Fournisseur d'identité OIDC

Générer et configurer les clés RSA du fournisseur

```
$ apt install python-jwcrypto
$ python
Python 2.7.8 (default, Oct 18 2014, 12:50:18)
[GCC 4.9.1] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from jwcrypto import jwk
>>> k = jwk.JWK.generate(kty='RSA', size=2048)
>>> s = jwk.JWKSet()
>>> s['keys'].add(k)
>>> s.export()
'{"keys":[{"d":"Y9GJV37oeFlqz2vuZZDsBVfu4u6YLGbMR7ONhedvn3YTMvFxzmeSwjHd2pdoY05TBjgNbDk6QaCKVtqRrH
sbbrmYN_6aRIoZk9KiZqLPvnrRUbe0_lLaozJxZjPa4urb8-vsq_Y040DhUEiBog0xjq2RDg7qtcpi5nf0NRNhXbEm2dzIutH2
2e8WDoUMma8b64NeZgciSmvdu24UUG_eoAt7fsy8xfL81nxA6qk0mWFrjyCpvxmjEotuCBQ79JLzStWeiKkBx80mokNnMTD3bZ
m-coJRXEPTcKzmtG_NTmNGi1_IZwZslr5cB6F3cyMIsA1y4mYgYu46JNVvUlwIYQ","dp":"OSokOYunlOqOFxOXuMNfdxTyxi
2itS5dW2M4S0PBzwlfOU-fD7JWft94sRTOXCP65No_vmw-V2AHnRRj-GgcHdjyIgHkMV4bgtgUkV7HtAmsC_VAdYwnEWOfeaq-
izl34DRUt-qN_m6HzNVziC2JtiPum2ifFuS2vrvO4FTPMgE","dq":"QwnEqx3W1MOa5dRhboSwPu4_J5vgwqofjeKrQD1OmEj
eowwzJmj5aYiRysZW6IzV5L7XpW081EdxDXI6ddBGWRq-QFlpF9hGxGYQ9qmmohz_ZcVuw5qUdoF113tUA_9aPujG3LPV7S9Jt
-R1piN8b7HGUkz_DSVMLzgArlQ4F8k","e":"AQAB","kty":"RSA","n":"yDIt_sOfAY7h5kcyQLimct2R-4PF5K6Fb90xtE
AQmZiXzfW2LzhUX4Uow-XLU_shMrAwixAqc-P8A_Sg-IpCHqvcEaIt0tylGThaguN6e6kJxgTU22Oqx4QBCgejm9xMW9kNf15O
vudBiuxn5vveR8VFts_pWU-wCNwBw1AHx67E6dszP0C2G7ZN_7v9AI3f3fTj9S1AGzaJHX5bu9aSRFkmVk-K_VBOyovJYYXb3r
hFoy16fJWGsxSTLHD9LS_OvpN3_EIj82ziTB2pEAhMAN4uuB8QmhYvevBin96TbECNMCuIbxenbZYyn1FslXWn29-UA03f8-ja
u5PubcyEDQ","p":"-Rh3WQ2i3ona_U4kKMR_XGYF6JXwQJAcfUlSfkx6VBI3mmW5uXzWWj8KFgRZEklhGoEq57GCyPWiRARgF
OdLRHGQzJidvZaNFqpWw96oFP5eDaOdmLmtthr59l6kUqfQgUwRF_QeHEOlgQrmXBG7-j2hNt0L-YdQIHO0OUEU0Fk","q":"z
b67UTECBXKtRfrDoyOvxOc1g7FcUSwGI15Qc_VXIG8ktRtJtj-6ZsnHymO3MXYE3L0ucjxmivj9ow6yVvj6C9uMLmo8AUNhzzF
6_FCgHTMERu7pNeRU5ArLMMJA-A5dcMyLCPnGCVFKxhMCEbVeAMs0DfJA6CW1Gk4E61GfOtU","qi":"d4R9BfJKsnOA3ZHpTX
tn7mR0uvxPK-mGiYVVLmk0Ko7OSCQxjzYscfle8L3d0iwTPXVVVhazT5ZUN-jOEzmtJ4XQxnfgYqdfxgAXIwducaoz4aptW3GO
WcwK9sK_q89IdV3HRQdeJhwUG4IJINBtC7QYvwE9FNhWvC9DsB5VAEo"}]}'
>>>
```

