive et de copier le sous répertoire **source/CAS** dans un répertoire spécifique. Ce répertoire spécifique doit être dans le chemin de recherche PHP (**include_path** dans le fichier php.ini).

Ensuite, pour activer le client CAS PHP dans vos pages PHP, il faut ajouter quelques lignes de codes au début de vos pages.

2.6.5. Identification de l'utilisateur

Pour identifier un utilisateur connecté au serveur IdéoSSO, il faut ajouter les lignes suivantes dans vos pages PHP :

```
// import phpCAS lib
include_once('CAS/CAS.php');

// initialize phpCAS
phpCAS::client(CAS_VERSION_2_0,'homologation.ideosante.com',443,'/ideosso/');

// initialize serviceURL
phpCAS::setFixedServiceURL('http://monapplication/') ;

// force CAS authentication
phpCAS::forceAuthentication();
```

L'identifiant de l'utilisateur connecté peut être obtenu à partir de l'appel suivant :

```
if ( phpCAS::isAuthenticated() ) {
phpCAS::getUser()
}
```

2.6.6. Attributs d'authentification

Les informations sur l'authentification sont diffusées lors de la validation du ticket. Un exemple de flux XML est présenté ci-dessous :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:authenticationSuccess>
<cas:user>101</cas:user>
<cas:attributes>
<cas:uid>aidoin</cas:uid>
<cas:Personne.idNat>3000000000/0000</cas:Personne.idNat>
<cas:lastname>IDOIN</cas:lastname>
<cas:firstname>AGENT</cas:firstname>
<cas:Personne.id>101</cas:Personne.id>
<cas:username>101</cas:username>
<cas:remoteAddress>/90.83.182.213</cas:remoteAddress>
<cas:authLevel>login</cas:authLevel>
<cas:Personne.prenom>AGENT</cas:Personne.prenom>
<cas:Personne.nom>IDOIN</cas:Personne.nom>
<cas:Personne.historiqueIdNat>P000000000</cas:Personne.historiqueIdNat>
<cas:authMode>Classique</cas:authMode>
<cas:CompteUtilisateur.uid>aidoin</cas:CompteUtilisateur.uid>
...
</cas:attributes>
</cas:authenticationSuccess>
</cas:serviceResponse>
```

Les attributs disponibles sont décrits dans le paragraphe 2.5.4.

2.6.7. Informations complémentaires sur l'utilisateur

L'application cliente devra utiliser les mécanismes standards PHP pour faire appel au webservice « IdeoDirectoryManagerSecure » sécurisé par la norme WS-Security.

Le fichier WSDL de description est disponible à l'URL suivante :
https://www.mondomaine.fr/ideodirectory.services/IdeoDirectoryManagerSecure?wsdl

L'accès au webservice est sécurisé par une authentification WS-Security à l'URL suivante :
https://www.mondomaine.fr/idéoDirectory.services/IdeoDirectoryManagerSecure

2.6.8. Déconnexion de l'utilisateur

Pour déconnecter totalement l'utilisateur, il faut rediriger l'utilisateur vers l'URL du service de déconnexion du serveur IdéoSSO. On utilise les lignes de codes suivantes :

```
<a href=" https://.../ideosso/logout?redirect=http://www.google.fr">Logout</a>
```

Le serveur IdéoSSO détruit alors le TGT pour le navigateur Internet. Le paramètre **redirect** permet de rediriger le navigateur suite à la déconnexion.

2.7. Intégration pour une application WEB J2EE (CAS 3.0)

2.7.1. La documentation officielle

La documentation officielle sur l'intégration du client CAS J2EE se trouve à l'URL suivante :
https://github.com/apereo/java-cas-client

2.7.2. Téléchargement

L'archive contenant le client CAS J2EE se trouve à l'URL suivante :
http://repo2.maven.org/maven2/org/jasig/cas/client/

2.7.3. Système requis

La mise en œuvre d'un client CAS J2EE est simple. Ce client utilise le système de filtres défini dans les spécifications de l'API Java Servlet 2.3 (inclus dans J2EE 1.3). Il est donc très facilement intégrable à une application J2EE.

2.7.4. Installation

Pour installer ce client dans une application J2EE, il suffit d'ajouter le jar au **CLASSPATH** de l'application. Pour cela, il faut le copier dans le répertoire **WEB-INF/lib** de l'application Web J2EE. Il est possible d'ajouter ces jars dans le fichier **META-INF/application.xml** d'un EAR si les applications sont déployées au sein d'un EAR.

C'est la seule installation à faire sur une application J2EE pour mettre en place le client CAS J2EE.

2.7.5. Configuration

La documentation officielle se trouve à l'URL suivante : https://github.com/apereo/java-cas-client#configuration

2.7.6. Identification de l'utilisateur

Après l'exécution du client CAS J2EE et après la validation du ticket auprès du serveur IdéoSSO, les attributs suivants se trouvent en session :

Nom de l'attribut	*Description*	*Classe : Exemple*
_const_cas_assertion_	L'objet assertion indiquant que l'identification de l'utilisateur s'est bien déroulée.	**org.jasig.cas.client.validation.Assertion.**
Principal Name	L'identifiant de l'utilisateur connecté.	**String : 0000142**

L'application peut donc récupérer l'identifiant de l'utilisateur connecté avec le code suivant :

```
HttpSession session = request.getSession();
Assertion assertion = (Assertion) session.getAttribute("_const_cas_assertion_");
String identifiant = assertion.getPrincipal().getName()
```

Si le filtre **org.jasig.cas.client.util.HttpServletRequestWrapperFilter.HttpServletRequestWrapperFilter** est utilisé, alors, il est possible de récupérer directement l'identifiant de connexion à partir de la request.

```
Principal principal = request.getUserPrincipal();
String identifiant = principal.getName();
```

Ce code de récupération d'identifiant peut être intégré dans la première servlet appelée de l'application. Ensuite, l'application pourra authentifier le compte utilisateur en local grâce à cette identifiant.

2.7.7. Attributs d'authentification

Les informations sur l'authentification IdéoSSO sont disponibles dans l'assertion. L'application peut donc les récupérer avec le code suivant :

```
Assertion assertion = (Assertion) session.getAttribute("_const_cas_assertion_");
Map informations = assertion.getPrincipal().getAttributes();
```

2.7.8. Informations complémentaires sur l'utilisateur

L'application cliente devra utiliser les mécanismes standards J2EE pour faire appel au webservice « IdeoDirectoryManagerSecure » sécurisé par la norme WS-Security.

Le fichier WSDL de description est disponible à l'URL suivante :

