AGENCE
DU NUMÉRIQUE
EN SANTÉ

Projet plateforme numérique SAS

Spécifications techniques d'interopérabilité

INT_SSO

Délégation d'authentification par le SAS

Agence du Numérique en Santé
9, rue Georges Pitard – 75015 Paris
T. 01 58 45 32 50
esante.gouv.fr

Sommaire

Page de révisions

Historique des versions

Version	Date	Auteur	Description des modifications
V1.0	24/01/2022	ANS	Version initialisée
V1.1	25/05/2022	ANS	Mise au format ANS
V2.0	08/09/2022	ANS	Finalisation des spécifications suite à la concertation
V2.1	04/05/2023	ANS	Mise à jour des informations de l'environnement de recette
V2.2	09/06/2023	ANS	Mise à jour des URL ID Provider

Références associées

Référence*	Nom du livrable	Description
(1)	SAS_SPEC INT_R02_Gestion des comptes régulateurs	Spécifications techniques d'interopérabilité INT_R02 sur la gestion des comptes régulateurs

** La référence est utilisée par la suite dans ce document afin d'indiquer les renvois vers les documents identifiés ci-dessus.*

1. Objet du document

Le présent document constitue le livrable du contrat d'interfaçage relatif à la délégation d'authentification par le SAS. Il a pour objectif de décrire et apporter les éléments nécessaires à l'accompagnement des éditeurs pour l'implémentation du flux.

Le chapitre 2 présente le cas d'usage

Le chapitre 3 est consacré à l'implémentation du flux

Le chapitre 4 propose une FAQ liée au flux mis en place

2. Contexte

2.1. Description du flux

L'objectif de cette interface, flux **INT_SSO**, est de permettre à l'utilisateur de ne pas avoir à se réauthentifier d'une solution logicielle à l'autre et ainsi fluidifier le parcours utilisateur. Le but étant pour les solutions logicielles éditeurs de déléguer l'authentification à la plateforme numérique SAS.

Le schéma de présentation générale ci-dessous illustre le cas d'usage :

