

Microsoft Power Platform

- [Identité managée & Fédération d'identité](#)
 - [Mise en œuvre de l'identité managée Power Platform \(V2\)](#)

Identité managée & Fédération d'identité

Mise en œuvre de l'identité managée Power Platform (V2)

□ 1. QUOI ? (De quoi s'agit-il ?)

C'est la mise en place d'une passerelle de confiance sécurisée (**Fédération d'Identité - FIC**) entre le cloud Power Platform (SaaS) et votre infrastructure Azure.

- **Pour faire simple** : Cela permet à un plug-in Dataverse développé par votre collègue de se connecter à vos ressources Azure (comme un Key Vault ou une API sur une VM) **sans jamais stocker ni utiliser de mot de passe, de secret d'application ou de fichier de certificat dans Azure**. Le plug-in "emprunte" une identité Azure de manière transparente.
-

□□ 2. POURQUOI ? (Quel est le but et le bénéfice ?)

- **Éliminer les risques** : Plus aucun "Client Secret" ne transite dans le code ou dans Dataverse. Vous supprimez le risque de fuite de mots de passe et la corvée de devoir les renouveler tous les ans dans le portail Azure.
- **Pourquoi la "Version 2" ?** La V1 plantait dès qu'un certificat contenait un accent ou une virgule. La V2 transforme tout en hachage standardisé (ASCII Base64URL), ce qui rend le système 100 % fiable.
- **Pourquoi une UAMI plutôt qu'une Inscription d'application ?** L'identité managée affectée par l'utilisateur (UAMI) est le standard recommandé par Microsoft. Elle n'a aucun secret à gérer, contrairement à une inscription d'application qui est plus lourde à maintenir. On ne choisit l'inscription d'application que si le plug-in doit se connecter à des abonnements Azure chez des clients externes (Multi-tenant).

☐☐ 3. QUI ? (Qui fait quoi ? Matrice des rôles)

Le travail est strictement séparé pour que chacun reste dans son domaine d'expertise :

- **Le Sysadmin (Vous) ☐☐ :**

- Vous créez l'identité (UAMI) dans Azure.
- Vous fournissez le certificat pour signer le code.
- Vous configurez le lien de fédération (FIC) dans le portail Azure.
- Vous attribuez les droits d'accès (RBAC) sur les ressources de production.

- **Le Développeur ☐☐ :**

- Il écrit le code du plug-in.
 - Il signe son code avec votre certificat.
 - Il enregistre son plug-in dans Dataverse et le lie à l'UAMI.
 - Il n'a aucun droit d'administration sur votre infrastructure Azure de production.
-

☐☐ 4. LE CERTIFICAT (Comment le créer et le fournir ?)

Le certificat ne sert pas à s'authentifier auprès du portail Azure (l'UAMI s'en charge sans secret). Il sert uniquement au développeur pour **signer son code** afin de prouver son identité. En entreprise, c'est au Sysadmin de le générer pour garder le contrôle des clés.

☐☐ Option A : Méthode rapide pour le Déploiement / Test (Script PowerShell)

À exécuter dans une console PowerShell en tant qu'administrateur pour générer un certificat auto-signé :

```
# 1. Génération du certificat de signature de code
$cert = New-SelfSignedCertificate `
    -Type "CodeSigningCert" `
```

```
-Subject "CN=Certificat-Plugin-PowerPlatform-Dev" `
-KeyExportPolicy "Exportable" `
-KeyLength 2048 `
-CertStoreLocation "cert:\CurrentUser\My" `
-NotAfter (Get-Date).AddYears(1)
```

2. Définition d'un mot de passe sécurisé pour le fichier PFX

```
\$password = ConvertTo-SecureString "MettreUnMotDePasseFortIci123!" -AsPlainText -Force
```

3. Exportation du certificat au format .pfx (contenant la clé privée)

```
\(cert \vert{ } Export-PfxCertificate -FilePath "C:\Certificats\PluginSignCert.pfx" -Password  
\)password
```

- **Livrable** : Vous transmettez le fichier `PluginSignCert.pfx` et son mot de passe au développeur pour qu'il l'injecte dans Visual Studio.

☐☐ Option B : Méthode pour la Production (PKI d'entreprise / ADCS)

1. Dupliquer le modèle par défaut **Signature de code** (*Code Signing*) sur votre serveur CA.
2. Dans l'onglet *Request Handling*, cocher impérativement **Allow private key to be exported** (Autoriser l'exportation de la clé privée).
3. Demander le certificat via la console `mmc` locale de votre poste de gestion.
4. Exporter le certificat obtenu au format **.PFX (inclure la clé privée)** avec un mot de passe fort, puis le transmettre au développeur.
5. *Rappel* : Notez bien la date d'expiration dans votre calendrier pour anticiper le renouvellement du fichier côté dev.

☐☐ 5. OÙ ? (Où est-ce que ça se configure et se gère ?)

Toute votre partie se gère dans le **Portail Azure** et dans **Microsoft Entra ID** :

1. **Création de l'UAMI** : Dans votre groupe de ressources Azure (comme une VM classique).
2. **Configuration du FIC** : Directement à l'intérieur de votre ressource UAMI, dans l'onglet *Informations d'identification fédérées*.

3. **Attribution des droits** : Sur la ressource finale (ex: dans l'onglet *Contrôle d'accès (IAM)* de votre Azure Key Vault).
 4. **Supervision et Logs** : Dans *Microsoft Entra ID > Applications d'entreprise* (en pensant à filtrer le type d'application sur *Identités managées*). C'est là que vous verrez les logs de connexion pour le troubleshooting.
-

⚙️ 6. COMMENT ? (Quelles sont les étapes clés pour vous ?)

Pour réussir le déploiement sans encombre, suivez ce fil conducteur :

1. **Créer l'UAMI** (via le portail ou en une ligne Azure CLI) et donner son **Client ID** au développeur.
2. **Fournir le certificat** (généré à l'Étape 4) au développeur pour qu'il signe son code.
3. **Récupérer les billes du développeur** une fois son code signé : il doit vous donner son *ID d'environnement Power Platform* et les *hachages de son certificat*.
4. **Créer le lien FIC** dans l'UAMI en renseignant scrupuleusement la chaîne d'identification V2 :
`/eid1/c/pub/t/{tenantId}/a/qzXoWDkuqUa3l6zM5mM0Rw/n/plugin/e/{environmentId}/i/{issuerHash}/s/{subjectHash}`
5. **Donner les droits RBAC** (ex: rôle *Utilisateur des secrets de Key Vault*) à l'UAMI sur votre ressource Azure.
6. **En cas de pépin (Erreur AADSTS700213)** : Pas de panique, cela signifie simplement que la longue chaîne textuelle du FIC au point n°4 contient une erreur de casse, un espace en trop ou un mauvais hachage. Il suffit de la vérifier avec le développeur.