```
https://www.mondomaine.fr/ideodirectory.services/IdeoDirectoryManagerSecure?wsdl
```

L'accès au webservice est sécurisé par une authentification WS-Security à l'URL suivante :
https://www.mondomaine.fr/ideodirectory.services/IdeoDirectoryManagerSecure

2.7.9. Déconnexion de l'utilisateur

Pour déconnecter du serveur IdéoSSO, il suffit de faire une redirection du navigateur vers l'URL suivante :
https://.../ideosso/logout?redirect=http://www.google.fr

Le serveur IdéoSSO détruit alors le TGT pour le navigateur Internet. Le paramètre **redirect** permet de rediriger le navigateur suite à la déconnexion.

3. Authentification déléguée (Service Provider)

IdéoSSO est aussi un fournisseur de service. Le serveur peut déléguer l'authentification à un autre fournisseur d'identité (Identity Provider). Il supporte les protocoles et standards présentés ci-après.

3.1. Ouverture contextuelle

3.1.1. Ouverture contextuelle sollicitée (Web)

L'ouverture contextuelle sollicitée est une authentification déléguée et passive de l'utilisateur au serveur IdéoSSO ayant pour objectif d'ouvrir une application cible de façon transparente.

On parle d'ouverture contextuelle **sollicitée** car c'est IdéoSSO qui, suite au déclenchement via l'URL, sollicite l'IDP pour une authentification. Cette mise en œuvre (aller-retour) nécessite un environnement Web, c'est-à-dire un navigateur Internet.

Elle est déclenchée par l'URL suivante :
https://www.domaine.fr/ideosso/login?needs_client_redirection=true&client_name=<identifiant_niveau>&service=<service_url>&redirect=<redirect_url>

Les paramètres de l'URL sont :

Caractéristique	Valeur	Exemple
Needs_client_redirection	True. Ce paramètre est obligatoire.	true
<Service_URL>	Identifiant de l'application cible dans l'annuaire IdéoDirectory. Ce paramètre doit être URL encodé.	https://www.domaine.fr/application/
<Redirect_URL>	URL de redirection suite à l'authentification passive réussie. Ce paramètre doit être URL encodé.	https://www.domaine.fr/application/OuvertureContextuelle?param1=value1&param2=value2
<Identifiant_Niveau>	Identifiant du niveau ADFS	saml2_adfs, saml2_xxx, ideosso_env1

Exemple d'URL déclenchant l'authentification déléguée ADFS pour l'ouverture contextuelle de l'application cible :

https://www.domaine.fr/ideosso/login?needs_client_redirection=true&client_name=saml2_adfs&service=https%3A%2F%2Fwww.domaine.fr%2Fapplication%2F&redirect=https%3A%2F%2Fwww.domaine.fr%2Fapplication%2FOuvertureContextuelle%3Fparam1%3Dvalue1%26param2%3Dvalue2

L'ouverture contextuelle **est** supportée par tous les protocoles et standards présentés ci-après.

3.1.2. Ouverture contextuelle non sollicitée (RelayState)

L'ouverture contextuelle non sollicitée est une authentification déléguée et passive de l'utilisateur au serveur IdéoSSO ayant pour objectif d'ouvrir une application cible.

On parle d'ouverture contextuelle **non sollicitée** car c'est l'IDP qui sollicite directement IdéoSSO pour une authentification.

L'ouverture contextuelle non sollicitée **n'est pas** supportée par tous les protocoles. Elle est spécifiée dans les protocoles et standards présentés ci-après.

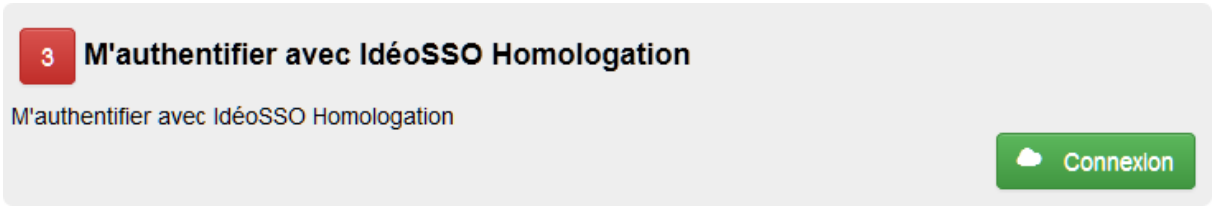
3.2. Authentification déléguée IdéoSSO (IdeoSSODelegate)

Le serveur IdéoSSO peut déléguer l'authentification à un autre serveur IdéoSSO. Les protocoles utilisés sont CAS 2.0 ou 3.0. https://apereo.github.io/cas/5.0.x/protocol/CAS-Protocol.html

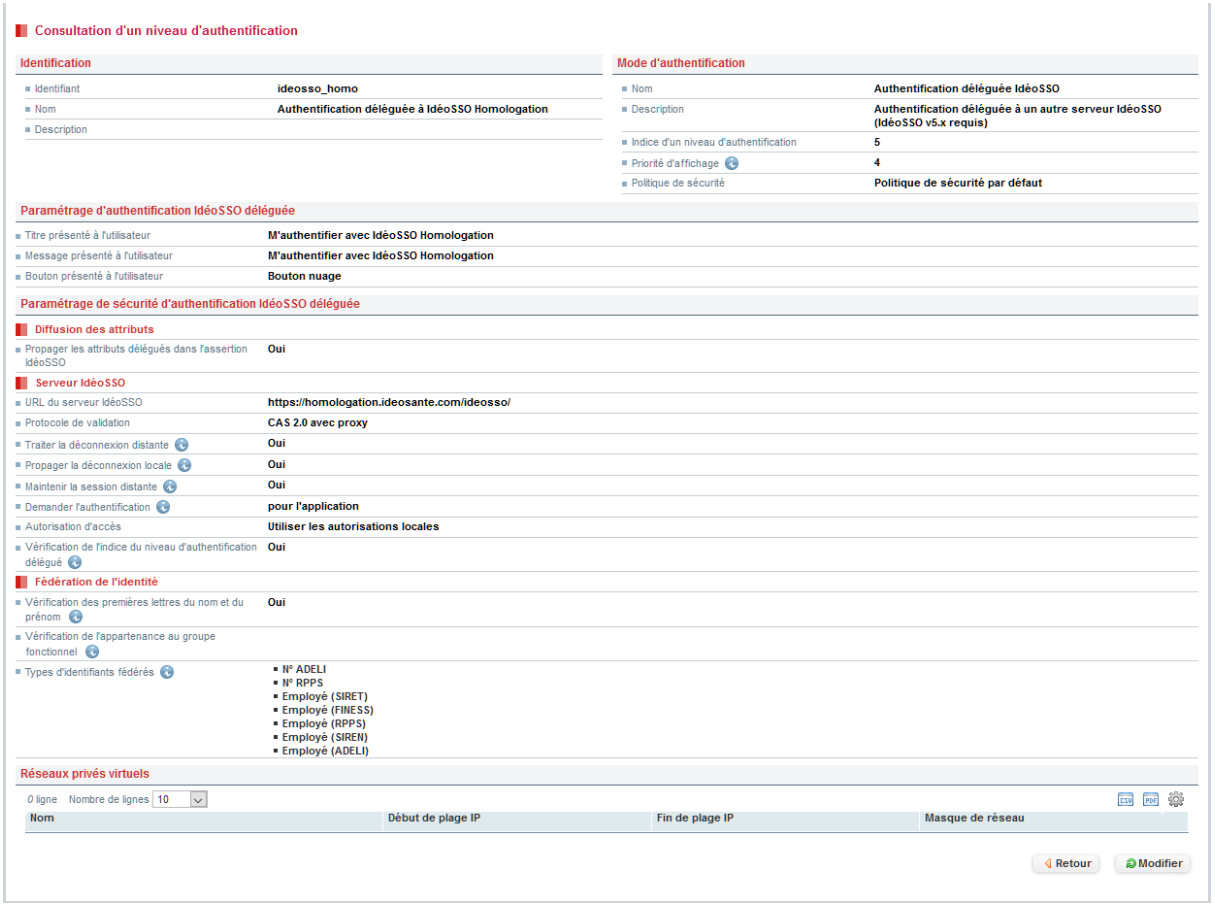
3.2.1. Cas d'utilisation

Cette authentification déléguée peut être mis en œuvre afin de réutiliser l'authentification locale du système d'information hospitalier d'un établissement vers un espace numérique régional de santé ou vers le système d'information d'un groupement hospitalier de territoire.

3.2.2. Mire d'authentification



3.2.3. Paramètres de l'authentification IdéoSSO



Les niveaux d'authentification sont identifiés par leur identifiant. Dans l'exemple ci-dessus, l'identifiant du niveau est « ideosso_homo ».

3.2.4. Fédération de l'identité

La fédération des identités est effectuée à partir des identifiants nationaux du compte utilisateur. L'identifiant national doit être véhiculé dans les attributs de l'assertion CAS 3.0 ou SAML 1.1.

Les attributs utilisés pour la recherche d'identité sont :

Nom de l'attribut	Description	Provenance	Exemple
firstname (*)	Prénom de la personne	Annuaire Ref.	AGENT
lastname (*)	Nom de la personne	Annuaire Ref.	IDO-IN
Personne.idNat (*)	**Identifiant de la personne**	**Annuaire Ref.**	**00B1038344**
Personne.historiqueIdNat	**Historique des identifiant de la personne**	**Annuaire Ref.**	**Tableau des anciens idNat**
NiveauAuthentification.authNiveauIndice	Indice du niveau d'authentification	Auth. SSO	Valeur de 1 à 10 (cps = 5)

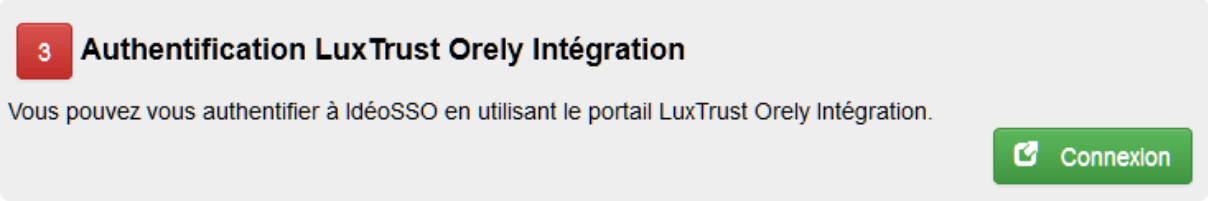
3.3. Authentification déléguée Portail Orely LuxTrust (LuxTrustOrely)

Le serveur IdéoSSO peut déléguer l'authentification au Portail Orely LuxTrust. https://www.luxtrust.lu/

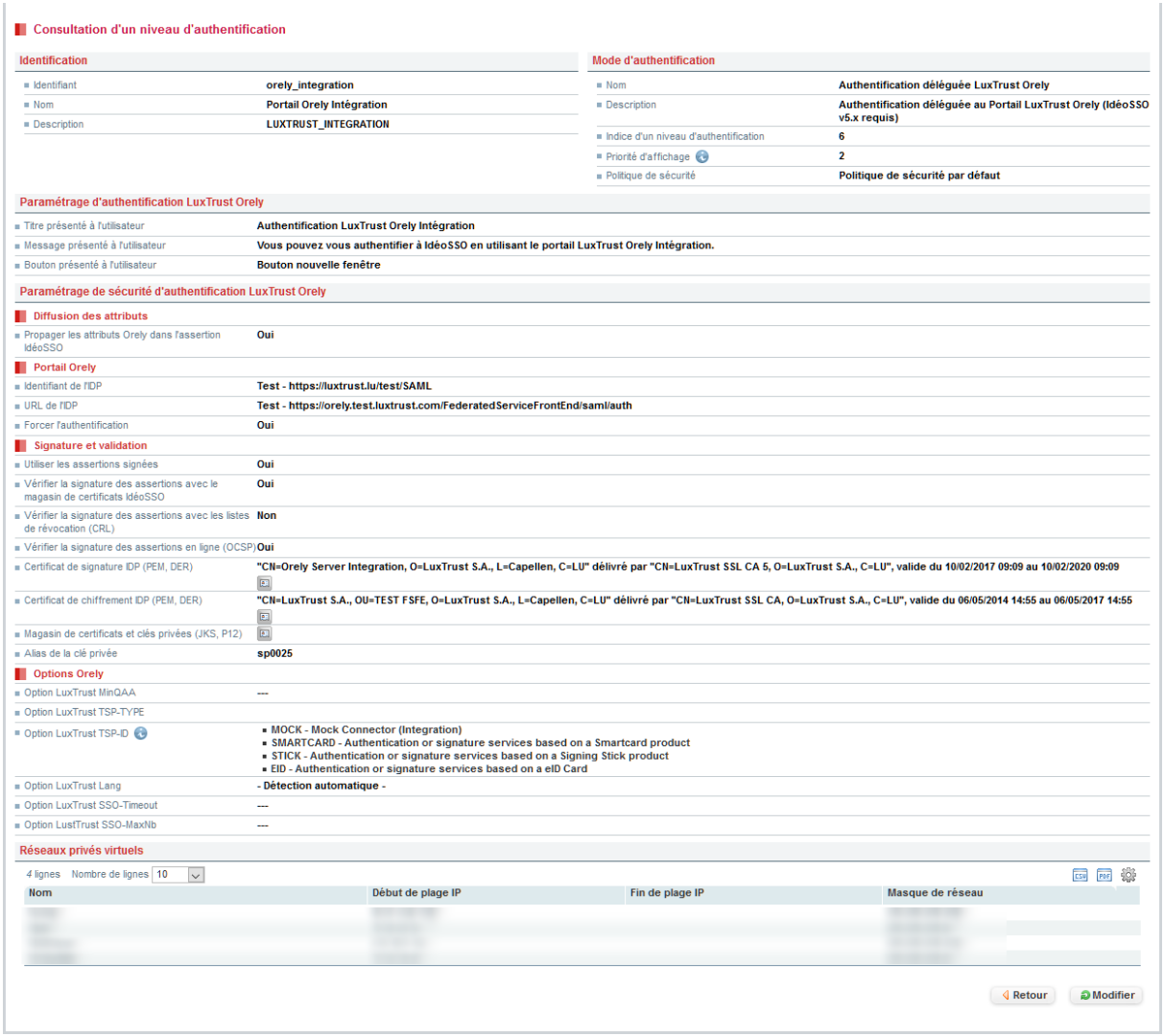
3.3.1. Cas d'utilisation

Cette authentification déléguée est mise en œuvre afin d'utiliser l'authentification nationale du Portail Orely au Luxembourg.

3.3.2. Mire d'authentification



3.3.3. Paramètres de l'authentification du Portail Orely



Les niveaux d'authentification sont identifiés par leur identifiant. Dans l'exemple ci-dessus, l'identifiant du niveau est « orely_integration ».

3.3.4. Fédération de l'identité

La fédération des identités est effectuée à partir des informations extraites du certificat LuxTrust fourni par le Portail Orely.

3.4. Authentification déléguée SAML 2.0 (SAML2WebSSO)

Le serveur IdéoSSO peut déléguer l'authentification à un fournisseur d'identité SAML 2.0 (Identity Provider). Le profil mis en œuvre est *SAML 2.0 Web Browser SSO Profile*. Une description du profil est disponible à l'URL suivante : https://en.wikipedia.org/wiki/SAML_2.0#SP_POST_Request.3B_IdP_POST_Response

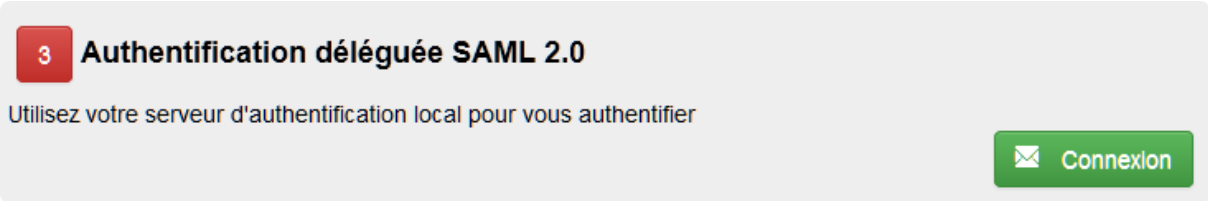
La documentation officielle est disponible sur le site de l'OASIS. https://www.oasis-open.org/committees/security/

Dans ce cas, le serveur IdéoSSO a le rôle de fournisseur de service (Service Provider).

3.4.1. Cas d'utilisation

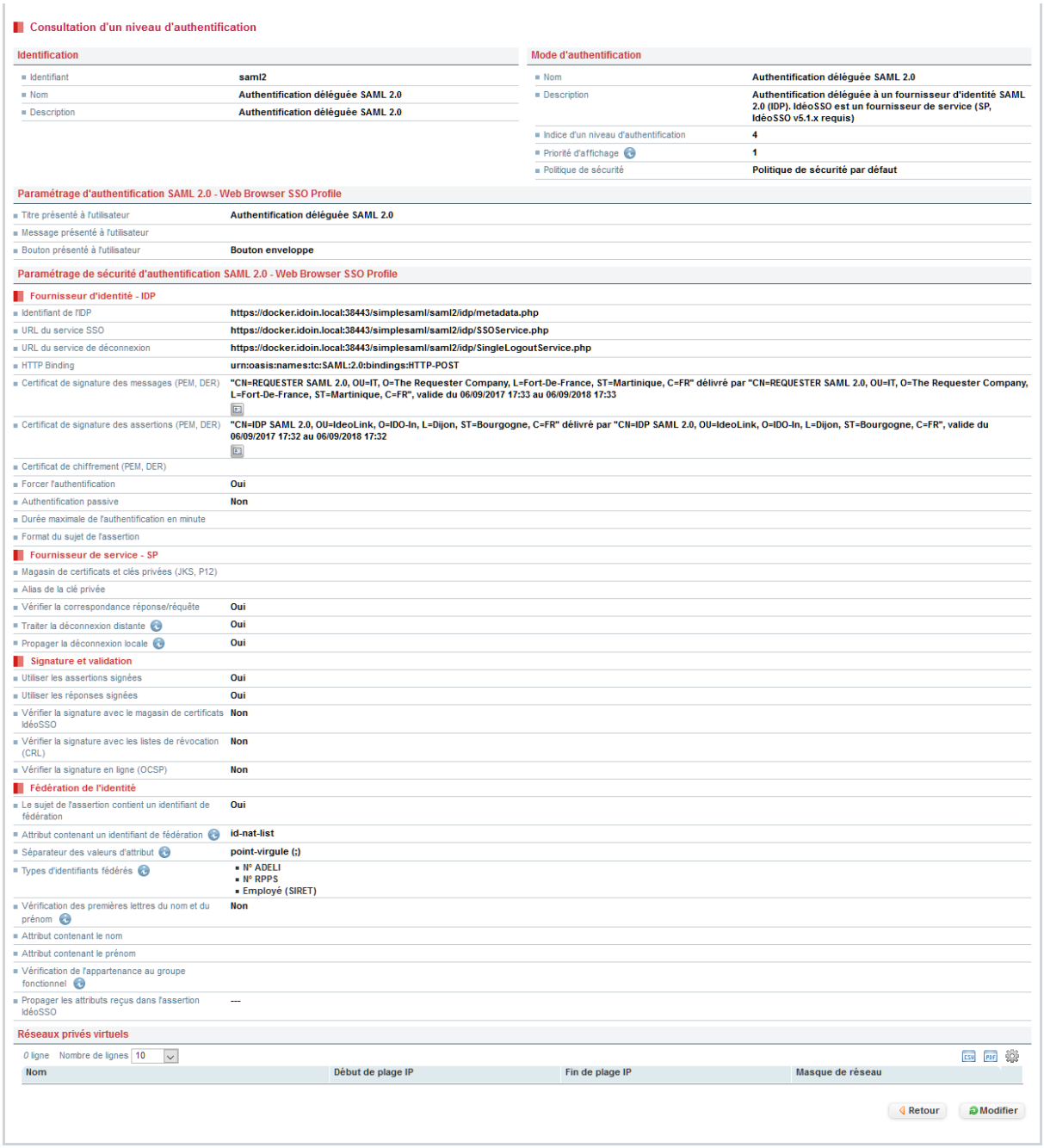
Cette authentification déléguée peut être mis en œuvre pour fédérer plusieurs systèmes d'information.

3.4.2. Mire d'authentification



3.4.3. Paramètres de l'authentification SAML 2.0

Les caractéristiques de l'authentification SAML 2.0 sont paramétrables dans le niveau d'authentification associé.



Consultation d'un niveau d'authentification

Identification

Identifiant	saml2
Nom	Authentification déléguée SAML 2.0
Description	Authentification déléguée SAML 2.0

Mode d'authentification

Nom	Authentification déléguée SAML 2.0
Description	Authentification déléguée à un fournisseur d'identité SAML 2.0 (IDP). IdéoSSO est un fournisseur de service (SP, IdéoSSO v5.1.x requis)
Indice d'un niveau d'authentification	4
Priorité d'affichage	1
Politique de sécurité	Politique de sécurité par défaut

Paramétrage d'authentification SAML 2.0 - Web Browser SSO Profile

Titre présenté à l'utilisateur	Authentification déléguée SAML 2.0
Message présenté à l'utilisateur	
Bouton présenté à l'utilisateur	Bouton enveloppe

Paramétrage de sécurité d'authentification SAML 2.0 - Web Browser SSO Profile

Fournisseur d'identité - IDP

Identifiant de l'IDP	https://docker.idoin.local:38443/simplesaml/saml2/idp/metadata.php
URL du service SSO	https://docker.idoin.local:38443/simplesaml/saml2/idp/SSOService.php
URL du service de déconnexion	https://docker.idoin.local:38443/simplesaml/saml2/idp/SingleLogoutService.php
HTTP Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
Certificat de signature des messages (PEM, DER)	"CN=REQUESTER SAML 2.0, OU=IT, O=The Requester Company, L=Fort-De-France, ST=Martinique, C=FR" délivré par "CN=REQUESTER SAML 2.0, OU=IT, O=The Requester Company, L=Fort-De-France, ST=Martinique, C=FR", valide du 06/09/2017 17:33 au 06/09/2018 17:33
Certificat de signature des assertions (PEM, DER)	"CN=IDP SAML 2.0, OU=IdeoLink, O=IDO-In, L=Dijon, ST=Bourgogne, C=FR" délivré par "CN=IDP SAML 2.0, OU=IdeoLink, O=IDO-In, L=Dijon, ST=Bourgogne, C=FR", valide du 06/09/2017 17:32 au 06/09/2018 17:32
Certificat de chiffrement (PEM, DER)	
Forcer l'authentification	Oui
Authentification passive	Non
Durée maximale de l'authentification en minute	
Format du sujet de l'assertion	

Fournisseur de service - SP

Magasin de certificats et clés privées (JKS, P12)	
Alias de la clé privée	
Vérifier la correspondance réponse/réquête	Oui
Traiter la déconnexion distante	Oui
Propager la déconnexion locale	Oui

Signature et validation

Utiliser les assertions signées	Oui
Utiliser les réponses signées	Oui
Vérifier la signature avec le magasin de certificats IdéoSSO	Non
Vérifier la signature avec les listes de révocation (CRL)	Non
Vérifier la signature en ligne (OCSP)	Non

Fédération de l'identité

Le sujet de l'assertion contient un identifiant de fédération	Oui
Attribut contenant un identifiant de fédération	id-nat-list
Séparateur des valeurs d'attribut	point-virgule (;)
Types d'identifiants fédérés	▪ N° ADELI ▪ N° RPPS ▪ Employé (SIRET)
Vérification des premières lettres du nom et du prénom	Non
Attribut contenant le nom	
Attribut contenant le prénom	
Vérification de l'appartenance au groupe fonctionnel	
Propager les attributs reçus dans l'assertion IdéoSSO	---

Réseaux privés virtuels

0 ligne Nombre de lignes 10

Nom	Début de plage IP	Fin de plage IP	Masque de réseau

Retour Modifier

Les niveaux d'authentification sont identifiés par un identifiant. Dans l'exemple ci-dessus, l'identifiant du niveau est : « saml2 ».

3.4.4. Certificat serveur X509

Afin de garantir la non-répudiation et la non-interception des messages, le standard SAML 2.0 met en œuvre le chiffrement et la signature électronique. Le fournisseur d'identité doit être en possession un ou plusieurs certificats serveurs afin d'assurer la sécurité des échanges avec le serveur IdéoSSO.

3.4.5. Fournisseur d'identité (IDP)

Les caractéristiques attendues du fournisseur d'identité SAML 2.0 sont :

3.4.5.1 Identity provider

Caractéristique	Valeur	Exemple
Protocole	SAML 2.0	urn:oasis:names:tc:SAML:2.0:protocol
Identifiant	Chaine de caractères	https://idp.domaine.fr/...

3.4.5.2 Service SSO

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL de destination	URL sécurisée HTTPS	https://.../ssoService
AuthNRequest signée	Optionnel	
Authentification passive	Non	
Forcer l'authentification	Oui	
Response signée	Oui	
Assertion signée	Oui	
Assertion chiffrée	Optionnel	
Format du sujet	Non spécifié	urn:oasis:names:tc:SAML:2.0:nameidformat:unspecified

3.4.5.3 Service de déconnexion

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL de destination	URL sécurisée HTTPS	https://.../logoutService
LogoutRequest signée	Optionnel	
LogoutResponse signée	Non	

3.4.5.4 Signature et chiffrement

Caractéristique	Valeur	Exemple
Algorithme de Hachage	SHA1, SHA256	http://www.w3.org/2000/09/xmldsig#sha1 http://www.w3.org/2001/04/xmlenc#sha256
Algorithme de signature	RSA-SHA1, RSA-SHA256	http://www.w3.org/2000/09/xmldsig#rsa-sha1 http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Algorithme de chiffrement	AES256-CBC	http://www.w3.org/2001/04/xmlenc#aes256-cbc

Suite au paramétrage du niveau d'authentification associé, les métadonnées attendues de l'IDP sont disponibles à l'URL suivante :
https//www.domaine.fr/ideosso/saml2/sp/metadata/idp?client_name=<identifiant_niveau>
Ci-dessous, un exemple de métadonnées SAML 2.0 attendues pour l'IDP traitant l'authentification.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#" xmlns:mdalg="urn:oasis:names:tc:SAML:metadata:algsupport"
xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://docs.oasis-
open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd">
 <EntityDescriptor entityID="https://idp-exemple.domaine.fr:38443/simplesaml/saml2/idp/metadata.php">
  <Extensions>
   <mdalg:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
   <mdalg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
   <mdalg:SigningMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
   <mdalg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
   <mdalg:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
  </Extensions>
  <IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
   <Extensions>
    <mdui:UIInfo>
     <mdui:DisplayName>Authentification déléguée SAML 2.0</mdui:DisplayName>
     <mdui:Description>Authentification déléguée SAML 2.0</mdui:Description>
    </mdui:UIInfo>
   </Extensions>
   <KeyDescriptor use="signing">
    <ds:KeyInfo>
     <ds:X509Data>
      <ds:X509Certificate>LS0tLS1CRUdJ...BVEUtLS0tLQo=</ds:X509Certificate>
     </ds:X509Data>
    </ds:KeyInfo>
   </KeyDescriptor>
   <KeyDescriptor use="signing">
    <ds:KeyInfo>
     <ds:X509Data>
      <ds:X509Certificate>LS0tLS1CRUdJT...UtLS0tLQo=</ds:X509Certificate>
     </ds:X509Data>
    </ds:KeyInfo>
   </KeyDescriptor>
   <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://idp-
exemple.domaine.fr:38443/simplesaml/saml2/idp/SingleLogoutService.php" />
   <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameidformat:unspecified</NameIDFormat>
   <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" Location="https://idp-
exemple.domaine.fr:38443/simplesaml/saml2/idp/SSOService.php" />
  </IDPSSODescriptor>
 </EntityDescriptor>
</EntitiesDescriptor>
```