Recopier la chaîne contenant le document JSON dans le fichier /etc/authentic2/config.py ou le settings.json du tenant, comme ceci:

```
A2_IDP_OIDC_JWKSET = {"keys":[{"d":"Y9GJV37oeFlqz2vuZZDsBVfu4u6YLGbMR7ONhedvn3YTMvFxzmeSwjHd2pdoY0
5TBjgNbDk6QaCKVtqRrHsbbrmYN_6aRIoZk9KiZqLPvnrRUbe0_lLaozJxZjPa4urb8-vsq_Y040DhUEiBog0xjq2RDg7qtcpi
5nf0NRNhXbEm2dzIutH22e8WDoUMma8b64NeZgciSmvdu24UUG_eoAt7fsy8xfL81nxA6qk0mWFrjyCpvxmjEotuCBQ79JLzSt
WeiKkBx80mokNnMTD3bZm-coJRXEPTcKzmtG_NTmNGi1_IZwZslr5cB6F3cyMIsA1y4mYgYu46JNVvUlwIYQ","dp":"OSokOY
unlOqOFxOXuMNfdxTyxi2itS5dW2M4S0PBzwlfOU-fD7JWft94sRTOXCP65No_vmw-V2AHnRRj-GgcHdjyIgHkMV4bgtgUkV7H
tAmsC_VAdYwnEWOfeaq-izl34DRUt-qN_m6HzNVziC2JtiPum2ifFuS2vrvO4FTPMgE","dq":"QwnEqx3W1MOa5dRhboSwPu4
_J5vgwqofjeKrQD1OmEjeowwzJmj5aYiRysZW6IzV5L7XpW081EdxDXI6ddBGWRq-QFlpF9hGxGYQ9qmmohz_ZcVuw5qUdoF11
3tUA_9aPujG3LPV7S9Jt-R1piN8b7HGUkz_DSVMLzgArlQ4F8k","e":"AQAB","kty":"RSA","n":"yDIt_sOfAY7h5kcyQL
imct2R-4PF5K6Fb90xtEAQmZiXzfW2LzhUX4Uow-XLU_shMrAwixAqc-P8A_Sg-IpCHqvcEaIt0tylGThaguN6e6kJxgTU22Oq
x4QBCgejm9xMW9kNf15OvudBiuxn5vveR8VFts_pWU-wCNwBw1AHx67E6dszP0C2G7ZN_7v9AI3f3fTj9S1AGzaJHX5bu9aSRF
kmVk-K_VBOyovJYYXb3rhFoy16fJWGsxSTLHD9LS_OvpN3_EIj82ziTB2pEAhMAN4uuB8QmhYvevBin96TbECNMCuIbxenbZYy
n1FslXWn29-UA03f8-jau5PubcyEDQ","p":"-Rh3WQ2i3ona_U4kKMR_XGYF6JXwQJAcfUlSfkx6VBI3mmW5uXzWWj8KFgRZE
klhGoEq57GCyPWiRARgFOdLRHGQzJidvZaNFqpWw96oFP5eDaOdmLmtthr59l6kUqfQgUwRF_QeHEOlgQrmXBG7-j2hNt0L-Yd
QIHO0OUEU0Fk","q":"zb67UTECBXKtRfrDoyOvxOc1g7FcUSwGI15Qc_VXIG8ktRtJtj-6ZsnHymO3MXYE3L0ucjxmivj9ow6
yVvj6C9uMLmo8AUNhzzF6_FCgHTMERu7pNeRU5ArLMMJA-A5dcMyLCPnGCVFKxhMCEbVeAMs0DfJA6CW1Gk4E61GfOtU","qi"
:"d4R9BfJKsnOA3ZHpTXtn7mR0uvxPK-mGiYVVLmk0Ko7OSCQxjzYscfle8L3d0iwTPXVVVhazT5ZUN-jOEzmtJ4XQxnfgYqdf
xgAXIwducaoz4aptW3GOWcwK9sK_q89IdV3HRQdeJhwUG4IJINBtC7QYvwE9FNhWvC9DsB5VAEo"}]}
```

Ajouter un client OIDC

- Pointer votre navigateur sur http://idp/admin/authentic2_idp_oidc/oidcclient/
- Cliquer sur Ajouter un oidc client
- Choisir un identifiant et un identifiant court
- Modifier éventuellement le client_id généré pour utiliser l'identifiant court
- Choisir le schéma d'autorisation:
  - authorization code flow : schéma pour les applications web classique, le code d'authentification est récupéré directement par le serveur est résolu en access token

    - implicit flow : l'access token est directement passé à l'application sans résolution d'un code en s'authentifiant via le client-secret, adapté pour les applications mobiles ou pure JS-client,
- Définir les URIs de redirection, i.e. la liste des URLs sur lesquelles l'IdP redirigera l'utilisateur après l'authentification, attention ce doit être l'URL exacte passée dans le paramètre redirect_uri de la requête d'authentification aucune variation n'est permise (même dans la query string), la norme exige que cette URL utilise HTTPS, mais authentic ne force pas ce comportement
- Définir les URIs de redirection après déconnexion: idem que la précédente mais pour le point d'accès de déconnexion,
- Éventuellement définir une URI d'identifiant de secteur, voir plus loin,
- Définir la politique des identifiants :
    - identifiant unique : retourne l'UUID des utilisateurs dans authentic (unique donc pour tous les services),
    - par paire : un identifiant unique est généré par service en faisant le hash d'un identifiant de secteur (s'il n'y a qu'une URL de redirection c'est le nom de domaine de cette URL, sinon c'est celui de l'URI de l'identifiant de secteur, cela permet de partager les mêmes identifiants uniques entre des services différents), de l'uuid de l'utilisateur et d'une clé secrète (settings.SECRET_KEY, qui doit donc être sauvegardé précieusement),
    - Choisir un algorithme de signature pour les IDToken (= assertions OIDC), soit RSA qui utilisera la clé générée plus haut, soit HMAC qui utilisera le client_secret de chaque service.

Fonctionnement du consentement et de la génération des subs

Politique des identifiants \ Mode de consentement	Par RP	Par OU
email	rien de spécial	rien de spécial
uuid	rien de spécial	rien de spécial
pairwise irreversible	hash de l'uuid + (domaine de l'unique redirect_uri ou domaine de l'identifiant de secteur)	hash de l'uuid + slug de l'OU
pairwise reversible	chiffrement AES de l'uuid par settings.SECRET_KEY salé avec (domaine de l'unique redirect_uri ou domaine de l'identifiant de secteur)	chiffrement AES de l'uuid par settings.SECRET_KEY salé avec le slug de l'OU

Donc pour avoir un même sub sur plusieurs RPs, plusieurs méthodes:

- choisir un identifiant global comme email ou uuid
- choisir pairwise irreversible avec consentement par RP et une même URI d'identifiant de secteur soit consentement par OU et mettre les RPs dans la même OU
- idem avec pairwise reversible