**Couplage d’une application avec le système d’authentification**

**« openID Connect»**

**1 Introduction et contexte**

Un système d'information (SI) est un ensemble de ressources et de processus permettant de gérer l'information au sein d'une organisation. Il fournit aux utilisateurs une large gamme de services informatiques, notamment des applications métier, des services de communication et de collaboration et des outils de gestion de projet. Certains de ces services nécessitent une authentification par proxy pour vérifier l'accès attribué.

L'objet de cette annexe est de présenter la solution d'authentification proposée pour ces services, et les limitations techniques d'interface qu'elle implique. Il porte exclusivement sur les services fournis aux utilisateurs ou partenaires identifiés. Les services fournis aux utilisateurs, tels que les services de support, reposent sur un autre système d'authentification et de gestion des droits, qui n'est pas couvert par cette annexe.

Les services concernés peuvent être des services délivrés depuis l'infrastructure interne du SI, tels que des applications métiers, des services de stockage ou des outils de collaboration. Il peut également s'agir de services du modèle SaaS (Software as a Service), fournis par des tiers, tels que des services de messagerie, de gestion de la relation client ou de gestion de projet.

Dans tous les cas, le système d'authentification proposé doit permettre aux utilisateurs du SI d'utiliser les services avec une authentification unique sans avoir à se reconnecter à chaque connexion (SSO - Single Sign On). Le système doit également respecter les politiques de sécurité et de gestion des identités en vigueur dans l'organisation, en tenant compte des éventuelles limitations d'intégration avec d'autres systèmes existants.

Image représentant la SSO


**2 Références normatives**

La solution d'authentification proposée est basée sur la norme OpenID Connect, qui est basée sur le mécanisme d'authentification OAuth 2.0 (RFS 6749).

Nous allons donc définir la notion « d’OpenID Connect ».

OpenID Connect fournit une fine couche d'identité qui permet aux clients de valider l'identité d'un utilisateur final à l'aide de l'authentification fournie par un serveur d'autorisation. Cette architecture REST interopérable utilise JSON comme format de données pour fournir des informations de session authentifiées et d'utilisateur final à divers clients tels que des applications Web, mobiles et JavaScript. La spécification OpenID Connect est extensible et prend en charge des fonctionnalités facultatives telles que le chiffrement des données d'identité, la découverte dynamique des fournisseurs OpenID et la gestion des sessions.Contrairement à l'attribution d'une clé, la solution d'authentification proposée attribue un badge à l'utilisateur, contenant les informations d'authentification ainsi que les autorisations spécifiques associées.

Le service d'authentification (IP) est responsable de la délégation de l'authentification auprès de ce service. Le titulaire doit se connecter à ce service d'authentification pour accéder aux services qui nécessitent une authentification.

**3 Eléments de principes généraux**

Dans le contexte du standard OpenID Connect, les éléments suivants sont précisés :

* Service Provider (SP) : Le SP représente le service proposé par le titulaire. Ce service peut être proposé en mode SaaS (Software as a Service), où le service est délivré par un tiers, ou en mode hébergé, où le service est hébergé sur l'infrastructure interne du titulaire. (WAMPServer)
* Identity Provider (IP) : L'IP est le service d'authentification mis en place par le titulaire. Il s'agit du service qui permet de vérifier l'identité de l'utilisateur final et de délivrer les informations d'authentification nécessaires pour accéder aux services proposés par le SP. L'IP est responsable de la délégation de l'authentification auprès de ce service et doit être conforme aux politiques de sécurité et de gestion des identités en vigueur dans l'organisation.(Google)
* Client : Le client est une application ou un service qui accède au service proposé par le titulaire en utilisant les informations d'authentification délivrées par l'IP. Le client peut être une application Web, mobile ou une application native. (Php storm)
* Utilisateur final : L'utilisateur final est l'utilisateur du service proposé par le titulaire. L'utilisateur final se connecte au service en utilisant les informations d'authentification délivrées par l'IP.page1image164793600

**4 Rappel sur les mécanismes d’authentification délégués en OpenID Connect**

Ce chapitre a pour objet de rappeler les principes généraux de l’authentification OpenID Connect.

**4.1 Séquence d’authentification**

De manière synthétique le schéma suivant récapitule la séquence d’authentification d’un utilisateur :

Une image contenant diagramme

Description générée automatiquement

**Phase 1 :** Lorsqu'un utilisateur envoie une requête au Service Provider (SP) sans code d'identification valide ou avec un token obsolète, le SP redirige l'utilisateur vers l'Identity Provider (IP) pour l'authentification.

**Phase 2 :** L'utilisateur fournit les éléments d'identification requis par l'IP, tels que les identifiants, l'authentification multi-facteurs, etc. Si l'IP vérifie avec succès ces informations, il envoie un code d'autorisation à l'utilisateur, qui le transmet ensuite au SP dans sa requête.