3.4.6. Fournisseur de service (SP)

Le serveur IdéoSSO a le rôle de fournisseur de service SAML 2.0. Les caractéristiques du serveur IdéoSSO en tant que fournisseur de service SAML 2.0 sont :

3.4.6.1 Service Provider

Caractéristique	Valeur	Exemple
Protocole	SAML 2.0	urn:oasis:names:tc:SAML:2.0:protocol

Caractéristique	Valeur	Exemple
Identifiant	Chaine de caractères	https://www.domaine.fr/ideosso/ saml2/sp/metadata?client_name=<identifiant_niveau>

3.4.6.2 Service de traitement des assertions

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL de destination	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/ login?client_name=<identifiant_niveau>
Response signée	Oui	
Assertion signée	Oui	
Assertion chiffrée	Optionnel	
Format du sujet	Non spécifié	urn:oasis:names:tc:SAML:2.0:nameidformat:unspecified

3.4.6.3 Service de traitement de la déconnexion

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL d'écoute	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/ saml2/sp/slo/browser?client_name=<identifiant_niveau>
URL de retour	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/ saml2/sp/slo/response?client_name=<identifiant_niveau>
LogoutRequest signée	Non	
LogoutResponse signée	Non	

3.4.6.4 Signature et chiffrement

Caractéristique	Valeur	Exemple
Algorithme de Hachage	SHA1, SHA256	http://www.w3.org/2000/09/xmldsig#sha1 http://www.w3.org/2001/04/xmlenc#sha256
Algorithme de signature	RSA-SHA1, RSA-SHA256	http://www.w3.org/2000/09/xmldsig#rsa-sha1 http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Algorithme de chiffrement	AES256-CBC	http://www.w3.org/2001/04/xmlenc#aes256-cbc

Suite au paramétrage du niveau d'authentification associé, les métadonnées attendues de l'ADFS sont disponibles à l'URL suivante :
https//www.domaine.fr/ideosso/saml2/sp/metadata/idp?client_name=<identifiant_niveau>

Ci-dessous, un exemple de métadonnées SAML 2.0 attendues pour l'ADFS traitant l'authentification.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:mdalg="urn:oasis:names:tc:SAML:metadata:algsupport"
        xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
        xsi:schemaLocation="http://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd">
        <EntityDescriptor entityID="http://adfs-exemple.domaine.fr/adfs/services/trust">
                <Extensions>
```

```
                        <mdalg:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                        <mdalg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                        <mdalg:SigningMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
                        <mdalg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256" />
                        <mdalg:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"
/>
                </Extensions>
                <IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
                        <Extensions>
                                <mdui:UIInfo>
                                        <mdui:DisplayName>Mon compte Pro</mdui:DisplayName>
                                        <mdui:Description>Mon compte Pro</mdui:Description>
                                </mdui:UIInfo>
                        </Extensions>
                        <KeyDescriptor use="signing">
                                <ds:KeyInfo>
                                        <ds:X509Data>

        <ds:X509Certificate>LS0tLS1CRUdJTiB...ElGSUNBVEUtLS0tLQo=</ds:X509Certificate
                                        </ds:X509Data>
                                </ds:KeyInfo>
                        </KeyDescriptor>
                        <KeyDescriptor use="signing">
                                <ds:KeyInfo>
                                        <ds:X509Data>

        <ds:X509Certificate>LS0tLS1CRUdJTiB...ElGSUNBVEUtLS0tLQo=</ds:X509Certificate
                                        </ds:X509Data>
                                </ds:KeyInfo>
                        </KeyDescriptor>
                        <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://adfs-exemple.domaine.fr/adfs/ls/" />
                        <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</NameIDFormat>
                        <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://adfs-exemple.domaine.fr/adfs/ls/" />
                </IDPSSODescriptor>
        </EntityDescriptor>
</EntitiesDescriptor>
```

3.4.7. Fournisseur de service (SP)

Le serveur IdéoSSO a le rôle de fournisseur de service SAML 2.0. Les caractéristiques du serveur IdéoSSO en tant que fournisseur de service SAML 2.0 sont :

3.4.7.1 Service Provider

Caractéristique	Valeur	Exemple
Protocole	SAML 2.0	urn:oasis:names:tc:SAML:2.0:protocol
Identifiant	Chaine de caractères	https://www.domaine.fr/ideosso/saml2/sp/<identifiant_niveau>

3.4.7.2 Service de traitement des assertions

Caractéristique	Valeur	Exemple

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL de destination	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/login?client_name=<identifiant_niveau>
Response signée	Oui	
Assertion signée	Oui	
Assertion chiffrée	Optionnel	
Format du sujet	Transient	urn:oasis:names:tc:SAML:2.0:nameid-format:transient

3.4.7.3 Service de traitement de la déconnexion

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL d'écoute	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/saml2/sp/slo/logout?client_name=<identifiant_niveau>
URL de retour	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/saml2/sp/slo/logout?client_name=<identifiant_niveau>
LogoutRequest signée	Oui	
LogoutResponse signée	Oui	

3.4.7.4 Signature et chiffrement

Caractéristique	Valeur	Exemple
Algorithme de Hachage	SHA1, SHA256	http://www.w3.org/2000/09/xmldsig#sha1 http://www.w3.org/2001/04/xmlenc#sha256
Algorithme de signature	RSA-SHA1, RSA-SHA256	http://www.w3.org/2000/09/xmldsig#rsa-sha1 http://www.w3.org/2001/04/xmldsig-more#rsa-sha256
Algorithme de chiffrement	AES256-CBC	http://www.w3.org/2001/04/xmlenc#aes256-cbc

Suite au paramétrage du niveau d'authentification associé, les métadonnées du SP sont disponibles à l'URL suivante : https//www.domaine.fr/ideosso/saml2/sp/metadata?client_name=<identifiant_niveau>

Ci-dessous, un exemple de métadonnées SAML 2.0 du serveur IdéoSSO en tant que fournisseur de service.

```
<?xml version="1.0" encoding="UTF-8"?>
<EntitiesDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:mdalg="urn:oasis:names:tc:SAML:metadata:algsupport"
        xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://docs.oasis-open.org/security/saml/v2.0/saml-schema-metadata-2.0.xsd">
        <EntityDescriptor entityID="https://sp-exemple.domaine.fr/ideosso/saml2/sp/adfs-dev">
                <Extensions>
                        <mdalg:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                        <mdalg:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sha256" />
                        <mdalg:SigningMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
                        <mdalg:SigningMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256" />
                        <mdalg:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
```

```
                </Extensions>
                <SPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
                        <Extensions>
                                <mdui:UIInfo>
                                        <mdui:DisplayName>Mon compte Pro</mdui:DisplayName>
                                        <mdui:Description>Mon compte Pro</mdui:Description>
                                </mdui:UIInfo>
                        </Extensions>
                        <KeyDescriptor use="signing">
                                <ds:KeyInfo>
                                        <ds:X509Data>

        <ds:X509Certificate>LS0tLS1CRUdJTiB...ElGSUNBVEUtLS0tLQo=</ds:X509Certificate
                                        </ds:X509Data>
                                </ds:KeyInfo>
                        </KeyDescriptor>
                        <KeyDescriptor use="encryption">
                                <ds:KeyInfo>
                                        <ds:X509Data>

        <ds:X509Certificate>LS0tLS1CRUdJTiB...ElGSUNBVEUtLS0tLQo=</ds:X509Certificate
                                        </ds:X509Data>
                                </ds:KeyInfo>
                        </KeyDescriptor>
                        <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://sp-exemple.domaine.fr/ideosso/saml2/sp/slo/logout?client_name=saml2_adfs-dev"
ResponseLocation="https://sp-exemple.domaine.fr/ideosso/saml2/sp/slo/logout?client_name=saml2_adfs-dev"
/>
                        <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-
format:transient</NameIDFormat>
                        <AssertionConsumerService index="1" isDefault="true"
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
        Location="https://sp-exemple.domaine.fr/ideosso/login?client_name=saml2_adfs-dev" />
                </SPSSODescriptor>
        </EntityDescriptor>
</EntitiesDescriptor>
```

3.4.8. Déclaration de la relation d'approbation

La déclaration de la relation d'approbation dans l'ADFS utilise les métadonnées SAML 2.0 du serveur IdéoSSO en tant que fournisseur de service.

Pour rappel, elles sont disponibles à l'URL suivante :
https//www.domaine.fr/ideosso/saml2/sp/metadata?client_name=<identifiant_niveau>

La déclaration de la relation d'approbation est décrite en annexe de ce document.

3.4.9. Obtention de l'assertion SAML

L'assertion SAML est délivrée par le fournisseur d'identité suite à une requête du serveur IdéoSSO (authNRequest) et à l'authentification réussie de l'utilisateur.

Ci-dessous est présentée une assertion SAML 2.0 compatible et signée :

```
<Assertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion" ID="_034a3a1e-f8ea-41b7-a1b1-957cc1d3e8bb"
IssueInstant="2018-06-08T19:06:56.107Z" Version="2.0">
   <Issuer>http://adfs-exemple.domaine.fradfs/services/trust</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/2001/04/xmldsig-more#rsa-sha256"/>
        <ds:Reference URI="#_034a3a1e-f8ea-41b7-a1b1-957cc1d3e8bb">
          <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/2001/04/xmlenc#sha256"/>
          <ds:DigestValue>vvVAMX/8FHalhiC63SPw6KyicFKkyprUztSlAZNBLuI=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>MO0SHS3J62...YnzU8owOA==</ds:SignatureValue>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>LS0tLS1CRUdJTiB...ElGSUNBVEUtLS0tLQo=</ds:X509Certificate
        </ds:X509Data>
      </KeyInfo>
    </ds:Signature>
    <Subject>
      <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:transient">mbrisou@ideosante.local</NameID>
      <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <SubjectConfirmationData InResponseTo="_siqqgptc1oashxyh963rxcnwrimt5boxneqrx0y"
NotOnOrAfter="2018-06-08T19:11:56.107Z" Recipient="https://sp-
exemple.domaine.fr/ideosso/login?client_name=saml2_adfs-dev"/>
      </SubjectConfirmation>
    </Subject>
    <Conditions NotBefore="2018-06-08T19:06:56.104Z" NotOnOrAfter="2018-06-08T20:06:56.104Z">
      <AudienceRestriction>
        <Audience>https://sp-exemple.domaine.fr/ideosso/saml2/sp/adfs-dev</Audience>
      </AudienceRestriction>
    </Conditions>
    <AttributeStatement>
      <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn">
        <AttributeValue>mbrisou@IDEOSANTE.LOCAL</AttributeValue>
      </Attribute>
      <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname">
        <AttributeValue>BRISOU</AttributeValue>
      </Attribute>
      <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname">
        <AttributeValue>MARTIAL</AttributeValue>
      </Attribute>
      <Attribute Name="CompteUtilisateur.uid">
        <AttributeValue>mbrisou</AttributeValue>
      </Attribute>
    </AttributeStatement>
    <AuthnStatement AuthnInstant="2018-06-08T19:06:55.968Z" SessionIndex="_034a3a1e-f8ea-41b7-a1b1-
957cc1d3e8bb">
      <AuthnContext>
        <AuthnContextClassRef>urn:federation:authentication:windows</AuthnContextClassRef>
      </AuthnContext>
    </AuthnStatement>
  </Assertion>
```

La signature de l'assertion SAML 2.0 par l'IDP est obligatoire afin d'assurer la non-répudiation de l'origine.

3.4.10. Transmission de l'assertion SAML

L'assertion SAML 2.0 est transmise au serveur IdéoSSO dans une réponse SAML 2.0. Le transport supporté et mis en œuvre est *HTTT-POST Binding*. Les échanges de messages d'un transport *HTTP-POST Binding* sont présentés dans la documentation officielle SAML 2.0.

Ci-dessous, un exemple de réponse SAML 2.0 contenant l'assertion compatible et signée :

```
<?xml version="1.0"?>
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="_ee70f881-f9fe-4243-b696-
f22cf419db22" Version="2.0" IssueInstant="2018-06-08T19:06:56.107Z" Destination="https://sp-
exemple.domaine.fr/ideosso/login?client_name=saml2_adfs-dev"
Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified"
InResponseTo="_siqqgptc1oashxyh963rxcnwrimt5boxneqrx0y">
  <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">http://adfs-
exemple.domaine.fr/adfs/services/trust</Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <Assertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion" ID="_034a3a1e-f8ea-41b7-a1b1-957cc1d3e8bb"
IssueInstant="2018-06-08T19:06:56.107Z" Version="2.0">
          ...
  </Assertion>
</samlp:Response>
```

La signature de la réponse SAML 2.0 par l'ADFS est nécessaire afin de garantir la non-interception de la réponse.

3.4.11. Fédération de l'identité (UPN)

La fédération des identités est effectuée à partir des identifiants nationaux du compte utilisateur. L'identifiant national peut être véhiculé dans le sujet et/ou dans un attribut de l'assertion. Dans le cas d'une fédération avec ADFS, l'identifiant national véhiculé doit être au format User-Principal-Name.

La description de l'attribut User-Principal-Name est disponible à l'URL suivante : https://msdn.microsoft.com/en-us/library/windows/desktop/aa380525(v=vs.85).aspx

Ci-dessous, l'UPN est véhiculé dans le sujet de l'assertion :

```
<Subject>
    <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:transient">mbrisou@ideosante.local</NameID>
    <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
      <SubjectConfirmationData InResponseTo="_siqqgptc1oashxyh963rxcnwrimt5boxneqrx0y"
NotOnOrAfter="2018-06-08T19:11:56.107Z" Recipient="https://sp-
exemple.domaine.fr/ideosso/login?client_name=saml2_adfs-dev"/>
    </SubjectConfirmation>
</Subject>
```

Ci-dessous, l'UPN est présent dans un attribut standard de l'assertion :

```
<Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn">
      <AttributeValue>mbrisou@IDEOSANTE.LOCAL</AttributeValue>
</Attribute>
```

Afin de garantir une meilleure correspondance de l'identité, un contrôle supplémentaire sur le nom et le prénom peut être effectué par IdéoSSO.

