FUSION DIRECTORY

12 avril 2023

Portrait de la gestion des identités avec des logiciels Libres en 2023 :

Un voyage dans les logiciels
le fonctionnel
et
la technique

Benoit Mortier - CEO @ FusionDirectory

Sommaire

- Pourquoi utiliser des logiciels libres pour la gestion des identités ?
- Du fonctionnel au technique
- La technique au secours du fonctionnel
- Développement et customisation : un point fort des logiciels libres !
- Questions / Réponses
- Des pistes pour aller plus loin

Benoit Mortier
Ceo de FusionDirectory

- Fondateur et CEO de FusionDirectory
- Spécialiste de la gestion des identités
- Coprésidence du thème Administration système lors des rencontres mondiales du logiciel Libre de 2007 à 2013
- Anime régulièrement de nombreuses conférences.

Pourquoi utiliser des logiciels libres pour la gestion des identités ?

Les 4 libertés du logiciel libre

- liberté 0 : la liberté d'exécuter le programme comme vous le souhaitez, dans n'importe quel but.

- Liberté 1 : la liberté d'étudier le fonctionnement du programme et de le modifier pour effectuer ses tâches informatiques comme souhaité.
  - l'accès au code source est une condition nécessaire

- liberté 2 : la liberté de redistribuer des copies, aidant ainsi votre voisin.

- liberté 3 : la liberté de distribuer des copies de vos versions modifiées à d'autres personnes ; ce faisant, cela donne à toute la communauté la possibilité de bénéficier des modifications.
  - l'accès au code source est une condition nécessaire.

Avantages des logiciels libres

- Code source ouvert
- Pas de frais de licence
- Pas de verrouillage du fournisseur
- Code plus sécurisé grâce à une audibilité complète
- Accès ouvert - tests illimités avant adoption
- Transparence totale
- Contributions communautaires disponibles et encouragées
- Des solutions peuvent être construites au-dessus des logiciels
- L'opportunité d'influencer la roadmap produit

Désavantages des logiciels libres

- Pas de connecteur tout fait pour d'autres applicatifs en général
- L'intégration peut être plus complexe
- Nécessite un expertise plus importante
- Difficulté de sélection des différentes briques qui constituent la solution

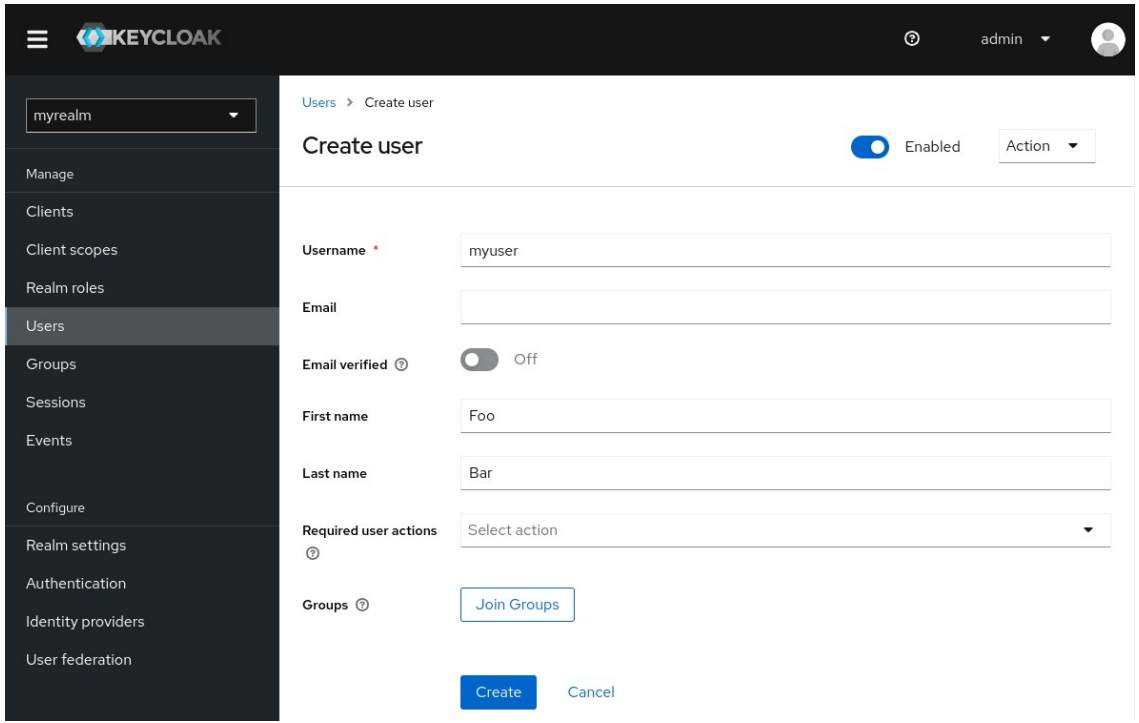
Malgré tout ...