**Phase 3 :** L'utilisateur soumet une nouvelle requête au SP, cette fois avec le code d'autorisation obtenu auprès de l'IP.

**Phase 4 :** Le SP envoie une requête à l'IP, incluant le code d'autorisation de l'utilisateur et les identifiants de l'application (client\_id et client\_secret), pour obtenir un token d'accès. Ce token contient des informations sur le profil et l'email de l'utilisateur (scope "profile" et "email"). Le SP vérifie la validité du token et sa signature en utilisant la clé publique de l'IP.

**Phase 5 :** Si le token est valide et conforme, le SP vérifie les autorisations de l'utilisateur et fournit les ressources demandées.

Afin de mieux comprendre, nous pourrions prendre un nouvel exemple.

Nous avons un Grant flow avec :

* Le client
* L’autorisation serveur
* Le Resource Owner
* Les ressources

Ici , nous avons aussi différentes phases.

1. Le client envoie une requête d’autorisation au serveur d’autorisation
2. L’autorisation serveur va demander à la resource owner son consentement. « Est-ce que j’autorise le client à accéder à tes données ? ».
3. (Si oui) L’autorisation serveur se redirige vers le client avec un code
4. Le client récupère l’autorisation code et fait une nouvelle requête à l’autorisation serveur
5. (Si oui) Il lui renvoie un Access Token (et possiblement un Refresh Token).
6. Le client accède aux ressources

**4.2 Découverte des services de l’IP et abonnement au service**

Cette phase de fonctionnement avec l’IP nécessite d’avoir identifié́ les services de l’IP (découverte des services de l’IP) :

* URL d’authentification proposée par l’IP : mire de login vers laquelle le SP renvoie le client lorsque le code est absent ou invalide
* Le « token endpoint » qui permet au SP de récupérer le token (JWS respectant le RFS 7519) d’un utilisateur avec le code qu’il a fourni.
* Le jeu de clefs publiques qui permet de vérifier la signature par l’IP des tokens JWS qui lui seront confiés.

**4.3 Gestion des habilitations**

Les habilitations peuvent être gérées de deux manières différentes:

*4.3.1* *Gestion autonome par l’application des habilitations sur un pivot « profil »*

L’IP fournit uniquement un service d’authentification. C’est le SP qui est en charge au regard de son paramétrage de gérer les profils d’habilitation de chaque utilisateur. Dans ce contexte, les scopes « profil » et « email » peuvent être utilisés pour assurer la correspondance entre un profil de droits affecté par l’application et un utilisateur.

Les appels à l’IP relèvent donc du fonctionnement « standard claims » c’est-à-dire sur les scopes prédéfinis par le standard openID Connect.

Ce fonctionnement est le fonctionnement privilégié.

*4.3.2* *Habilitations délivrées par l’IP*

Le SP récupère les profils de l’utilisateur au travers d’une requête « User Info Request » sur l’URL dédiée à cette fonction par l’IP. La gestion de l’affectation d’un profil à l’utilisateur est alors déléguée à l’IP.

Les appels à l’IP relèvent donc du fonctionnement « custom claims » c’est-à-dire sur des scopes spécifiques à définir entre l’IP et le SP.

Ce fonctionnement n’est pas privilégié dans un premier temps pour les applications connectées à l’IP. Le fonctionnement défini au §4.3.1 basé sur les « standard claims » doit être privilégié.

La définition des rôles relève de la responsabilité de l’application.

**5 Contraintes techniques à respecter**

**5.1 Chiffrement des échanges**

Toutes les communications entre le client, le SP et l’IP sont chiffrées conformément aux exigences Oauth2 du RFC6749. Ces communications doivent être conformes au protocole TLS dans sa version 1.2 au minimum.

**5.2** **Actualisation des clefs publiques de l’IP**

Le SP doit s’assurer d’une redécouverte régulière des services de l’IP pour garantir en particulier le rafraîchissement du jeu de clefs publiques de l’IP.

Ces données doivent être actualisées au minimum une fois par jour.

**5.3** **Mode de fonctionnement autorisé avec l’IP**

Pour des raisons de sécurité, seul le mode « Authorization Code Flow » est supporté avec l’IP. Les tokens fournis par l’IP sont conservés par le SP et non communiqués à l’utilisateur.

Les modes « Implicit Flow » et « Hybrid Flow » ne sont pas supportés et ne doivent pas être utilisés par le SP.

**5.4** **Phase d’échange entre l’utilisateur et l’IP**

En aucun cas, le SP ne doit mémoriser les identifiants d’authentification de l’utilisateur pour réaliser directement la phase d’authentification.

De la même manière, le SP doit systématiquement renvoyer vers la page d’authentification de l’IP sans s’intercaler dans le circuit de communication entre l’utilisateur et l’IP.

**5.5** **Refresh token**

Le SP peut avoir recours à un « Refresh Token » pour renouveler un « Access Token ». Le SP est en charge de la conservation sécurisée du « Refresh Token » distribué par l’IP.