```
<Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname">
      <AttributeValue>BRISOU</AttributeValue>
</Attribute>
<Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname">
      <AttributeValue>MARTIAL</AttributeValue>
</Attribute>
```

Le comportement et les contrôles de la fédération sont paramétrables dans le niveau d'authentification associé.

3.4.12. Dispositifs d'authentification ADFS

Les dispositifs présentés par l'ADFS sont paramétrables dans le niveau d'authentification associé. IdéoSSO fixe, par défaut, le palier de l'authentification en fonction des dispositifs utilisés.

Le tableau ci-dessous présente les paliers d'une authentification ADFS.

Nom complet	Mode d'authentification	Indice
~~**Login/mot de passe HTTP**~~	~~urn:oasis:names:tc:SAML:2.0:ac:classes:Password~~	~~0~~
Login/mot de passe HTTPS	urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport	1
Certificat SSL	urn:oasis:names:tc:SAML:2.0:ac:classes:TLSClient	2
Certificat X509	urn:oasis:names:tc:SAML:2.0:ac:classes:X509	2
Session Windows	urn:federation:authentication:windows	4
Authentification Kerberos	urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos	4

3.4.13. Ouverture contextuelle non sollicitée (RelayState)

L'ouverture contextuelle est une authentification déléguée et passive de l'utilisateur au serveur IdéoSSO ayant pour objectif d'ouvrir une application cible.

En tant que fournisseur de service SAML 2.0, le serveur IdéoSSO supporte les réponses non sollicitées. C'est-à-dire que le serveur accepte les réponses SAML 2.0 dont il n'est pas à l'initiative. La réponse SAML 2.0 est transmise à l'initiative de l'ADFS.

La fonctionnalité SAML 2.0 mise en œuvre se nomme *Unsolicited Reponses*.

Le tableau ci-dessous présente les caractéristiques de cette réponse non sollicitée :

Caractéristique	Valeur	Exemple
Binding	HTTP-POST Binding	urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST
URL de destination	URL sécurisée HTTPS	https://www.domaine.fr/ideosso/login?client_name=<identifiant_niveau>
Response signée	Oui	
Assertion signée	Oui	
Assertion chiffrée	Optionnel	
Format du sujet	Transient	urn:oasis:names:tc:SAML:2.0:nameid-format:transient
InResponseTo	Vide	

La documentation décrivant la mise en œuvre des réponses non sollicitées avec ADFS est disponible à l'URL suivante : https://blogs.technet.microsoft.com/askds/2012/09/27/ad-fs-2-0-relaystate/

Ci-dessous, un exemple de réponse non sollicitée :

```
<?xml version="1.0"?>
<samlp:Response xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol" ID="_9ed67bed-3fb1-46ed-b55c-
8912045d2ac0" Version="2.0" IssueInstant="2018-06-08T20:16:54.782Z" Destination="https://sp-
exemple.domaine.fr/ideosso/login?client_name=saml2_adfs-dev"
Consent="urn:oasis:names:tc:SAML:2.0:consent:unspecified">
  <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">http://adfs-
exemple.domaine.fr/adfs/services/trust</Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>
  <Assertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion" ID="_ac1a5376-a7f4-4c64-8ca3-344599631072"
IssueInstant="2018-06-08T20:16:54.712Z" Version="2.0">
    <Issuer>http://adfs-exemple.domaine.fr/adfs/services/trust</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/2001/04/xmldsig-more#rsa-sha256"/>
        <ds:Reference URI="#_ac1a5376-a7f4-4c64-8ca3-344599631072">
          <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/2001/04/xmlenc#sha256"/>
          <ds:DigestValue>ZYTfMYdZfVXeri5K7zBoyJFwCyyoYOZeIse0mWkpB3g=</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>KzMY0ECB11mM...g9mTcmPWA==</ds:SignatureValue>
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>MIIC6DCCAdCg...ulxWxxiZuc8A==</ds:X509Certificate>
        </ds:X509Data>
      </KeyInfo>
    </ds:Signature>
    <Subject>
      <NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:transient">mbrisou@ideosante.local</NameID>
      <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        <SubjectConfirmationData NotOnOrAfter="2018-06-08T20:21:54.782Z" Recipient="https://sp-
exemple.domaine.fr/ideosso/login?client_name=saml2_adfs-dev"/>
      </SubjectConfirmation>
    </Subject>
    <Conditions NotBefore="2018-06-08T20:16:54.677Z" NotOnOrAfter="2018-06-08T21:16:54.677Z">
      <AudienceRestriction>
        <Audience>https://sp-exemple.domaine.fr/ideosso/saml2/sp/adfs-dev</Audience>
      </AudienceRestriction>
    </Conditions>
    <AttributeStatement>
      <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname">
        <AttributeValue>MARTIAL</AttributeValue>
      </Attribute>
      <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname">
        <AttributeValue>BRISOU</AttributeValue>
      </Attribute>
```