- Conçus pour un grand nombre de situations différentes d'après des cas réels
- Ils sont adaptables à presque toute les situations possibles
- Il respectent le mantra « un outil, une fonction » qui évitent les solutions trop complexes et non maintenables.
- Ils sont évolutifs a votre propre rythme et ne vous forcent pas a des cycle de mises a jour forcés.
- Vous permettent de construire un gestion des identités a votre image

Les logiciels libres et la gestion des identités

- Les logiciels libres couvrent tout le champs d'application de la gestion des identités
- On retrouve des logiciels de SSO, de synchronisation des identités, de gestion des identités, parmis les plus courant on retrouve :
  - Pour les systèmes des sso
    - Keycloak
    - LemonLDAP::NG
  - Pour les outils de gestion des identités
    - FusionDirectory
    - Midpoint
  - Pour les outils de synchronisation
    - LDAP Synchronization Connector

KeyCloak : portail SSO



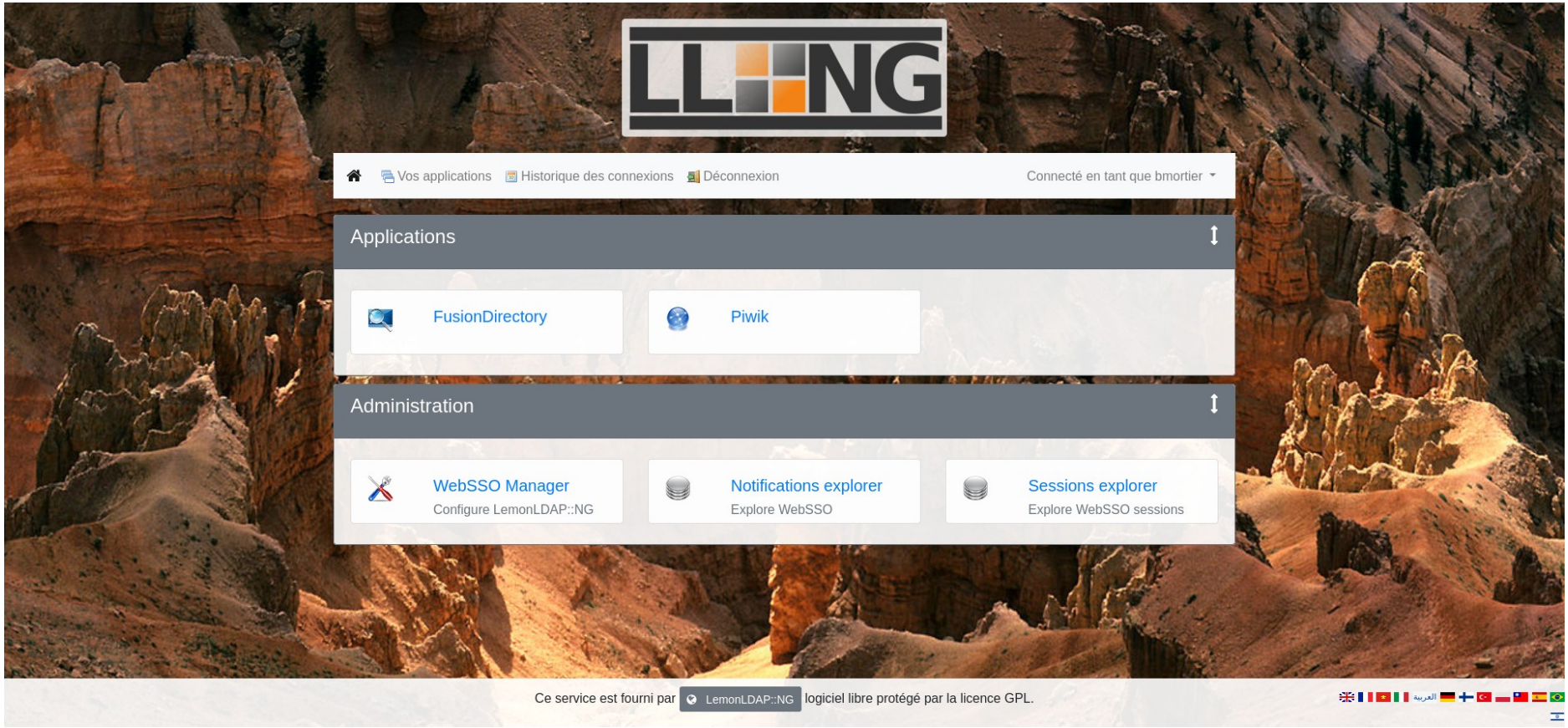
KeyCloak : portail SSO

- Authentification unique
- Connexion avec des comptes de réseau sociaux
- Fédération d'utilisateurs sur des annuaires dispersés (AD et OpenLDAP)
- Console d'administration
- Console de gestion de compte
- Protocoles standard, OpenID Connect, OAuth 2.0 et SAML.