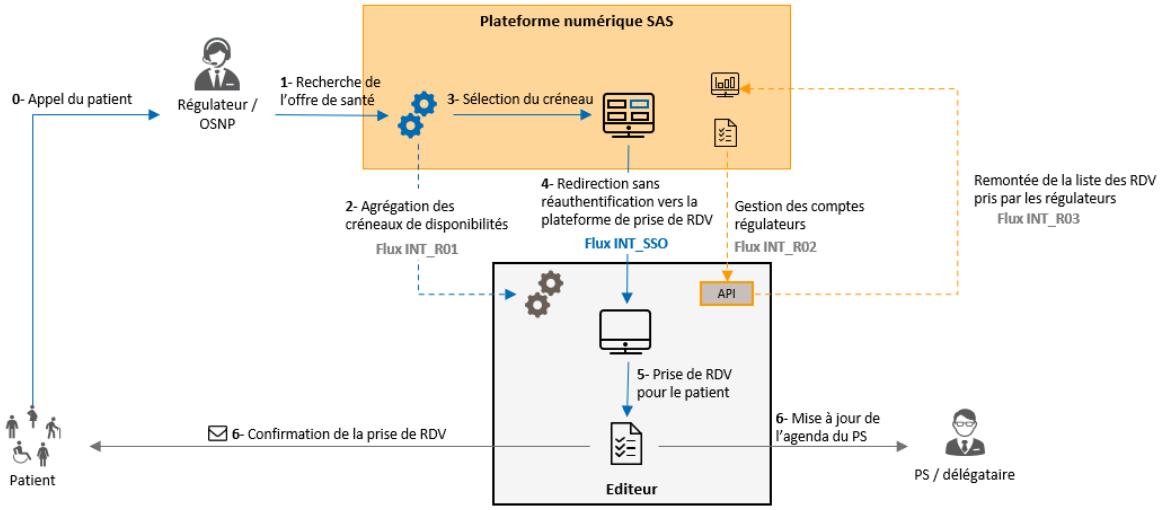


Schéma de représentation globale du parcours

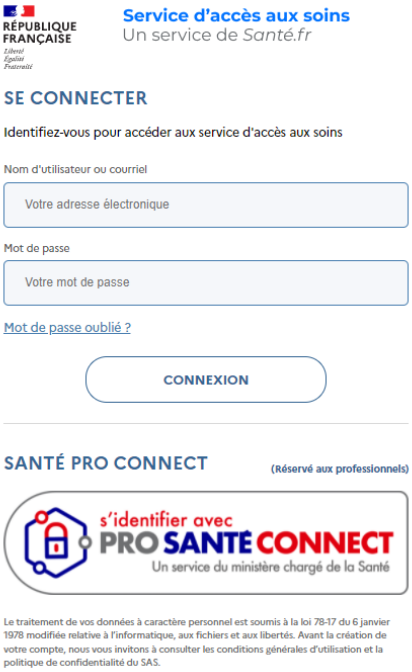
Après avoir agrégé les créneaux de disponibilités à la suite d'une recherche d'offre de santé, le régulateur sélectionne un créneau et est redirigé vers la plateforme de prise de RDV éditeur sans avoir à se réauthentifier. Dans la solution logicielle éditeur, le régulateur va pouvoir ensuite prendre RDV pour le patient.

Au moment de la redirection, la solution logicielle éditeur va contrôler la provenance de l'utilisateur (paramètre « origin » ajouté à l'URL de redirection) et si celui-ci n'est pas authentifié, déléguer l'authentification à la plateforme numérique SAS. En lien avec le flux INT_R02 sur la gestion des comptes régulateurs, la solution logicielle éditeur contrôlera et donnera les droits et habilitations correspondants à l'utilisateur.

Pour la mise en place de ce flux, il est nécessaire de s'assurer d'une technologie commune aux différentes plateformes. Les échangent reposent sur le protocole OpenID Connect (OIDC).

2.2. Méthodes de connexion à la plateforme numérique SAS

Les utilisateurs de la plateforme numérique SAS ont la possibilité de se connecter de deux manières différentes soit par login / mot de passe soit par Pro Santé Connect :



Page d'authentification de la plateforme numérique SAS

Méthodes	Description
Login / mot de passe	Complexité : 12 caractères minimum avec au moins 1 minuscule, 1 majuscule, 1 chiffre et 1 caractère spécial
Pro Santé Connect	Délégation d'authentification à Pro Santé Connect Authentification forte : • Connexion par carte CPx ou e-CPS • Protocole OpenID Connect

Point d'information : Des travaux sont en cours afin d'accompagner le déploiement de l'équipement nécessaire (lecteur de carte, cartes, etc.) aux régulateurs pour les inciter à adopter l'usage de ProSanté Connect. Une attention particulière est portée sur les modalités de connexions à la plateforme numérique SAS afin d'en renforcer la sécurité.

MISE EN PLACE DE LA DELEGATION D'AUTHENTIFICATION

Nous proposons de détailler dans cette section les spécifications nécessaires afin d'accompagner les éditeurs à l'implémentation du flux.

2.3. Introduction

2.3.1. Protocole OpenID Connect

OpenID Connect est une couche d'identification s'appuyant sur OAuth 2.0. Ce protocole permet aux clients de vérifier l'identité de l'utilisateur final sur la base de l'authentification effectuée par un serveur d'autorisation, ainsi que d'obtenir des informations sur l'utilisateur final d'une manière interopérable et de type REST.

OpenID Connect permet aux clients de tous types, y compris les clients Web, mobiles et JavaScript, de demander et de recevoir des informations de manière sécurisée sur les sessions authentifiées et les utilisateurs finaux.

La documentation OpenID Connect est consultable à ce lien : https://openid.net/connect/.

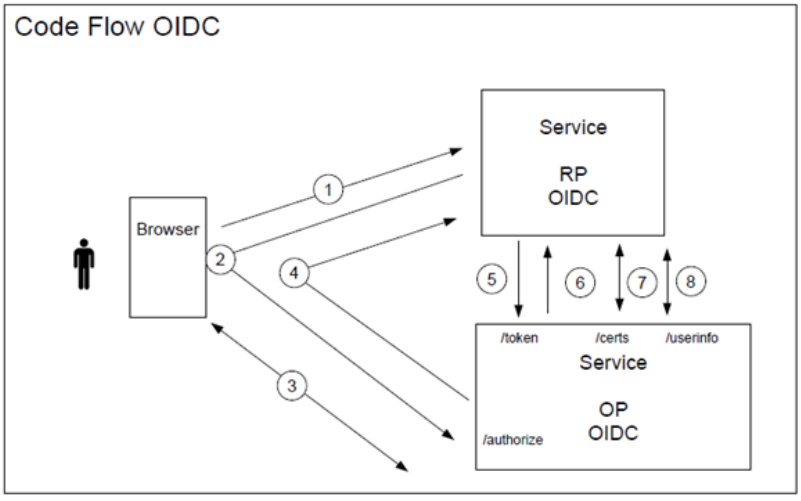
2.3.2. OpenID Connect « Authorization code flow »