```
    <Attribute Name="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn">
      <AttributeValue>mbrisou@IDEOSANTE.LOCAL</AttributeValue>
    </Attribute>
    <Attribute Name="psIdNat">
      <AttributeValue>579408857500053/8481</AttributeValue>
    </Attribute>
    <Attribute Name="psGroupes">
      <AttributeValue>Utilisateurs du domaine</AttributeValue>
      <AttributeValue>Administrateurs de l&#x2019;entreprise</AttributeValue>
    </Attribute>
    <Attribute Name="CompteUtilisateur.uid">
      <AttributeValue>mbrisou</AttributeValue>
    </Attribute>
    <Attribute Name="Personne.idNat">
      <AttributeValue>579408857500053/8481</AttributeValue>
    </Attribute>
  </AttributeStatement>
  <AuthnStatement AuthnInstant="2018-06-08T20:16:54.598Z" SessionIndex="_ac1a5376-a7f4-4c64-8ca3-
344599631072">
    <AuthnContext>
      <AuthnContextClassRef>urn:federation:authentication:windows</AuthnContextClassRef>
    </AuthnContext>
  </AuthnStatement>
 </Assertion>
</samlp:Response>
```

La réponse non sollicitée permet l'authentification passive de l'utilisateur. L'ouverture contextuelle de l'application cible est effectuée en utilisant le paramètre *RelayState* du transport *HTTP-POST Binding*. Suite à l'authentification passive, le serveur IdéoSSO redirigera le navigateur Internet vers l'URL spécifiée.

Le paramètre *RelayState* doit être au format suivant :
https://www.domaine.fr/ideosso/login?service=<service_url>&redirect=<redirect_url>

Les paramètres de l'URL sont :

Caractéristique	Valeur	Exemple
<Service_URL>	Identifiant de l'application cible dans l'annuaire IdéoDirectory. Ce paramètre doit être URL encodé.	https://www.domaine.fr/application/
<Redirect_URL>	URL de redirection suite à l'authentification passive réussie. Ce paramètre doit être URL encodé.	https://www.domaine.fr/application/OuvertureContextuelle?param1=value1&param2=value2

Un exemple de paramètre *RelayState* pour l'ouverture contextuelle de l'application cible :
https://www.domaine.fr/ideosso/login?service=https%3A%2F%2Fwww.domaine.fr%2Fapplication%2F&redirect=https%3A%2F%2Fwww.domaine.fr%2Fapplication%2FOuvertureContextuelle%3Fparam1%3Dvalue1%26param2%3Dvalue2

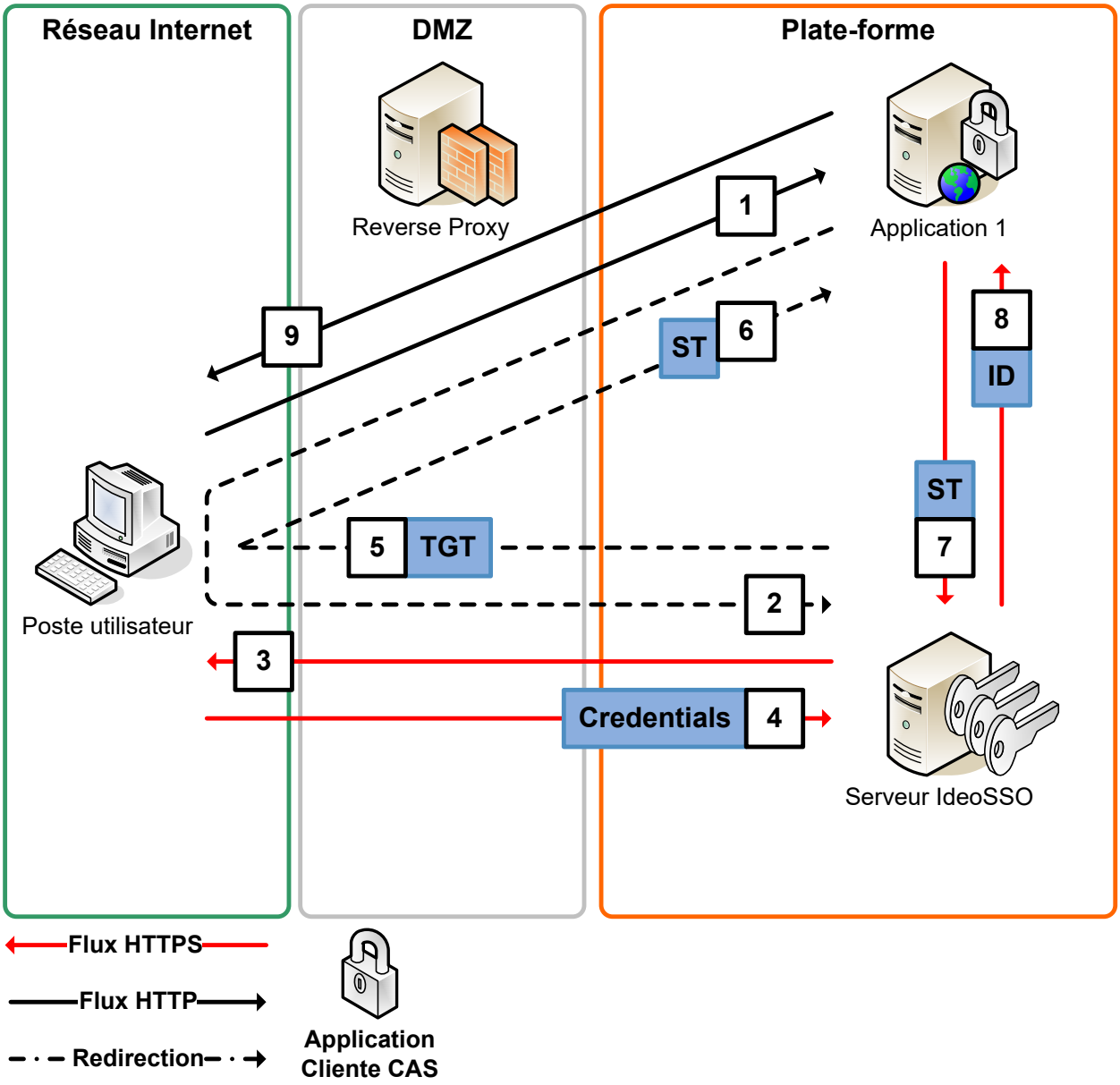
3.5. Authentification déléguée France Connect (FranceConnect)

A venir.

4. Annexe : Flux CAS 2.0 et CAS 3.0

Les paragraphes ci-après présentent les flux mis en œuvre dans le protocole CAS 2.0 et CAS 3.0.

4.1. Flux de l'identification de l'utilisateur



Processus d'identification d'un utilisateur via un serveur IdéoSSO :

1 : L'utilisateur souhaite se connecter à l'application **1**, il utilise son navigateur Internet pour se connecter à l'application.

2 : Le navigateur Internet ne présente pas de Service Ticket (ST) à l'application, l'application **1** redirige alors l'utilisateur vers le serveur IdéoSSO pour obtenir un ST pour l'application **1.**

3 : Pour obtenir un Service Ticket pour l'application **1**, le navigateur doit préalablement avoir obtenu un Granting Ticket (TGT). Dans le cas présent, le navigateur ne peut pas présenter de TGT au serveur IdéoSSO. Le serveur propose alors le formulaire d'authentification à l'utilisateur. Le flux utilisé est un flux HTTPS. La validation de ce formulaire permettra d'obtenir un TGT.

4 : L'utilisateur remplit le formulaire d'authentification choisi. Ce formulaire peut être un formulaire classique, par Identifiant/Mot de passe ou un formulaire évolué, par carte CPS. Ensuite, il soumet ce formulaire, par flux HTTPS, au serveur IdéoSSO.

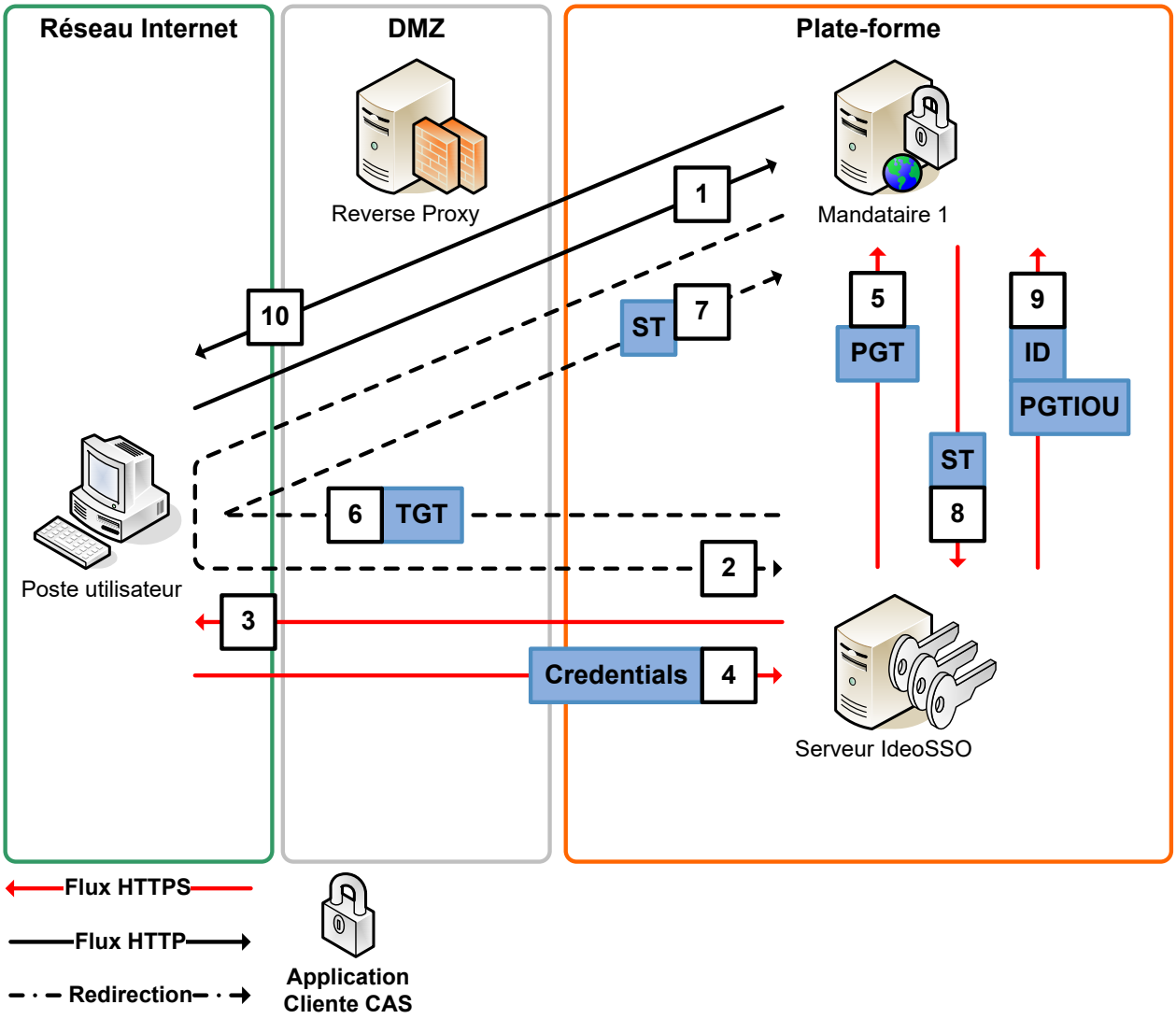
5 - 6: Le serveur IdéoSSO authentifie l'utilisateur grâce aux données entrées dans le formulaire d'authentification et aux données présentes dans l'annuaire de sécurité. Lorsque l'utilisateur est authentifié, le serveur IdéoSSO envoie un TGT et un ST non re-jouable pour l'application **1** au navigateur Internet et redirige le navigateur vers l'application **1**.

7 : L'application **1** valide le ST auprès du serveur IdéoSSO. Cette validation permet à l'application d'obtenir l'identité de l'utilisateur qui se présente à l'application. Elle est faite par des flux chiffrés par HTTPS.

8 : Le serveur IdéoSSO donne l'identifiant de l'utilisateur à l'application **1**.

9 : L'application **1** identifie donc l'utilisateur au sein de son système. Elle présente donc la page d'accueil personnalisée de l'application à utilisateur.

4.2. Flux de l'identification de l'utilisateur pour un mandataire



Processus d'identification d'un utilisateur pour un mandataire via un serveur IdéoSSO :

1 : L'utilisateur souhaite se connecter au mandataire **1**, il utilise son navigateur Internet pour se connecter au mandataire.

2 : Le navigateur Internet ne présente pas de ST au mandataire, le mandataire **1** redirige alors l'utilisateur vers le serveur IdéoSSO pour obtenir un ST pour le mandataire **1.**

3 : Pour obtenir un ST pour le mandataire **1**, le navigateur doit préalablement avoir obtenu un TGT. Dans le cas présent, le navigateur ne peut pas présenter de TGT au serveur IdéoSSO. Le serveur propose alors le formulaire d'authentification à l'utilisateur. Le flux utilisé est un flux HTTPS.

4 : L'utilisateur remplit le formulaire d'authentification choisi. Ce formulaire peut être un formulaire classique, par Identifiant/Mot de passe ou un formulaire évolué, par carte CPS. Ensuite, il soumet ce formulaire, par flux HTTPS, au serveur IdéoSSO.

5 : Le serveur IdéoSSO authentifie l'utilisateur grâce aux données entrées dans le formulaire d'authentification et aux données présentes dans l'annuaire de sécurité. De plus, il soumet une requête d'un Proxy Granting Ticket (PGT) au mandataire.

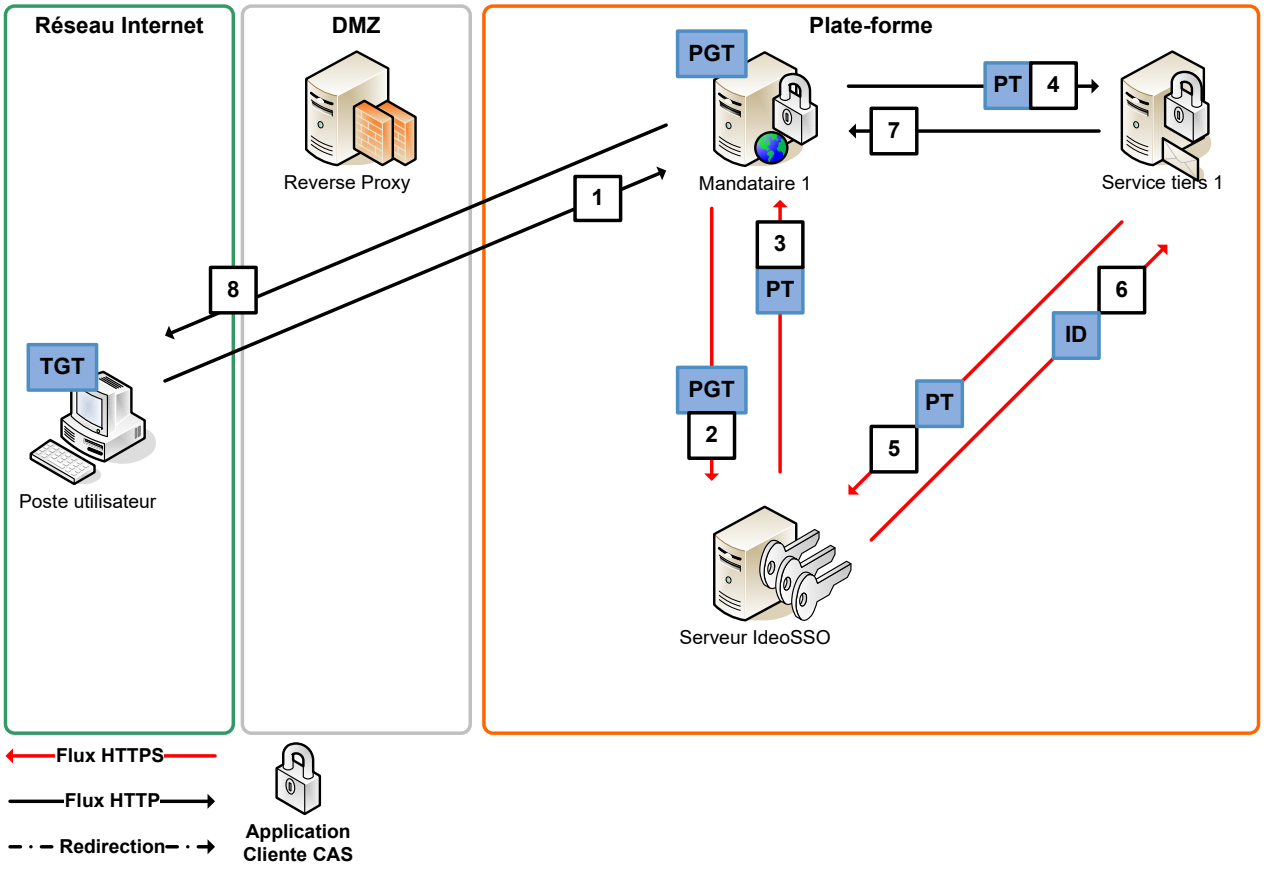
6 - 7: L'utilisateur est authentifié, le serveur IdéoSSO envoie un TGT et un ST non re-jouable pour le mandataire **1** au navigateur Internet et redirige le navigateur vers le mandataire **1**.

8 : Le mandataire **1** valide le ST auprès du serveur IdéoSSO. Cette validation permet à l'application d'obtenir l'identité de l'utilisateur qui se présente au mandataire. Elle est faite par des flux chiffrés par HTTPS. De plus, elle permet d'obtenir le PGT-IOU qui est en quelques sortes la deuxième partie du Proxy Granting Ticket. Ce PGT-IOU permet au mandataire de retrouver le PGT distribuer par le serveur IdéoSSO.

9 : Le serveur IdéoSSO donne l'identifiant de l'utilisateur au mandataire **1,** ainsi que le PGT-IOU.

10 : Le mandataire **1** identifie donc l'utilisateur au sein de son système. Il présente donc la page d'accueil personnalisée de l'application à utilisateur.

4.3. Flux de l'utilisation du mode mandataire



Processus d'utilisation de Proxy Ticket par un mandataire via un serveur IdéoSSO :

0 : L'utilisateur est préalablement authentifié auprès du serveur IdéoSSO via le processus de connexion à un mandataire.

1 : L'utilisateur souhaite utiliser un service tiers via le mandataire **1**. Il utilise son navigateur pour se connecter au mandataire **1.**

2 : Le mandataire présente son PGT auprès du serveur IdéoSSO afin d'obtenir un Proxy Ticket (PT).

3 : Le serveur IdéoSSO retourne le PT au mandataire pour le service **1** et l'utilisateur connecté.

4 : Le mandataire **1** présente le PT au service tiers **1.** Le service tiers 1 est aussi un client IdéoSSO, il est capable de valider les PT auprès du serveur de la plate-forme.

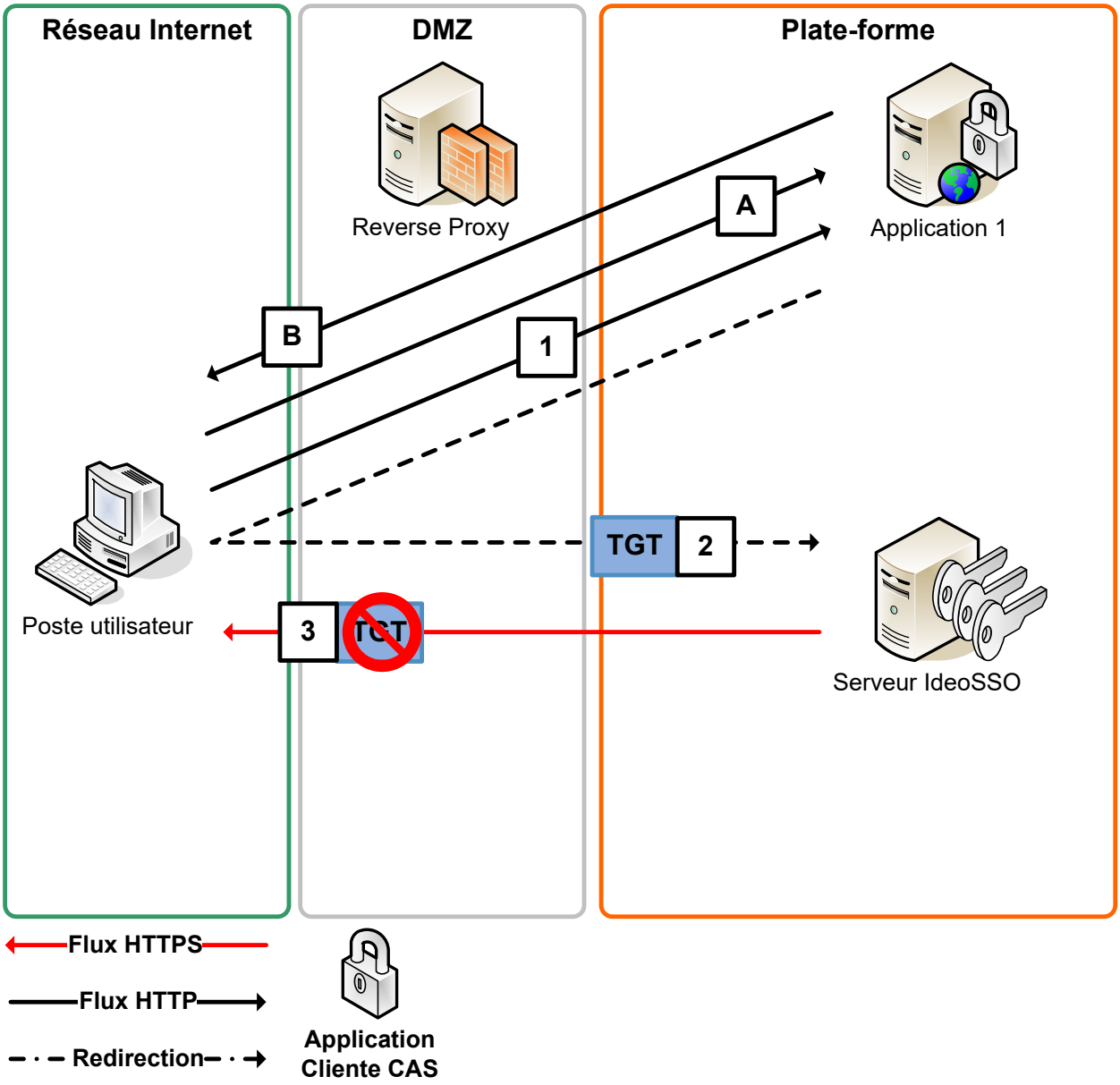
5 : Le service tiers **1** valide le PT auprès du serveur IdéoSSO. Cette validation permet au service tiers **1** d'obtenir l'identité de l'utilisateur qui se présente. Elle est faite par des flux chiffrés par HTTPS.

6 : Le serveur IdéoSSO donne l'identifiant de l'utilisateur au service tiers **1**. L'utilisateur est alors connecté au service tiers **1** via le mécanisme IdéoSSO.

7 : Le mandataire **1** utilise le service tiers 1 en étant connecté avec le compte de l'utilisateur.

8 : Le mandataire **1** présente la page de résultat au navigateur de l'utilisateur.

4.4. Flux de déconnexion de l'utilisateur



Processus de déconnexion partielle d'une application cliente IdéoSSO :

A : L'utilisateur fait une requête de déconnexion sur l'application **1.**

B : L'application **1** déconnecte l'utilisateur de l'application. Cette déconnexion ne déconnecte pas l'utilisateur du serveur IdéoSSO.

Processus de déconnexion totale d'une application cliente IdéoSSO :

1 : L'utilisateur fait une requête de déconnexion sur l'application **1.**

2: L'application **1** déconnecte l'utilisateur de l'application. De plus, elle redirige le navigateur vers le service de déconnexion du serveur IdéoSSO. Le navigateur Internet présente alors le TGT au serveur IdéoSSO.

3 : Le serveur IdéoSSO détruit le TGT du navigateur. L'utilisateur est alors totalement déconnecté du serveur IdéoSSO. A partir de la version 3.0 d'IdéoSSO, le serveur diffuse la connexion du compte utilisateur aux applications clientes.

5. Annexe : Services IdéoSSO

5.1. Les services officiels du serveur CAS

Les appels aux services se font en HTTPS, ce sont des requêtes HTTPS. Pour cela fonctionne, il faut que l'application faisant appel aux services fasse confiance au certificat utilisé pour la requête. Si ce n'est pas le cas, les connexions entre l'application et le serveur IdéoSSO ne pourront pas être établies. Les réponses du serveur CAS sont des flux XML simple. Les services de la distribution officielle de CAS sont :

- serviceValidate, proxyValidate
- proxy

Nous allons maintenant décrire les interfaces de ces services.

5.1.1. Le service serviceValidate, proxyValidate

L'URL de ce service est : https://.../ideosso/serviceValidate **ou** https://.../ideosso/proxyValidate

Le tableau suivant présente les paramètres attendus par ce service.

Nom du paramètre	*Description*	*Exemple*
`service`	L'identifiant de l'application	http://monapplication/
`ticket`	Le ticket à valider	**ST-X-YYYYYY….**
`pgtUrl`	L'URL du récepteur de proxyGrantingTicket.	http://monapplication/receptor

Le paramètre **service** correspond à l'identifiant de l'application faisant appel au service. Le paramètre **ticket** est un Proxy Ticket obtenu pour l'application afin d'utiliser le service.

Le paramètre **pgtUrl** indique l'URL du récepteur de Proxy Granting Ticket. Le service proxyValidate va faire une requête sur cette URL pour fournir à l'application le PGT.

La réponse positive du service est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
        <cas:authenticationSuccess>
                <cas:user>${userId}</cas:user>
                <cas:proxyGrantingTicket>${pgtIou}</cas:proxyGrantingTicket>
                <cas:proxies>
                        <cas:proxy>${proxy.principal.id}</cas:proxy>
            ...
          <cas:proxy>${proxy.principal.idN}</cas:proxy>
                </cas:proxies>
         </cas:authenticationSuccess>
</cas:serviceResponse>
```

La variable **userId** correspond à l'identifiant de l'utilisateur connecté au serveur IdéoSSO. La variable **pgtIou** correspond à la deuxième partie du PGT. Ce pgtIou sera utilisé avec le PGT (posté sur pgtUrl) pour obtenir un Service Ticket grâce du service proxy. Les variables **proxy.principal.idN** indiquent la liste des mandataires autorisés pour l'application.

La réponse négative en cas de problème est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
        <cas:authenticationFailure code='${code}'>
```

```
                ${description}
        </cas:authenticationFailure>
</cas:serviceResponse>
```

La variable **code** correspond au code de l'erreur. La variable **description** correspond à la description de l'erreur.

5.1.2. Le service proxy

L'URL de ce service est : https://.../ideosso/proxy

Le tableau suivant présente les paramètres attendus par ce service.

Nom du paramètre	*Description*	*Exemple*
`targetService`	L'identifiant de l'application souhaitant un Service Ticket	http://monapplication2/
`pgt`	Le PGT	TGT-X-YYYYYY….

Le paramètre **targetService** correspond à l'identifiant de l'application souhaitant le Proxy Ticket. Le paramètre **pgt** est le Proxy Granting Ticket obtenu pour l'application afin d'utiliser le service.

La réponse positive du service est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
        <cas:proxySuccess>
                <cas:proxyTicket>${ticket}</cas:proxyTicket>
        </cas:proxySuccess>
</cas:serviceResponse>
```

La variable **ticket** correspond au Proxy Ticket délivré pour l'application targetService.

La réponse négative en cas de problème est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
        <cas:proxyFailure code='${code}'>
                ${description}
        </cas:proxyFailure>
</cas:serviceResponse>
```

La variable **code** correspond au code de l'erreur. La variable **description** correspond à la description de l'erreur.

5.2. Les services additionnels IdéoSSO

Pour assurer une plus grande sécurité pour les applications clientes IdéoSSO, de nouveaux services ont été ajoutés au serveur IdéoSSO. La suite de cette partie présente ces nouveaux services. Les services supplémentaires sont les suivants :

- userInformation
- keepGrantorAlive
- deleteGrantor

5.2.1. Le service userInformation

Depuis la version 3.5 d'IdéoSSO, ce service est obsolète. Les informations fournies sont directement disponibles lors de la validation du ticket de service.

L'URL de ce service est : https://.../ideosso/userInformation

Le tableau suivant présente les paramètres attendus par ce service.

Nom du paramètre	*Description*	*Exemple*
`service`	L'identifiant de l'application	http://monapplication/
`ticket`	Le ticket à valider	ST-X-YYYYYY….
`user`	L'identifiant de l'utilisateur connecté	0000142

Le paramètre **service** correspond à l'identifiant de l'application faisant appel au service. Le paramètre **ticket** est un Proxy Ticket obtenu pour l'application afin d'utiliser le service. Le paramètre **user** doit être égal à l'identifiant de l'utilisateur connecté.

La réponse positive du service est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:informationSuccess>
        <cas:user>${username}</cas:user>
        <cas:userInformation>
                <cas:property name='${p.key}'>${p.value}</cas:property>
        ...
        </cas:userInformation>
</cas:informationSuccess>
</cas:serviceResponse>
```

La variable **username** correspond au paramètre user du service. Ensuite, c'est un ensemble de clef/valeur (**p.key/p.value)** qui liste les informations de l'utilisateur.