FUSION DIRECTORY

12 avril 2023

LemonLDAP::NG : portail SSO



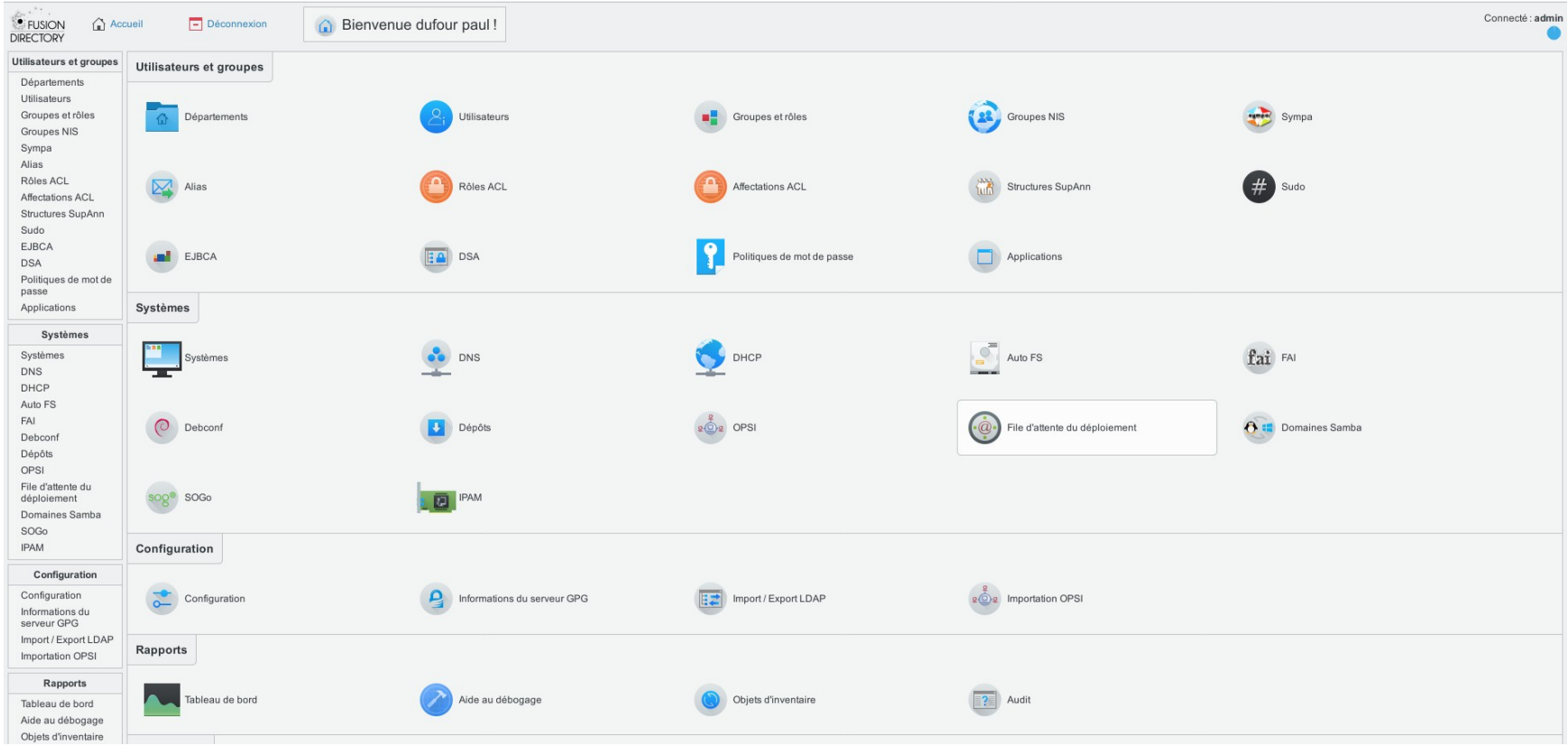
LemonLDAP::NG : portail SSO

- Authentification unique (SSO), contrôle d'accès
- Fournisseur de Service / Fournisseur d'Identité
- Support des protocoles LDAP, Active Directory, Kerberos, Base de données
- Support de CAS, SAML, OpenID Connect
- Connexion avec des comptes de réseau sociaux
- Réinitialisation de mot de passe et création de compte
- Authentification multi-facteurs

FUSION DIRECTORY

12 avril 2023

FusionDirectory : gestion des identités

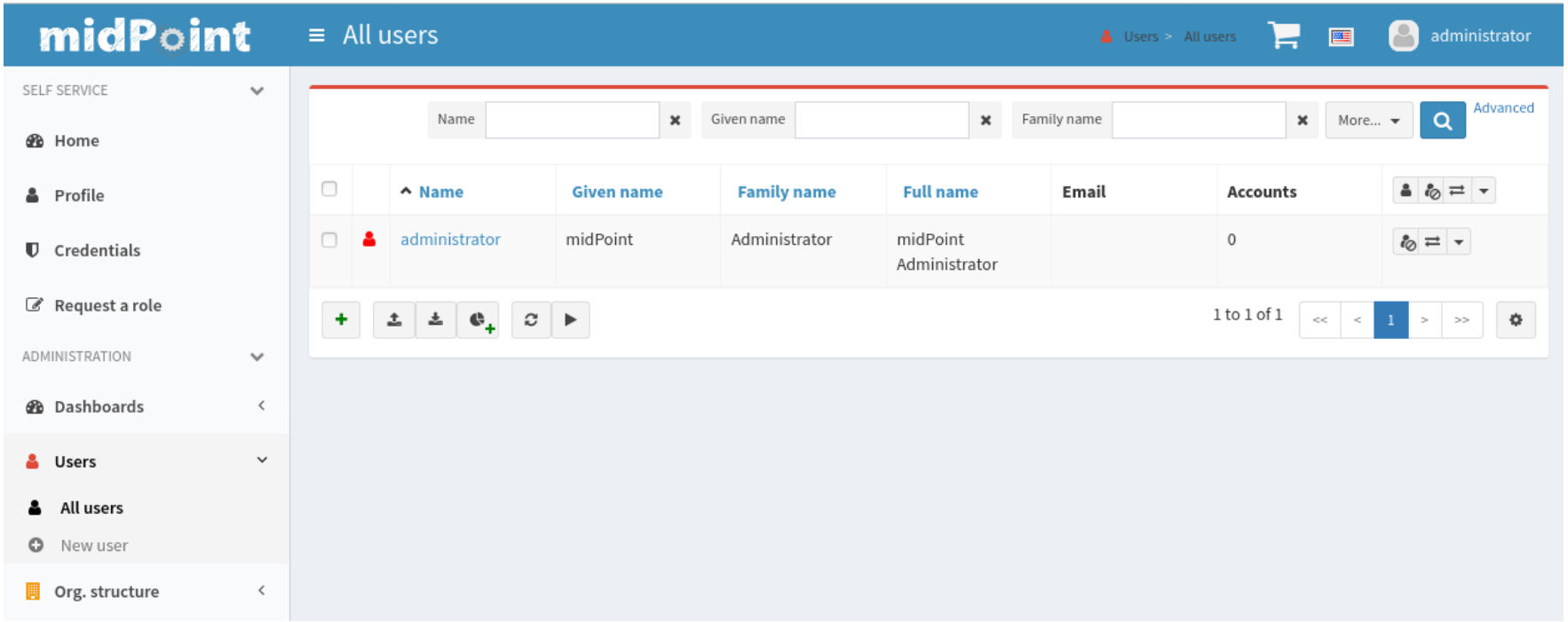


Portrait de la gestion des identités avec des logiciels Libres en 2023 : un voyage dans les logiciels, le fonctionnel et la technique