Dans le cadre des travaux avec la plateforme numérique SAS pour la mise en place du SSO par délégation d'authentification, nous exploiterons l'algorithme « **Authorization code flow** » décrit ci-dessous :

- Les tokens sont retournés uniquement par l'interface token,
- La récupération d'un token d'accès s'effectue en deux étapes :
  - Un code est retourné par l'interface d'autorisation,
  - Ce code est envoyé par le client à l'interface token.
- Le client doit être enregistré auprès du fournisseur OpenID (via un identifiant et un secret)

Ce flux prend en compte les échanges entre 3 acteurs, le navigateur (Browser ou Smartphone), le client (RP OIDC, correspondant à la solution logicielle éditeur) et le serveur (OP OIDC, correspondant au SAS).

Le schéma ci-dessous illustre le concept :



Authorization code flow

L'identifiant de l'utilisateur et son mot de passe ne sont pas fournis au RP.

1. Appel au service

2. Demande d'un code pour un accès au scope openid par le client via un redirect sur l'uri « authorize » (grant_type=code)
3. Authentification de l'utilisateur
4. Fourniture du code au client via un redirect sur l'uri de redirection du RP.
5. Demande de ID Token et de Access Token par le client en fournissant le code, le scope, l'identifiant et la méthode d'authentification du service sur l'uri « token ».
6. Fourniture de l'ID Token et de l'Access Token par l'OP au RP
7. Vérification de la signature de ID Token par le RP
8. Utilisation de l'Access Token pour l'accès à l'URL « userinfo ». Parfois l'ID Token suffit à fournir les informations utilisateur, cependant il peut être nécessaire de faire une requête sur l'URI « /userinfo » avec l'Access Token pour obtenir ces informations.

2.4. Description des flux

Les endpoint exposés sont des standards du protocole OpenID Connect et sont décrits ci-dessous.

2.4.1. Diagramme de séquence

Nous détaillons ci-dessous le diagramme de séquence contextualisé au projet SAS.

Lorsque l'utilisateur a réalisé une recherche d'offre de soins, il sélectionne ensuite un créneau qui va le rediriger vers la solution logicielle éditeur correspondante :