La réponse négative en cas de problème est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:informationFailure code='${code}'>
        ${description}
</cas:informationFailure>
</cas:serviceResponse>
```

La variable **code** correspond au code de l'erreur. La variable **description** correspond à la description de l'erreur.

5.2.2. Le service keepGrantorAlive

L'URL de ce service est : https://.../ideosso/keepGrantorAlive

Le tableau suivant présente les paramètres attendus par ce service.

Nom du paramètre	*Description*	*Exemple*
`service`	L'identifiant de l'application	http://monapplication/
`ticket`	Le ticket à valider pour utiliser le service	ST-X-YYYYYY….
`egt`	Le ticket d'expiration	EGT-X-YYYYY….

Le paramètre **service** correspond à l'identifiant de l'application faisant appel au service. Le paramètre **ticket** est un Proxy Ticket obtenu pour l'application afin d'utiliser le service.

Lors du premier appel du service, le ticket d'expiration n'est pas connu. Il faut donc utiliser le paramètre **ticket** pour utiliser le service. La réponse positive du service retournera le ticket d'expiration à utiliser pour les appels suivants du service afin de maintenant en vie le Ticket Granting Ticket.

La réponse positive du service est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:keepGrantorAliveSuccess>
        <cas:keepGrantorAliveTolerance>${expireGrantorTimeout}</cas:keepGrantorAliveTolerance>
        <cas:expireTicket>${expireGrantorTicket}</cas:expireTicket>
</cas:keepGrantorAliveSuccess>
</cas:serviceResponse>
```

La variable **expireGrantorTimeout** indique la durée de vie du ticket d'expiration. Lorsque ce ticket d'expiration expire, il va alors aussi faire expirer le TGT associé. La variable **expireGrantorTicket** correspond au ticket d'expiration qu'il faudra préciser lors des prochains appels du service.

La réponse négative en cas de problème est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:keepGrantorAliveFailure code='${code}'>
        ${description}
</cas:keepGrantorAliveFailure>
</cas:serviceResponse>
```

La variable **code** correspond au code de l'erreur. La variable **description** correspond à la description de l'erreur.

5.2.3. Le service deleteGrantor

L'URL de ce service est : https://.../ideosso/deleteGrantor

Le tableau suivant présente les paramètres attendus par ce service.

Nom du paramètre	*Description*	*Exemple*
`service`	L'identifiant de l'application	http://monapplication/
`ticket`	Le ticket à valider pour utiliser le service	ST-X-YYYYYY….

Le paramètre **service** correspond à l'identifiant de l'application faisant appel au service. Le paramètre **ticket** est un Proxy Ticket obtenu pour l'application afin d'utiliser le service.

La réponse positive du service est :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:deleteGrantorSuccess>
</cas:deleteGrantorSuccess>
</cas:serviceResponse>
```

Aucune information supplémentaire n'est retournée pour une réponse positive.

La réponse négative en cas de problème est :

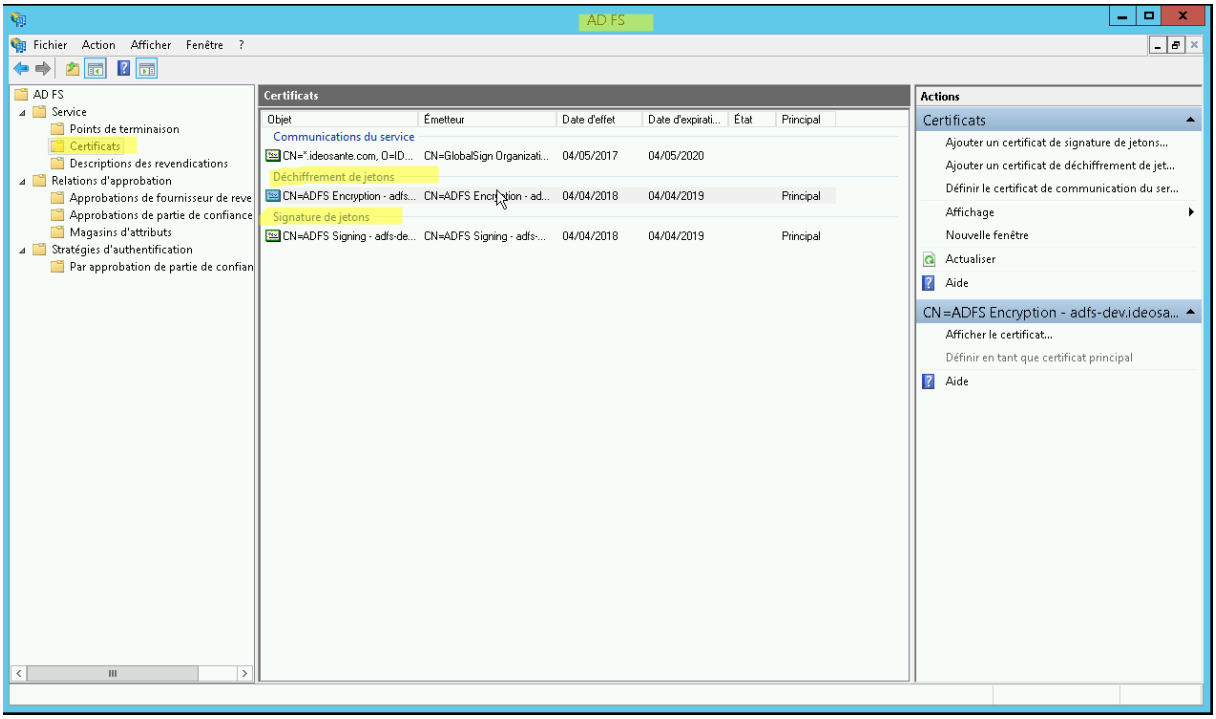
```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
<cas:deleteGrantorFailure code='${code}'>
        ${description}
</cas:deleteGrantorFailure>
</cas:serviceResponse>
```

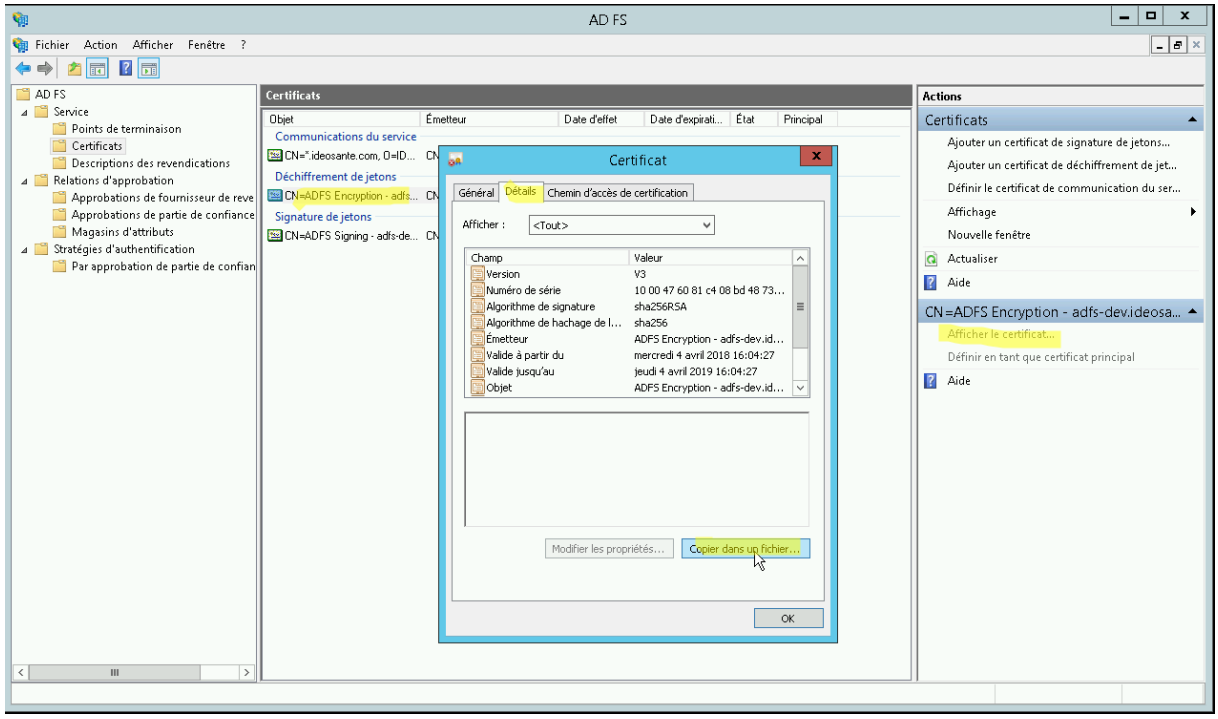
La variable **code** correspond au code de l'erreur. La variable **description** correspond à la description de l'erreur.

6. Annexe : Active Directory Federation Services

6.1. Export des certificats ADFS

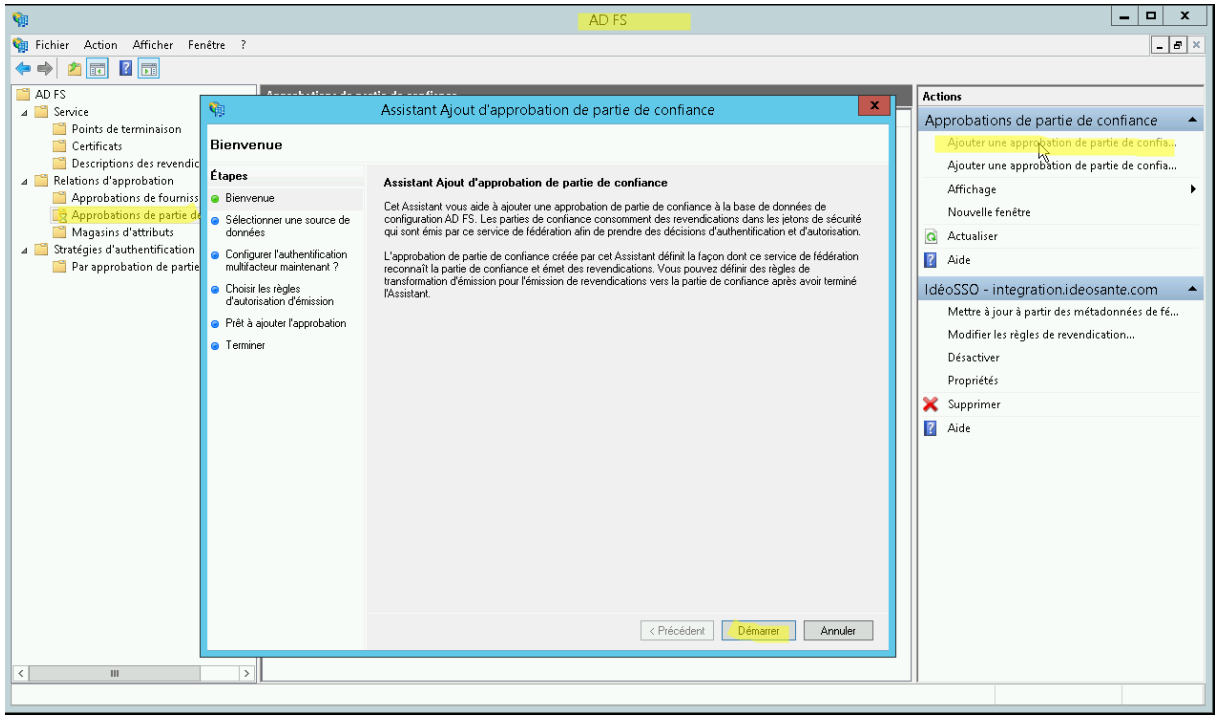
Pour déclarer le niveau d'authentification délégué ADFS dans IdéoSSO, les certificats X509 mis en œuvre sont nécessaires. Ces certificats sont disponibles dans la console d'administration ADFS.