FusionDirectory : gestion des identités

- Gestion des utilisateurs, groupes, rôles, courriels, dhcp, dns, cyrus, postfix
- Gestion des systèmes et des outils de déploiement FAI, OPSI
- Support complet SupAnn 2018, Sinaps Amue, PARTAGE de RENATER
- Contrôle d'accès fin pour la délégation de tâches
- Modèles personnalisables pour l'approvisionnement des données
- Triggers sur action(s) de création, modification, effacement, vérification
- Webservice REST
- Support CAS, LemonLDAP::NG, WebAuthn, Yubico

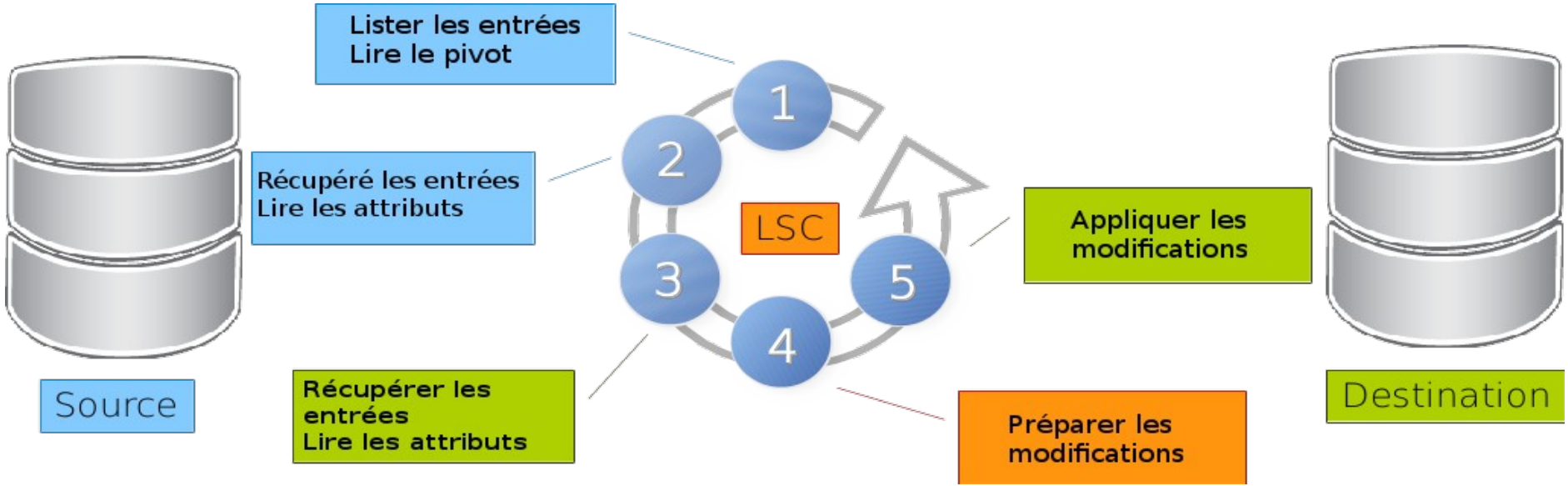
Midpoint : gestion des identités



Midpoint : gestion des identités

- Gouvernance et administration des identités
- Gestion des licences
- Réglementation et conformité
- Gestion de l'identité améliorant la confidentialité
- Libre service
- Évolutivité

LSC : synchronisation des données



LSC : synchronisation des données

- Connecteurs sources multiples :
  - N'importe quel serveur LDAPv3
  - Toute base de données avec un connecteur JDBC
  - Des fichiers à plat
  - Des tiers tels que Google Apps
  - Le webservice REST de FusionDirectory

- Prise en charge des finesses et des extensions LDAPv3 :
  - StartTLS, LDAPS, résultats paginés
  - Synchronisation LDAP / recherche persistante

FUSION DIRECTORY

12 avril 2023

Gestion des identités avec des logiciels libres : du fonctionnel au technique

Passage du technique au fonctionnel

- Pendant très longtemps on à privilégier une vue technique uniquement
- Il s'agissait d'un petite groupe de personnes généralement à la DSI qui gérait cet aspect.
- Ces 10 dernières années on à vu un changement de tendances ou l'ouverture de la gestion des identités vers l'extérieur à amené le fonctionnel au premier plan
- Aujourd'hui l'ensemble des acteurs de la société deviennent parties prenante à la gestion des identités que ce soit en tant que :
  - Fournisseur de données : RH, Applications métiers
  - Consommateur de données : service de support, utilisateurs …
  - Créateur de données : chef de services responsables d'utilisateurs