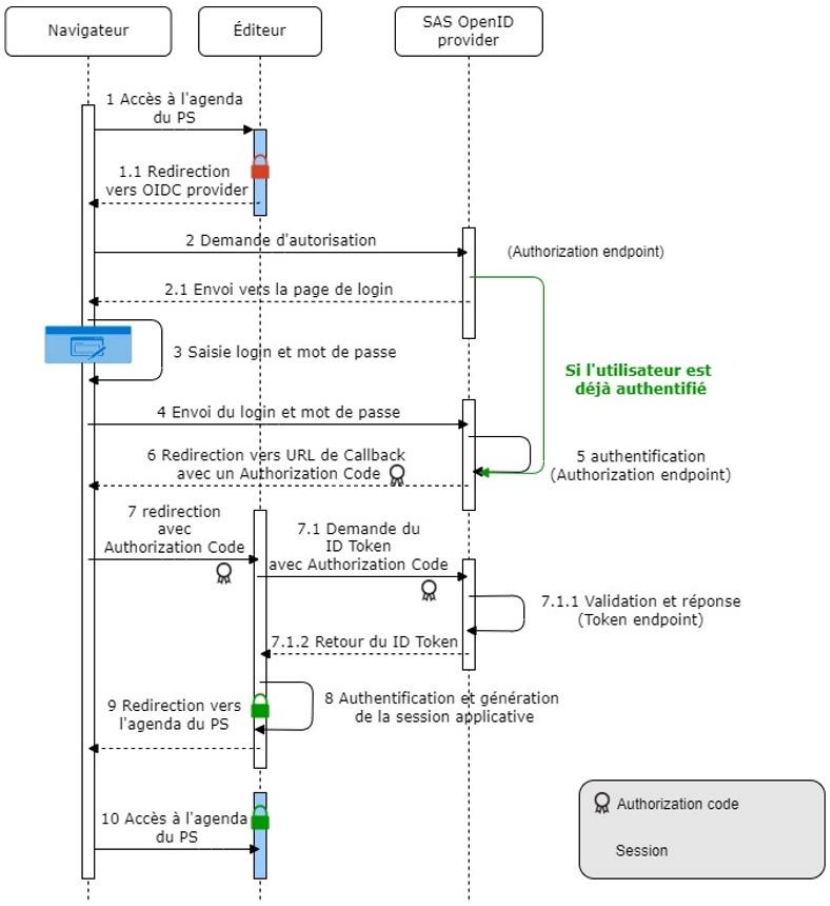


Diagramme de séquence

On prendra l'hypothèse que l'utilisateur est déjà authentifié à la plateforme numérique SAS :
1 - L'utilisateur est authentifié à la plateforme numérique SAS et est redirigé vers la solution logicielle éditeur (client)
1.1 - L'éditeur détecte la provenance de l'utilisateur grâce au paramètre « origin », identifie qu'il n'est pas authentifié et le redirige dans une séquence d'authentification avec le SAS
2 - Une demande de Authorization code est envoyée au SAS (cf. chapitre 3.2.2)
…
5 et 6 – Le serveur d'authentification SAS contrôle l'identité de l'utilisateur, génère un code d'autorisation et redirige l'utilisateur vers la solution logicielle éditeur (URL de Callback)
7 – Le code d'autorisation est récupéré par le client, qui va ensuite faire la demande d'un ID token et Acces token (cf. chapitre 3.2.3) au serveur d'authentification SAS. Le serveur vérifie également l'identité du client.
8 – Le client va contrôler grâce aux informations du ID token l'identité de l'utilisateur afin de lui générer une session avec les droits et habilitations correspondants (cf. chapitre 3.2.4 et spécifications du flux INT_R02 sur la gestion des comptes régulateurs ***(1)***) pour la solution logicielle éditeur
9 et 10 – L'utilisateur est connecté et est redirigé vers l'agenda du PS où il pourra prendre RDV pour le patient

2.4.2. Authorization code endpoint

Cette section décrit l'échange serveur / client pour la récupération du « Authorization code »

Paramètres de la requête client :

Paramètres	Description
endpoint	${URL serveur authentification}/realms/sas/protocol/openid-connect/auth
méthode	GET
client_id	${Éditeur}, identifiant du client
response_type	"code"
scope	"openid interop_editor", permet de définir les informations que l'on souhaite récupérer
redirect_uri	${callback uri}, URL vers laquelle l'utilisateur est redirigé à la suite de l'authentification
state	Valeur générée aléatoirement par le client, renvoyée telle quelle dans URL de callback pour être vérifiée par le serveur d'authentification SAS. *Note : ce champ permet de se prémunir contre les attaques CSRF. Cf. documentation OpenID Connect https://auth0.com/docs/secure/attack-protection/state-parameters*

Réponse du serveur d'authentification :

L'utilisateur s'authentifie, dans le cas où il ne l'était pas, puis le serveur d'authentification le redirige vers l'URL de callback transmise par l'éditeur (${callback uri}) avec le Authorization code généré.

Paramètres	Description
state	Valeur générée aléatoirement par le client, renvoyée telle quelle dans URL de callback pour être vérifiée par le serveur d'authentification SAS. *Note : ce champ permet de se prémunir contre les attaques CSRF. Cf. documentation OpenID Connect https://auth0.com/docs/secure/attack-protection/state-parameters*
session_state	Valeur représentant l'état de connexion de l'utilisateur sur le serveur

	https://openid.net/specs/openid-connect-session-1_0.html#CreatingUpdatingSessions
code	Authorization code

Exemple :

Requête client :

GET ${URL serveur authentication}/realms/sas/protocol/openid-connect/auth?client_id=${Éditeur}>&response_type=code&scope=openid interop_editor&redirect_uri=${callback uri}&state=**********

Réponse serveur :

https://${callback uri}?state=*****&session_state=****&code=*******

2.4.3. ID token et Access token endpoint

Cette section décrit l'échange serveur / client pour la récupération des « ID token » et « Access token » à partir du « Authorization code » récupéré (il pourra être réutilisé pour renouveler les tokens à leur expiration).

Dans le cadre des travaux menés avec la plateforme numérique SAS, nous exploiterons uniquement le « ID token ».

Le « Access token » peut être utilisé pour la récupération d'information sur l'utilisateur via le endpoint /userinfo. Nous n'utiliserons pas cette fonctionnalité.