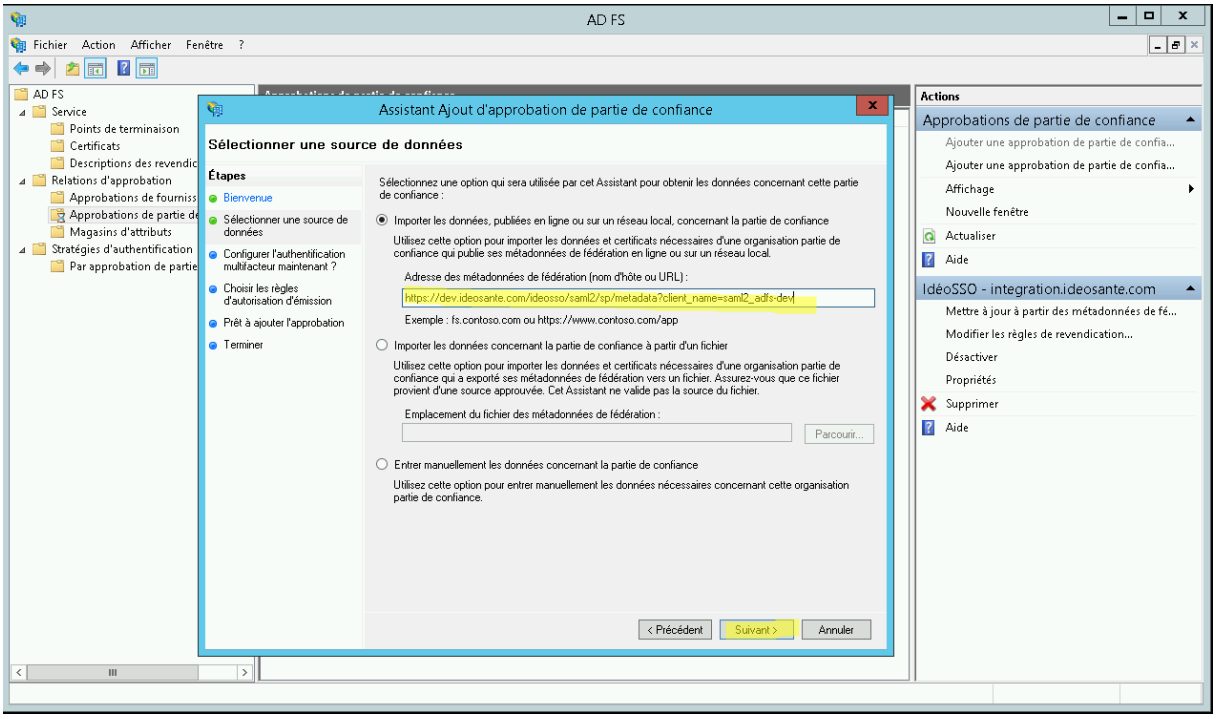


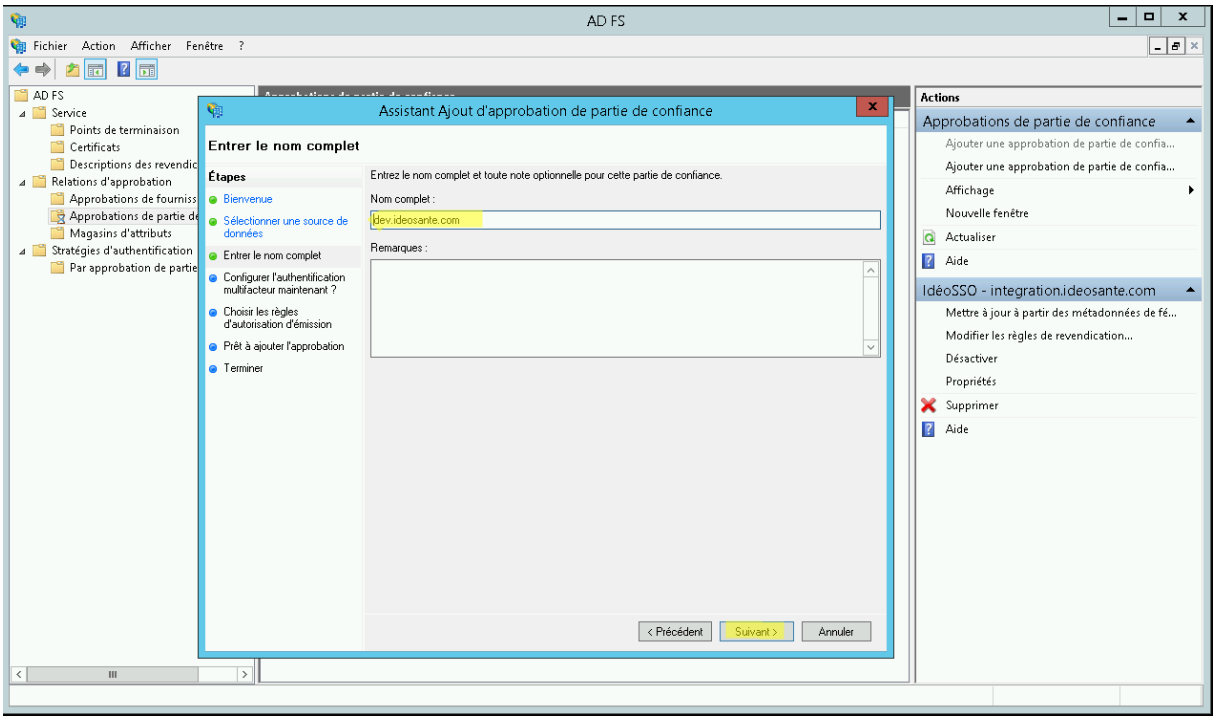


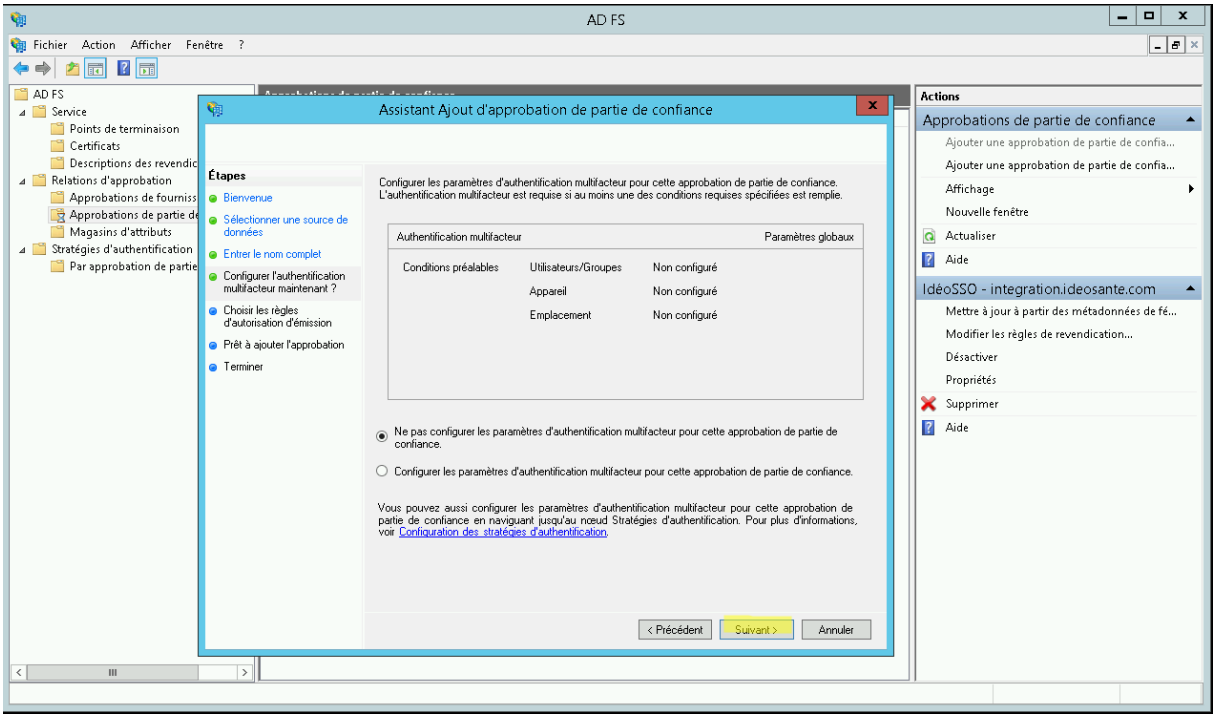
6.2. Déclaration de la relation d'approbation avec un serveur IdéoSSO

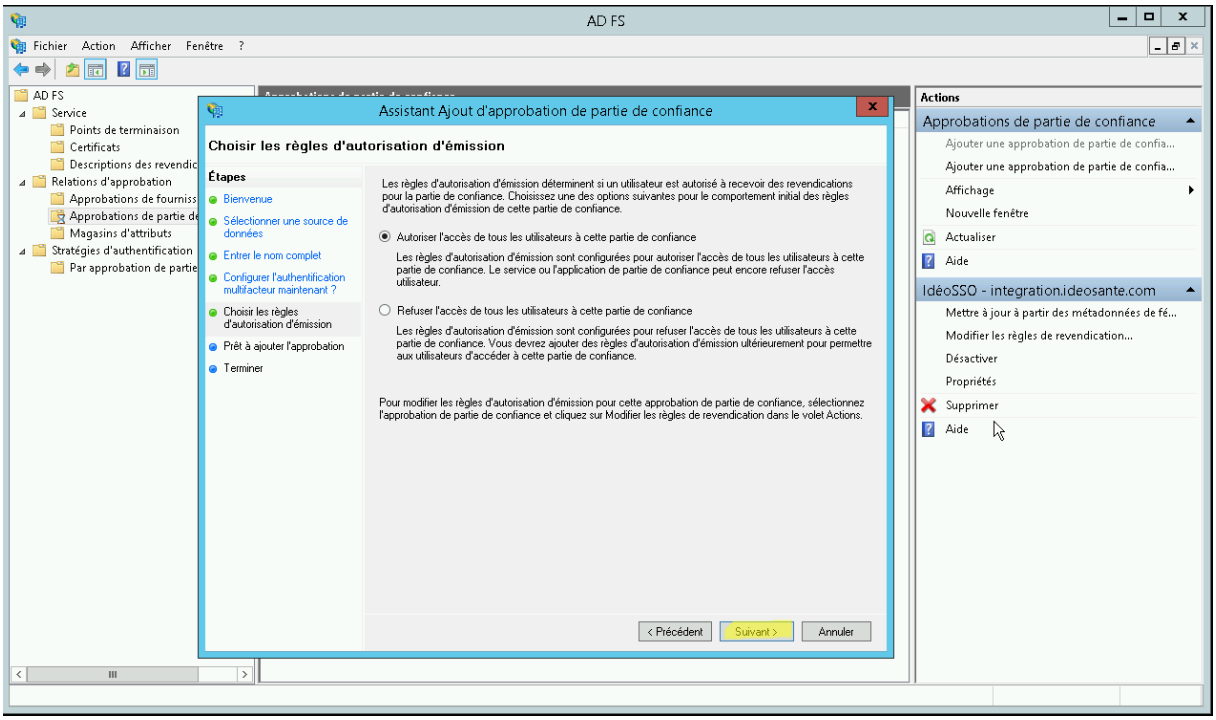
Les copies d'écran suivantes décrivent la déclaration de la relation d'approbation d'un niveau d'authentification déléguée ADFS d'un serveur IdéoSSO.