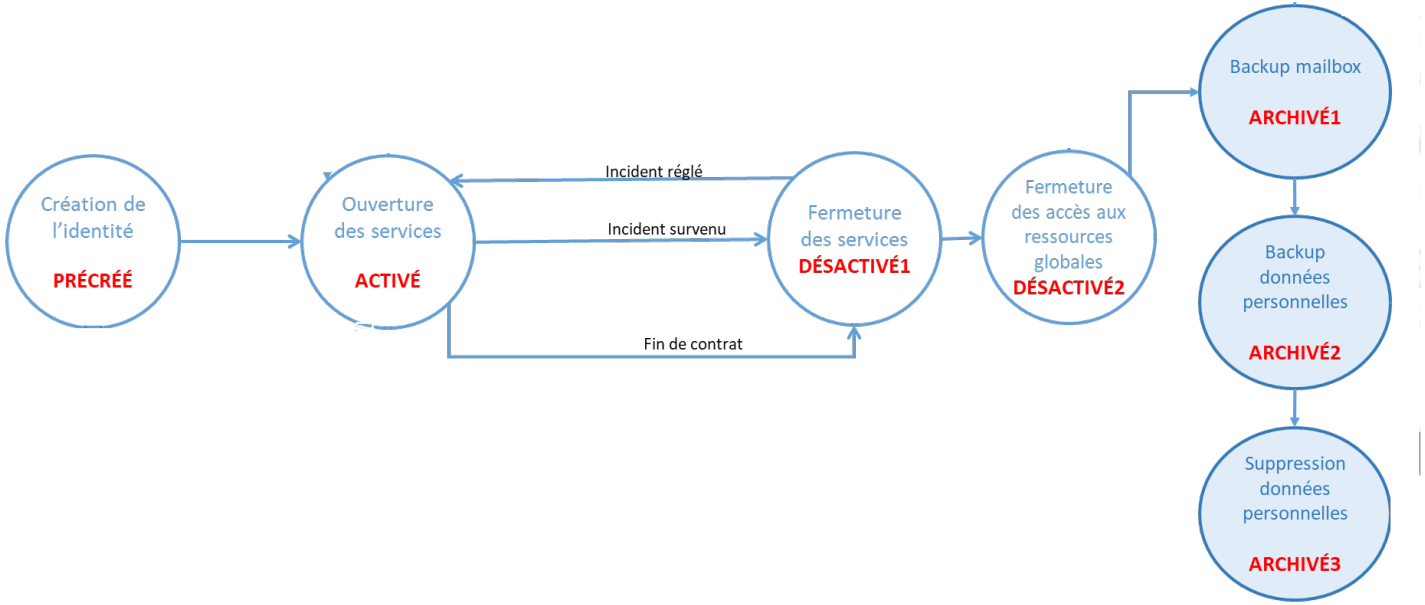
Cycle de vie : de nombreuses sources de données

- Les sources de données sont souvent multiples et variés.
- Les base de données métiers contiennent l'ensemble des informations de vérité pour construire le compte numérique d'un utilisateur.
- Le cycle de vie du compte numérique d'un utilisateur est intimement lié au changements réalises dans les bases de données métiers.
- L'on doit aussi intégrer les comptes numériques d'utilisateur externes mais non présent dans les bases de données métiers
- Il est important que chaque donnée utilisé dans la création d'un compte numérique vienne d'un source de vérité, acte administratif, contrat

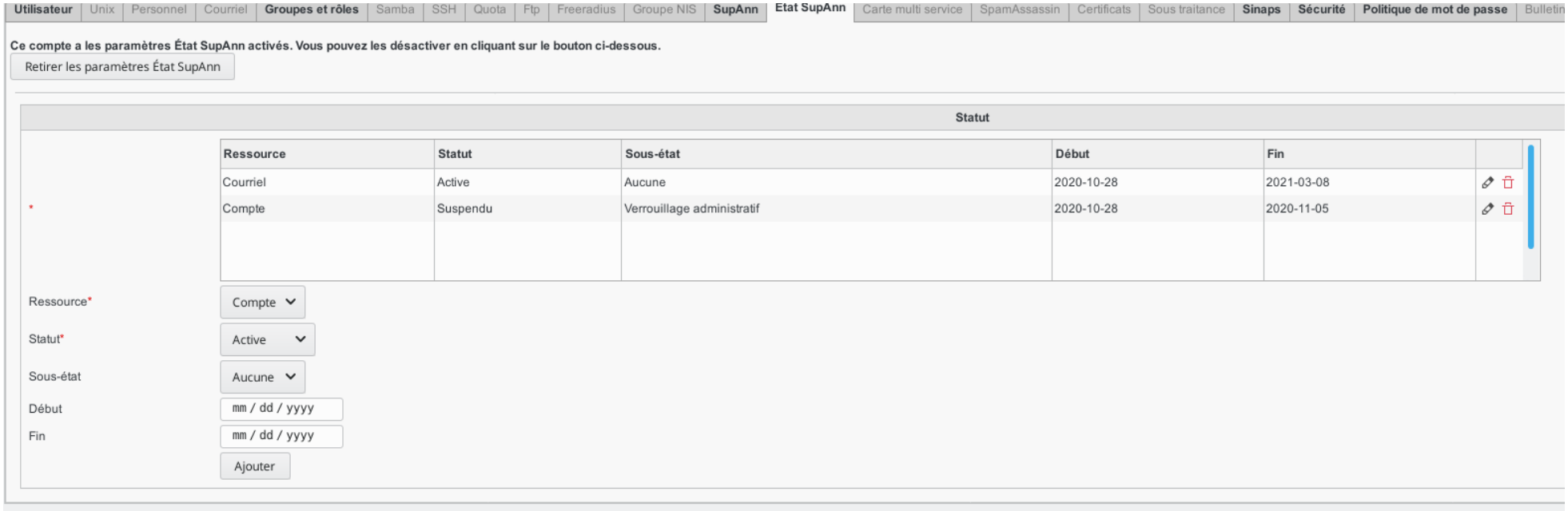
Cycle de vie : avoir une bonne gestion

- La création et la mise a jour des comptes est globalement bien géré
- La désactivation est souvent mal conçue et dépend d'action semi-manuelle
- La désactivation ne prend généralement pas en compte les différents actions nécessaire comme
  - Archivage du home
  - Désactivation et archivage de la boite de messagerie
  - Assurance que tout les droits on bien été enlevés dans les applications externes non lapidifiés
- Archivage du compte numérique au regard de la RGPD
- Archivage du compte numérique au regard de la nom réutilisation de certaines données sensible, email, uid, samAccountName etc...