Paramètres de la requête client :

Paramètres	Description
endpoint	${URL serveur authentification}/realms/sas/protocol/openid-connect/token
méthode	POST
header HTTP	Content-Type = application/x-www-form-urlencoded
client_id	${Éditeur}, identifiant du client
client_secret	${Éditeur secret}, secret du client
grant_type	"authorization_code"
code	code reçu du Authorization code endpoint
redirect_uri	${callback uri}, URL vers laquelle l'utilisateur est redirigé à la suite de l'authentification
scope	"openid interop_editor"

Réponse du serveur d'authentification :

Lorsque l'appel est réussi, la réponse est transmise au format JSON avec un code retour HTTP 200.

Paramètres	Description
id_token	Jeton d'identité sécurisé délivré par le serveur contenant les informations d'authentification de l'utilisateur. Il est représenté comme un jeton JWT
token_type	"Bearer"
session_state	Valeur représentant l'état de connexion de l'utilisateur sur le serveur *https://openid.net/specs/openid-connect-session-1_0.html#CreatingUpdatingSessions*
scope	"openid interop_editor", permet de définir les informations que l'on souhaite récupérer

Exemple :

Requête client :

```
curl --location --request POST '${URL serveur authentification}/realms/sas/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id=${Éditeur}' \
--data-urlencode 'client_secret=${Éditeur secret} \
--data-urlencode 'grant_type=authorization_code' \
--data-urlencode 'code=**********' \
--data-urlencode 'redirect_uri=${callback uri}' \
--data-urlencode 'scope=openid interop_editor'
```

Réponse serveur :

```
{
  "access_token": "<JWT SIGNED TOKEN HS256>",
  "expires_in": 300,
  "refresh_token": "<JWT SIGNED TOKEN HS256>",
  "refresh_expires_in": 1800,
  "token_type": "Bearer",
  "id_token": "<JWT SIGNED TOKEN HS256>",
  "not-before-policy": 1642408949,
  "session_state": "a1a9c592-c1de-4322-b105-202c8496f578",
  "scope": "openid interop_editor"
}
```

2.4.4. ID token et réconciliation d'identité

Nous nous intéresserons particulièrement au paramètre « **preferred_username** » du ID token qui correspond à l'email de l'utilisateur et est la donnée permettant de réaliser la réconciliation d'identité avec le compte utilisateur dans la solution logicielle éditeur.

La solution logicielle éditeur contrôle l'identité et génère la session applicative de l'utilisateur avec les droits et habilitations correspondants.

Paramètres	Description
preferred_username	Valeur correspondant à l'email de l'utilisateur
idp_connect	Valeur correspondant à la méthode d'authentification de l'utilisateur à la plateforme numérique SAS : • soit le champ est absent ou vide : connexion de l'utilisateur via login/mdp

	• soit le champ est valorisé à « psc » : connexion de l'utilisateur via ProSanté Connect
iss	Identité de l'émetteur du jeton. URL du endpoint de jetons du serveur d'authentification
sub	Identifiant technique de l'utilisateur final
aud	Ce champ contient la liste des identifiants des systèmes cibles pour lesquels le jeton a été fourni. La liste doit contenir au moins une entrée.
exp	Heure d'expiration du jeton (au format epoch)
iat	Heure d'émission du jeton (« issued at » au format epoch)
jti	Identifiant unique du jeton permettant de révoquer le jeton et empêcher le rejeu

Exemple de ID token décodé :


```
{
  "exp": 1642409914,
  "iat": 1642409614,
  "auth_time": 0,
  "jti": "e212e819-7e58-41f1-821e-2485472c332e",
  "iss": "${URL serveur authentification}/realms/sas",
  "aud": "SAS",
  "sub": "474f88bc-6193-4b60-86a6-2f454af30972",
  "typ": "ID",
  "azp": "SAS",
  "session_state": "a1a9c592-c1de-4322-b105-202c8406f578",
  "at_hash": "tdTtVMJ76VO_ht2AmkiUcQ",
  "acr": "1",
  "sid": "a1a9c592-c1de-4322-b105-202c9496f578",
  "email_verified": false,
  "preferred_username": "<EMAIL UTILISATEUR>",
  "idp_connect": "psc",
}
```


2.5. Référencement d'un éditeur

2.5.1. Environnements ANS

Dans le cadre de la recette connectée avec la plateforme numérique SAS, nous avons trois environnements qui sont mis à disposition : Recette, Préproduction (PPRD), Production (PROD).