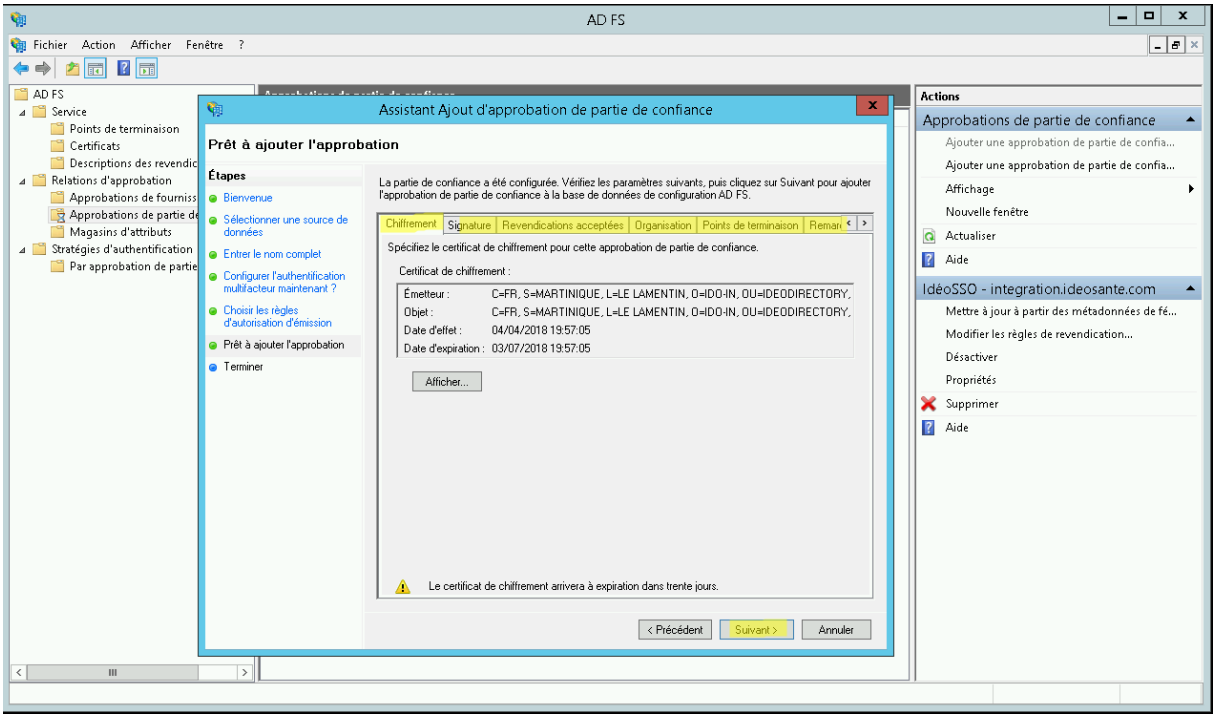


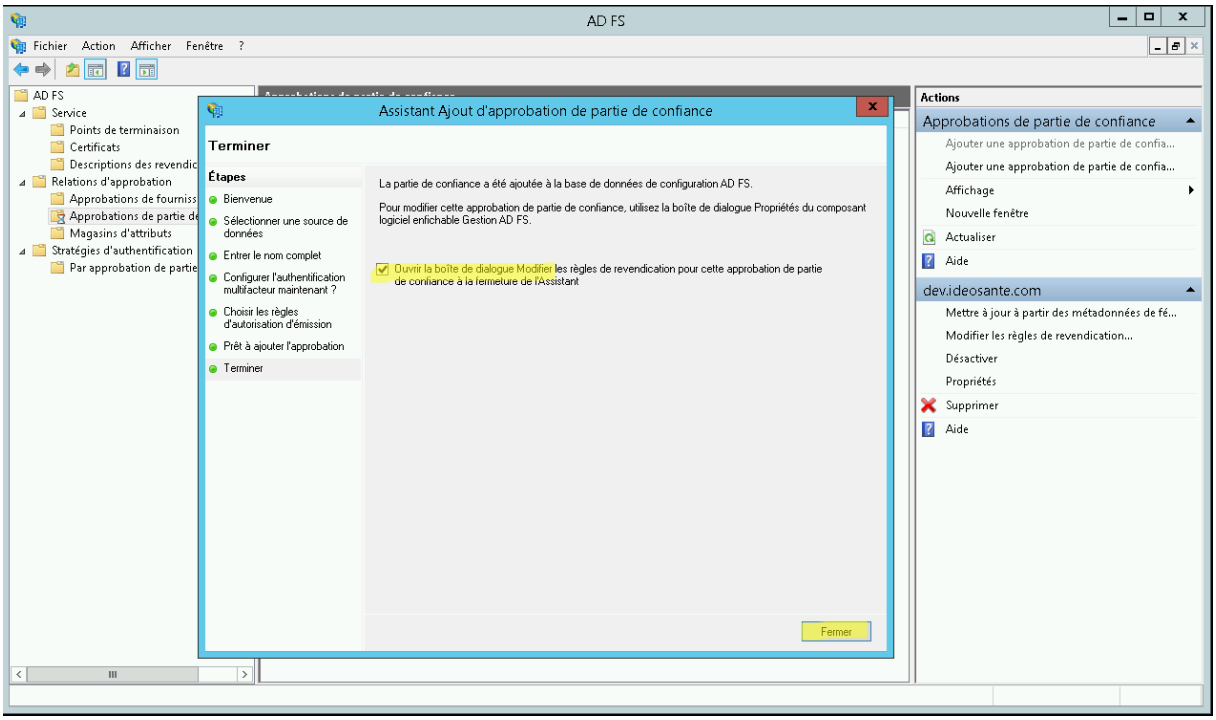


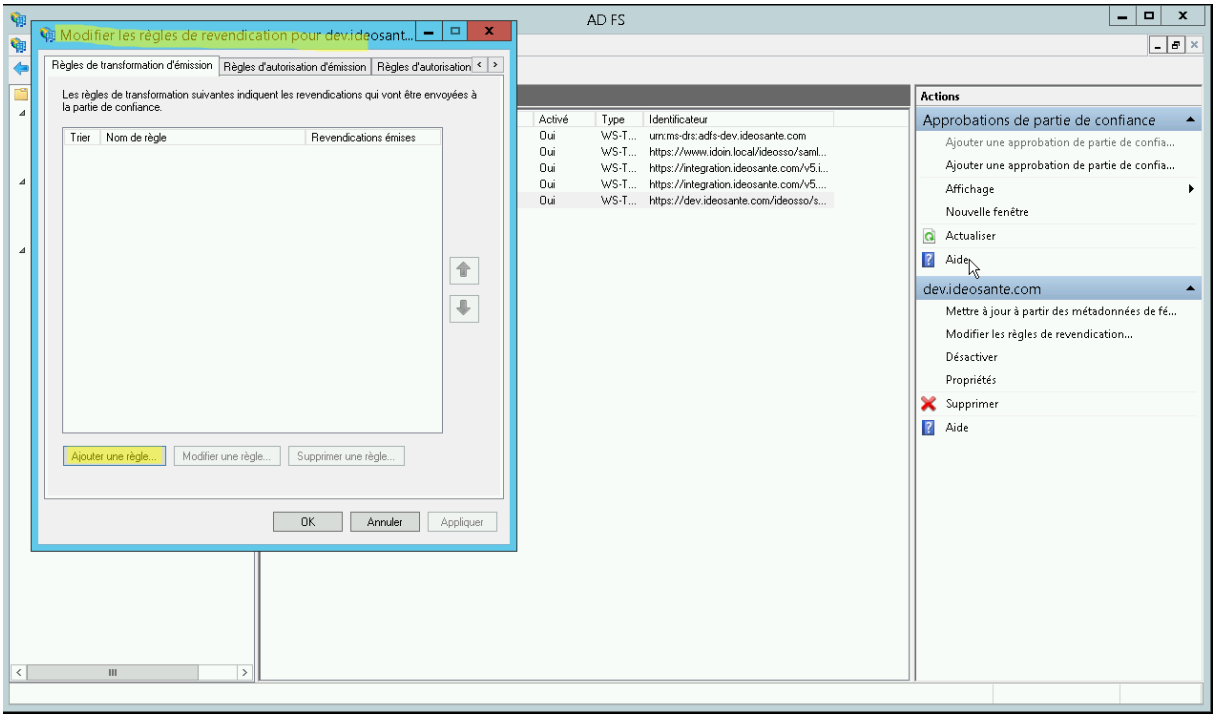


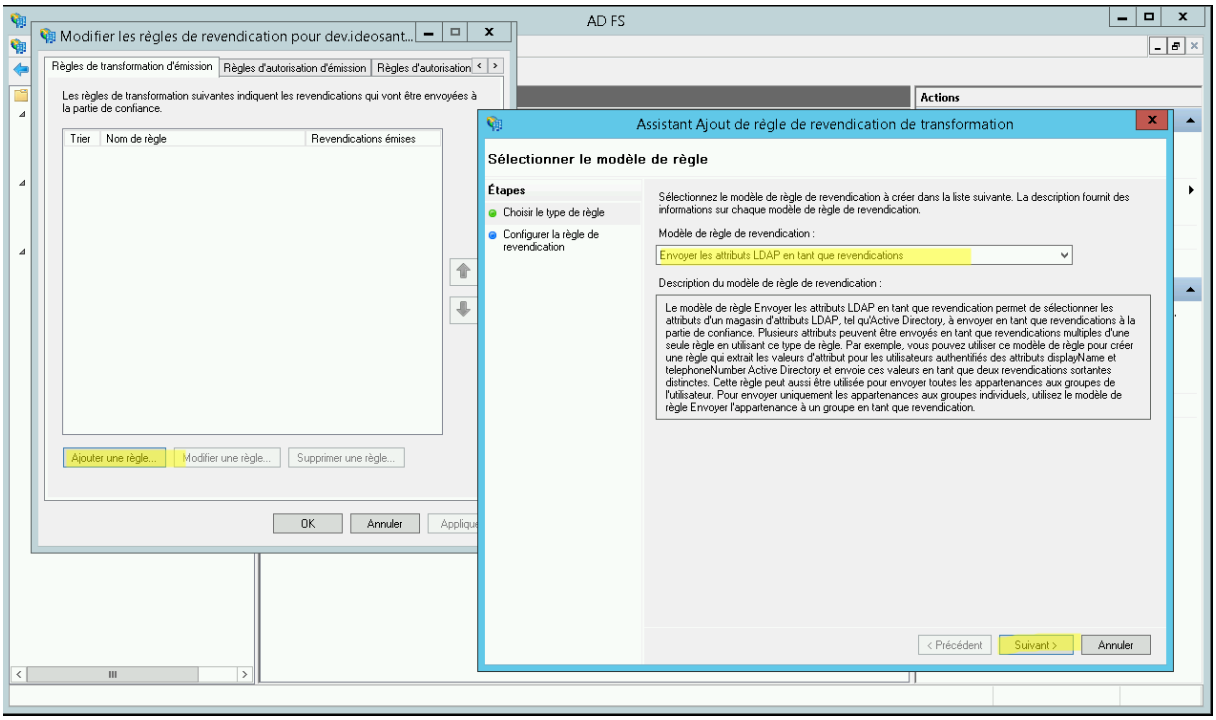


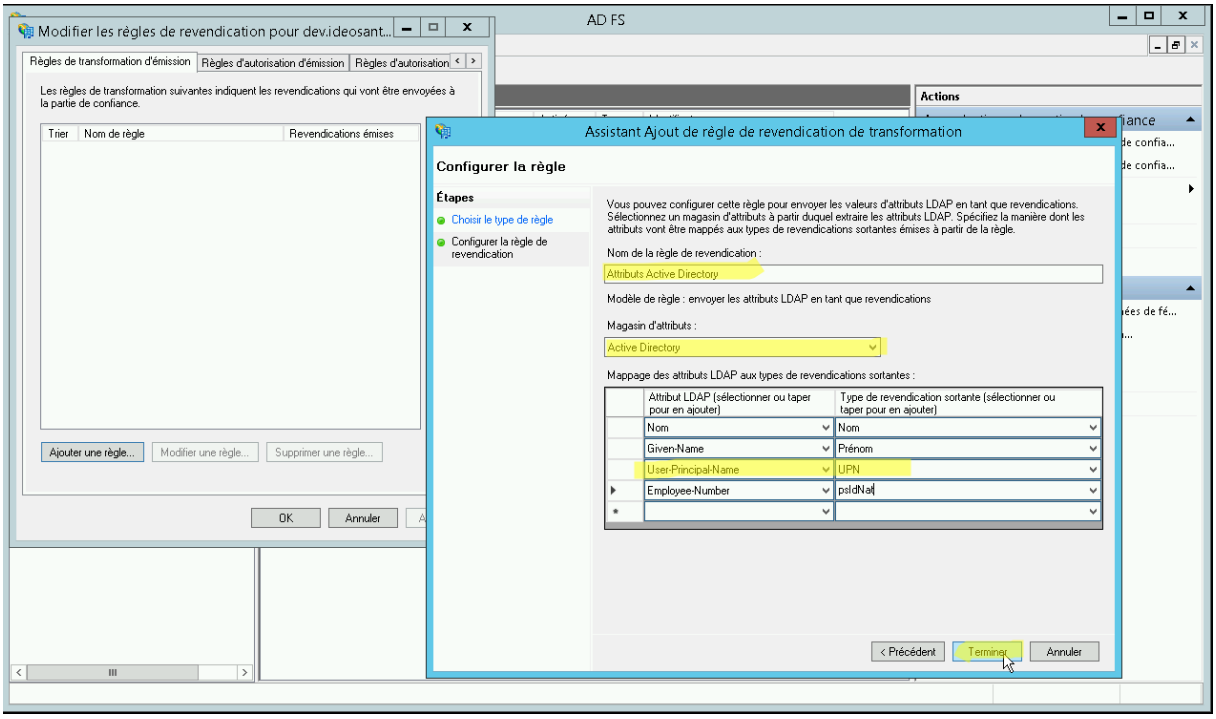


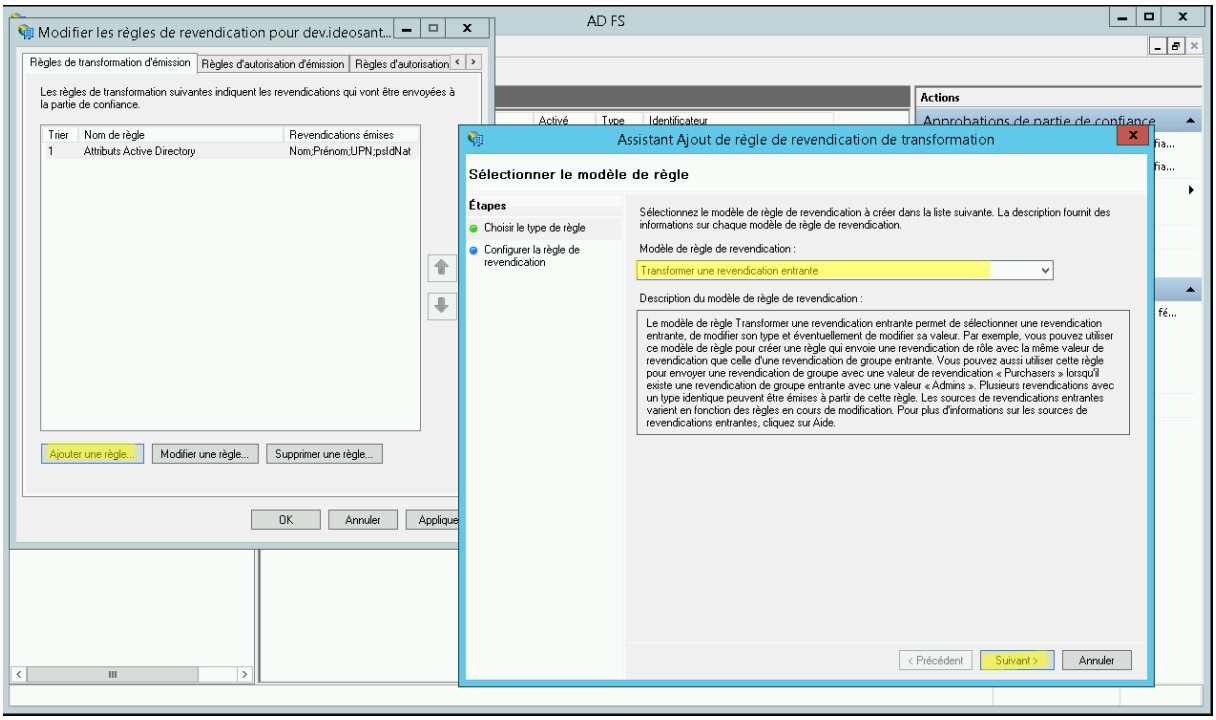


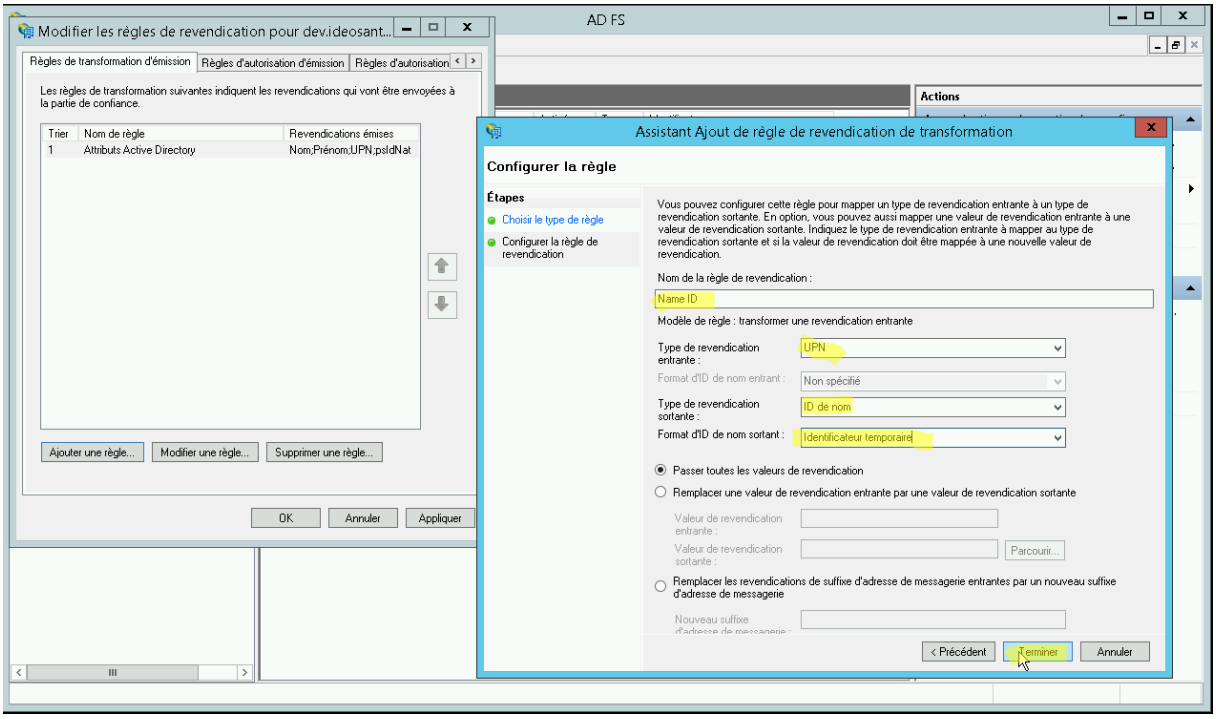


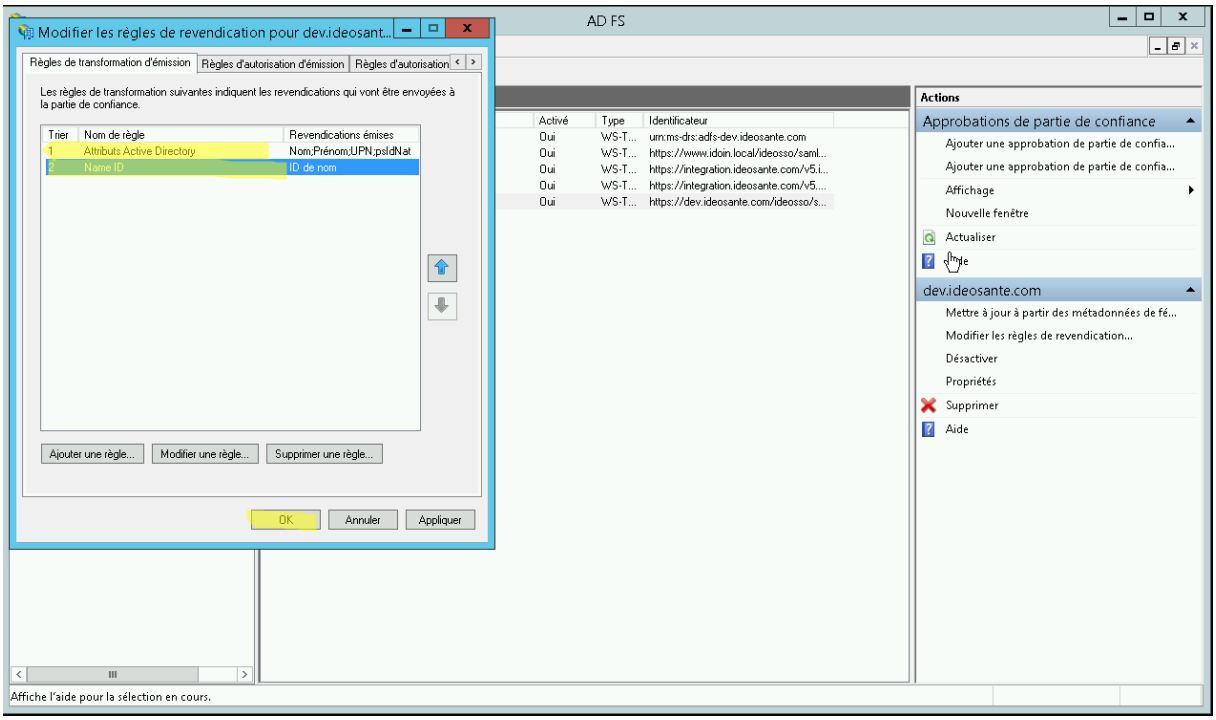


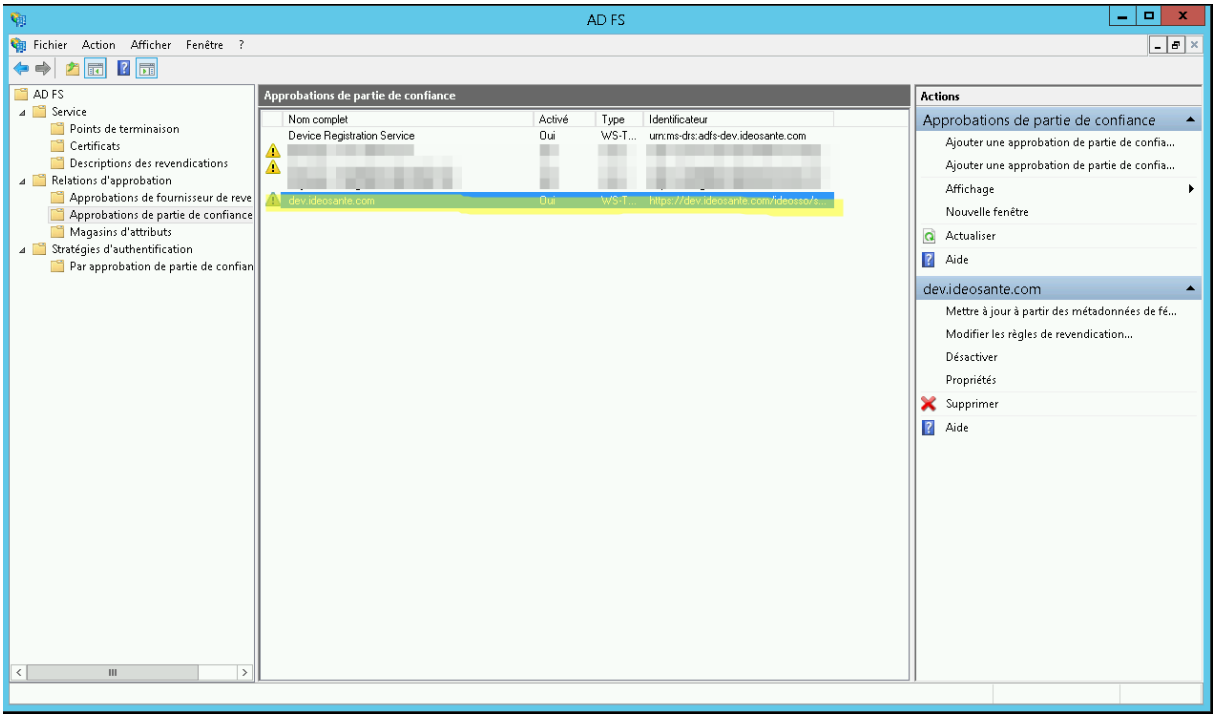












7. Autres Annexes

7.1. Liens Web

Les liens suivants présentent le serveur CAS officiel et différents clients CAS :

- APEREO : Le site officiel du serveur CAS du projet APEREO : https://www.apereo.org/projects/cas
- JASIG, EN : L'ancien site officiel du serveur CAS du projet JASIG de Princeton : http://www.jasig.org/cas
- YALE, EN : Le site originel de CAS de Yale : http://tp.its.yale.edu/tiki/tiki-index.php?page=CentralAuthenticationService
- ESUP-PORTAIL, FR : Un site français sur CAS pour l'intégration dans ESUP-PORTAIL : http://www.esup-portail.org/consortium/espace/SSO_1B/index.html
- JASIG Wiki, EN : Le wiki contenant toute la documentation officielle : https://wiki.jasig.org/display/CASC/Home