Cycle de vie: workflow



Cycle de vie : dans FusionDirectory



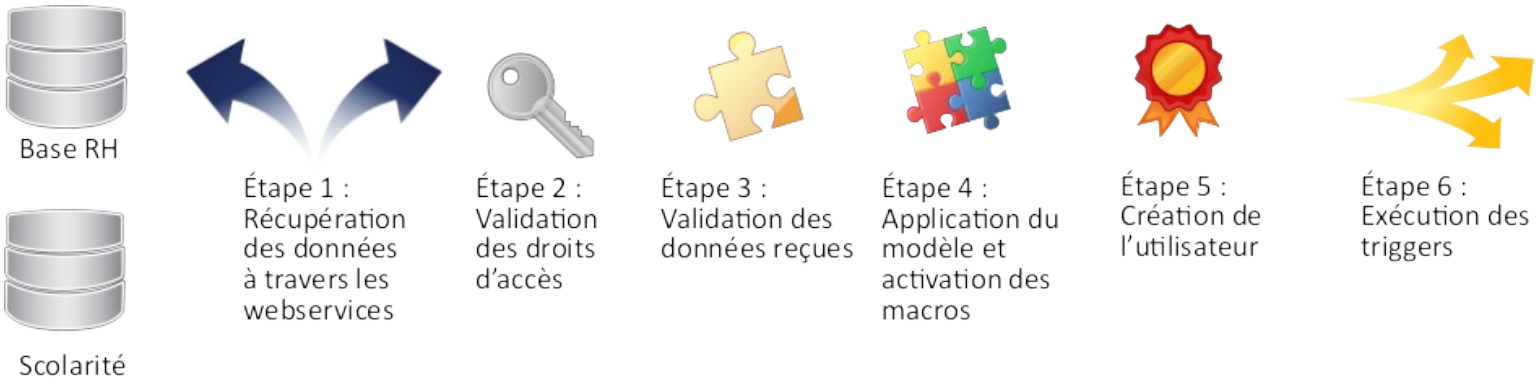
La technique au secours du fonctionnel

Améliorations techniques

- Pour répondre a tout ces besoins les logiciels libre de gestion des identités on du évoluer.
- FusionDirectory par exemple est passe d'un logiciel de gestion des identités à un portail de gestion des identités :
  - Webservices REST
  - Moteur de tâches
  - Moteur de modèles de mails
  - Gestion de l'archivage des comptes
  - Fusiondirectory Orchestrator : Orchestrateur avec endpoint REST
- LSC a vu la création d'un plugin webservice pour FusionDirectory afin de s'intégrer dans des workflow complexes

Webservice REST : FusionDirectory

- Le webservice REST permet l'intégration de la gestion des identités avec des applications tierces de manière plus fluide
- L'utilisation des modèles permet de construire des process de provisionnement qui valident les données
- Le déclenchement des triggers permet des actions supplémentaires



Webservice REST : FusionDirectory

GET /objects/user?base=ou=branche,dc=example,dc=com

→ { "uid=login,ou=people,dc=example,dc=com": "login" }

GET /objects/user/uid=login,ou=people,dc=example,dc=com/posixAccount/uidNumber

→ 1012

PUT /objects/user/uid=login,ou=people,dc=example,dc=com/posixAccount/uidNumber 1000

→ uid=login,ou=people,dc=example,dc=com

Workflow : Gérer des workflow sur mesure

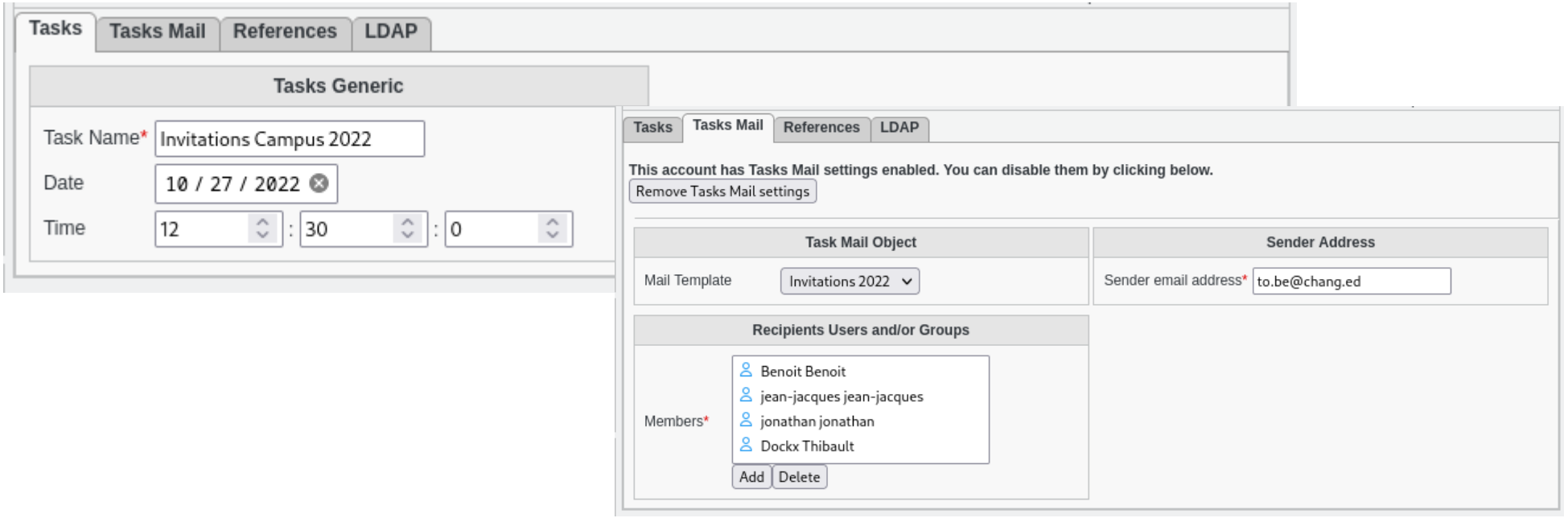
La partie workflow depuis les base métiers et vers d'autres applications et bien couverte fonctionnellement.