Environnements	URL serveur authentification
Recette	https://seconnecter.integration.santefr.esante.gouv.fr/
Préproduction	https://seconnecter.preproduction.santefr.esante.gouv.fr/
Production	https://seconnecter.sante.fr/

Dans le présent document, **${URL serveur authentification}** est à remplacer par l'une des lignes du tableau ci-dessus en fonction de l'environnement concerné.

En fonction du nombre d'environnements identifié côté éditeur nous pouvons avoir les configurations suivantes :

- Soit en raccordement un pour un si l'éditeur propose également trois environnements :
  - Recette <> Recette
  - PPRD <> PPRD
  - PROD <> PROD
- Soit en raccordement simplifié si l'éditeur propose deux environnements :
  - Recette ANS <> PPRD éditeur
  - PPRD ANS <> PPRD éditeur
  - PROD <> PRPOD

Afin de pouvoir identifier les environnements et la provenance des appels, nous ajoutons un paramètre « origin » à l'URL de redirection :

- Recette : origin=sas-integration
- PPRD : origin=sas-preprod
- PROD : origin=sas-prod

2.5.2. Travaux d'implémentation éditeur

Chaque client (éditeur) doit être référencé sur le serveur d'authentification SAS. La pile logicielle utilisée est Keycloak. La configuration mise en place côté serveur est la suivante :

Paramètres	Description
client ID	${Éditeur}, identifiant du client
client_secret	${Éditeur secret}, secret du client. Génération d'une clé unique servant de mot de passe pour l'authentification du client au serveur.
enabled	"ON", activation du client
client protocol	"openid-connect", protocole utilisé
access type	"confidential", correspond au niveau de sécurité d'accès attendu pour le client
valid Redirect URIs	${callback uri}, liste d'urls valident pour les redirect_uri
base URL	URL du client

Pour chaque éditeur, les informations suivantes sont à nous transmettre pour configurer le client :

- **Nom du client** : libellé représentant la solution logicielle éditeur
- **Adresses IP publiques** : liste des adresses IP publiques par environnement qui auront l'autorisation d'exploiter les endpoints du serveur d'authentification si applicable
- **URL de callback** (${callback uri}) : URL par environnement vers laquelle le serveur renvoi à la suite d'une authentification réussie

La plateforme numérique SAS fournira en retour les éléments **« client ID »** (${Éditeur}) et **« client secret »** (${Éditeur secret}) propres à chaque éditeur.

L'éditeur devra implémenter le protocole OpenID Connect sur son application et configurer les appels aux différents endpoint en se référant aux éléments partagés dans les parties précédentes :

Paramètres	Description
authorization	https://${URL serveur authentification}/realms/sas/protocol/openid-connect/auth

token	https://${URL serveur authentification}/realms/sas/protocol/openid-connect/token

Sécurisation des échanges : l'ensemble des transactions sont sécurisées par l'utilisation du protocole HTTPS sur le client et sur l'OIDC Provider.

Modalités de délégation d'authentification éditeur : L'éditeur, en tant que responsable est sujet des exigences imposées par le RIE, a le choix d'accepter ou de refuser la délégation d'authentification pour les utilisateurs connectés via login/mdp à la plateforme numérique SAS s'il juge que le niveau de sécurité est trop faible. Il s'agit d'une phase transitoire, à l'issue de laquelle il est prévu de renforcer la sécurisation des connexions et inciter l'utilisation de ProSanté Connect.

Le champs « idp_connect » transmis au sein de l'ID token permet à l'éditeur d'identifier le mode de connexion à la plateforme numérique SAS utilisé par l'utilisateur et de mettre en place s'il le juge nécessaire un filtrage pour refuser la délégation aux utilisateurs connectés via login/mdp.

3. FAQ

Cette section regroupe les réponses aux questions les plus fréquemment posées au cours des travaux de développements menés par les éditeurs, et les tests d'intégration.

Les redirects URLs dans /auth et /token doivent-elles être identiques ?
Il est préconisé de paramétrer la même redirect URI pour les redirections vers « /auth » et « /token » sous peine d'obtenir cette erreur :

```
400 Bad Request: invalid_grant - Incorrect redirect_uri
```

Sur quel champ se baser pour identifier l'utilisateur après réception du token JWT (ID token) ?
Dans le token JWT (ID token), c'est la donnée « preferred_username » qui correspond à l'email du régulateur sur laquelle il faut s'appuyer.

Quelle page de redirection est attendue pour un utilisateur qui n'a pas pu être authentifié (erreur, non habilité) ?
Il est attendu de rediriger un utilisateur non authentifié vers la page de connexion de la solution logicielle éditeur.