Il y a une demande croissante sur la possibilité d'agir de manière automatise sur des événements en fonction des données présente dans la gestion des identités

- La désactivation automatique de certaines ressources en fonction de date de fin de validité
- L'approbation de compte crées automatiquement ou par une certaine catégorie de personnel
- De déclencher automatiquement les actions de gestion de compte (création, activation, désactivation, suppression, …) en fonction de l'état de la ressource

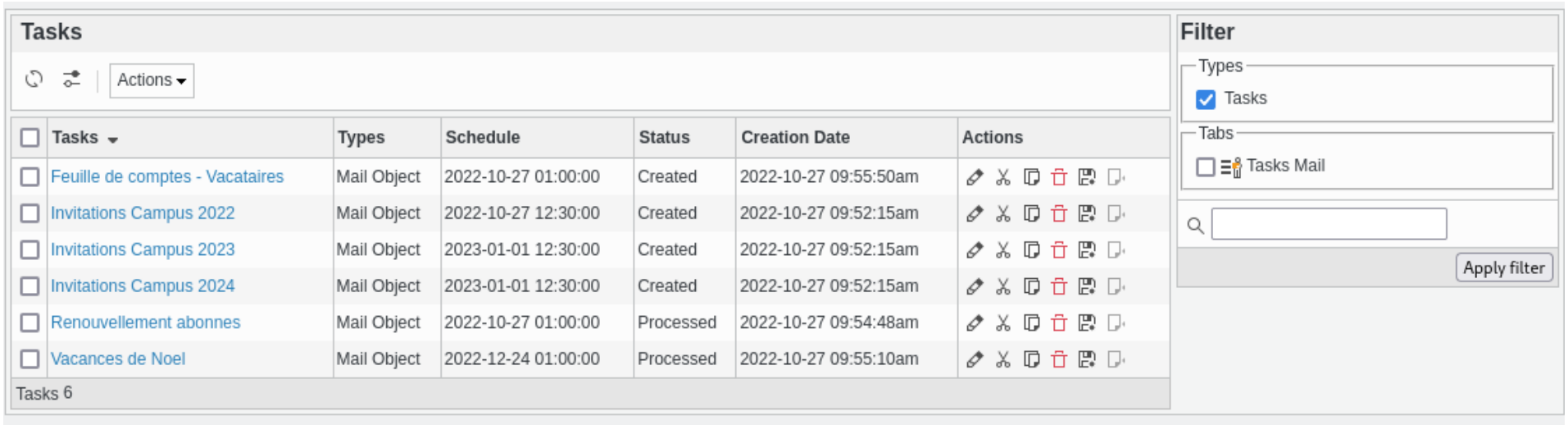
Workflow : Création de tâches

- Création de taches génériques
- Création d'une tache mail associé



Workflow : Suivi des tâches

- Tableau de bord permettant de suivre l'ensemble des taches et sous taches ainsi que leur statuts actuels



Orchestrateur : FusionDirectory Orchestrator

FusionDirectory Orchestrator fournit un webservice Web avec une API REST.

Il permet :

- Une gestion granulaire de certaines tâches spécialisées.
- Offre une vue simple sur l'état de chaque tâche.
- Actuellement uniquement des taches de type mail

Orchestrateur : exemple de fonctionnement

Dans cet exemple FusionDirectory orchestrator on va recupérer des tâches mail

- Obtenir le token d'accès

```
curl -X POST -H "Content-Type: application/json" https://orchestrator.fusiondirectory.org/orchestrator/api/login -d '{"username":"admin","password":"secret"}'
```

- Obtenir toute les taches mails

```
curl -X GET -H "Content-Type: application/json" -H "Authorization: Bearer {token}"  https://orchestrator.fusiondirectory.org/orchestrator/api/tasks/mail
```

- Demander l'exécution des taches mails en attente

```
curl -X PATCH -H "Content-Type: application/json" -H "Authorization: Bearer {token}"  https://orchestrator.fusiondirectory.org/orchestrator/api/tasks/mail
```

- Demander de refraichir le token

```
curl -X POST -H "Content-Type: application/json" https://orchestrator.fusiondirectory.org/orchestrator/api/refresh -d '{"token":"refresh-token"}'
```

Synchro : LSC plugin webservice FusionDirectory

Ce plugin utilise FusionDirectory REST API, disponible en version 1.4.

- Rajoute un service source/destination Webservice FusionDirectory
- Donne accès a l'API de FusionDirectory pour faire du scripting

Synchro : exemple de destination FusionDirectory

```
<pluginDestinationService implementationClass="org.lsc.plugins.connectors.fusiondirectory.Fus$
  <name>fusiondirectory-dest-service</name>
  <connection reference="fusiondirectory" />
  <fusiondirectory:serviceSettings>
    <name>fusiondirectory-service-settings</name>
    <connection reference="fusiondirectory" />
    <fusiondirectory:entity>USER</fusiondirectory:entity>
    <fusiondirectory:pivot>supannEmpId</fusiondirectory:pivot>
    <fusiondirectory:template>cn=template-lsc,ou=templates,ou=people,dc=formation-fusiondirec$
    <fusiondirectory:attributes>
      <fusiondirectory:tab name="user">
        <fusiondirectory:attribute>base</fusiondirectory:attribute>
        <fusiondirectory:attribute>sn</fusiondirectory:attribute>
        <fusiondirectory:attribute>givenName</fusiondirectory:attribute>
        <fusiondirectory:attribute multiple="true">title</fusiondirectory:attribute>
        <fusiondirectory:attribute>userPassword</fusiondirectory:attribute>
      </fusiondirectory:tab>
      <fusiondirectory:tab name="mailAccount">
        <fusiondirectory:attribute>mail</fusiondirectory:attribute>
      </fusiondirectory:tab>
      <fusiondirectory:tab name="supannAccount">
        <fusiondirectory:attribute>supannEmpId</fusiondirectory:attribute>
      </fusiondirectory:tab>
    </fusiondirectory:attributes>
  </fusiondirectory:serviceSettings>
</pluginDestinationService>
```

Cas d’usage : workflow complexes de gestion des identités

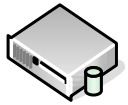
Cas concret : Université Unamur

- Utiliser les webservices de FusionDirectory et les modèles afin de provisionner les utilisateurs par type dans toute leur complexité
- Propager les mots de passe dans 3 ActiveDirectory différents
- Créer les fiches de contact utilisateurs Office 365

Cas concret : Université Unamur

UNIVERSITÉ DE NAMUR

Personnels
Étudiants



Contrôle d'accès

FUSION DIRECTORY

Création d'utilisateurs
Création Fiche Office 365

Synchronisation mot de passe

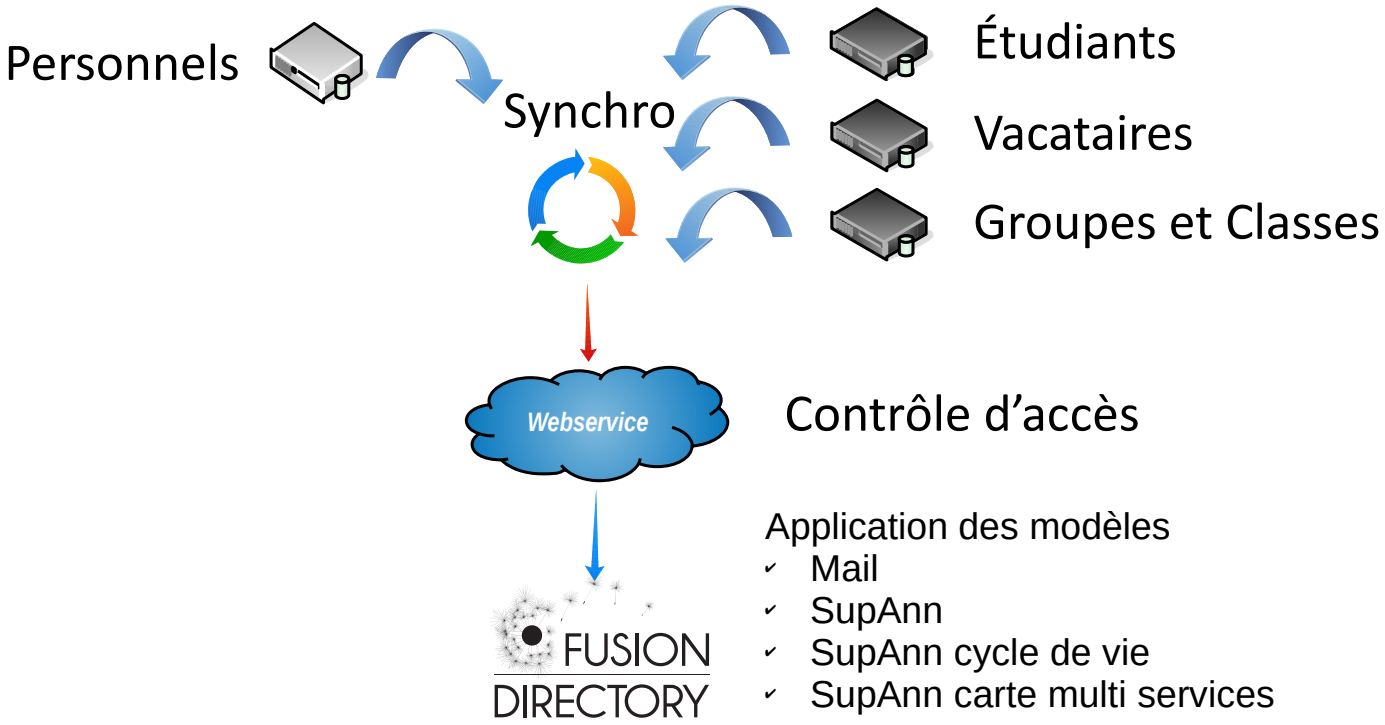
Cas concret : TELECOM SudParis

TELECOM SudParis

- Utiliser les webservices de FusionDirectory et les modèles afin de provisionner les utilisateurs par type dans toute leur complexité
- Utiliser les webservices de unicampus et FusionDirectory afin de propager les numéros de badges lors de leur création
- Utiliser les webservices de FusionDirectory afin d'archiver les comptes en fonction des statuts du cycle de vie

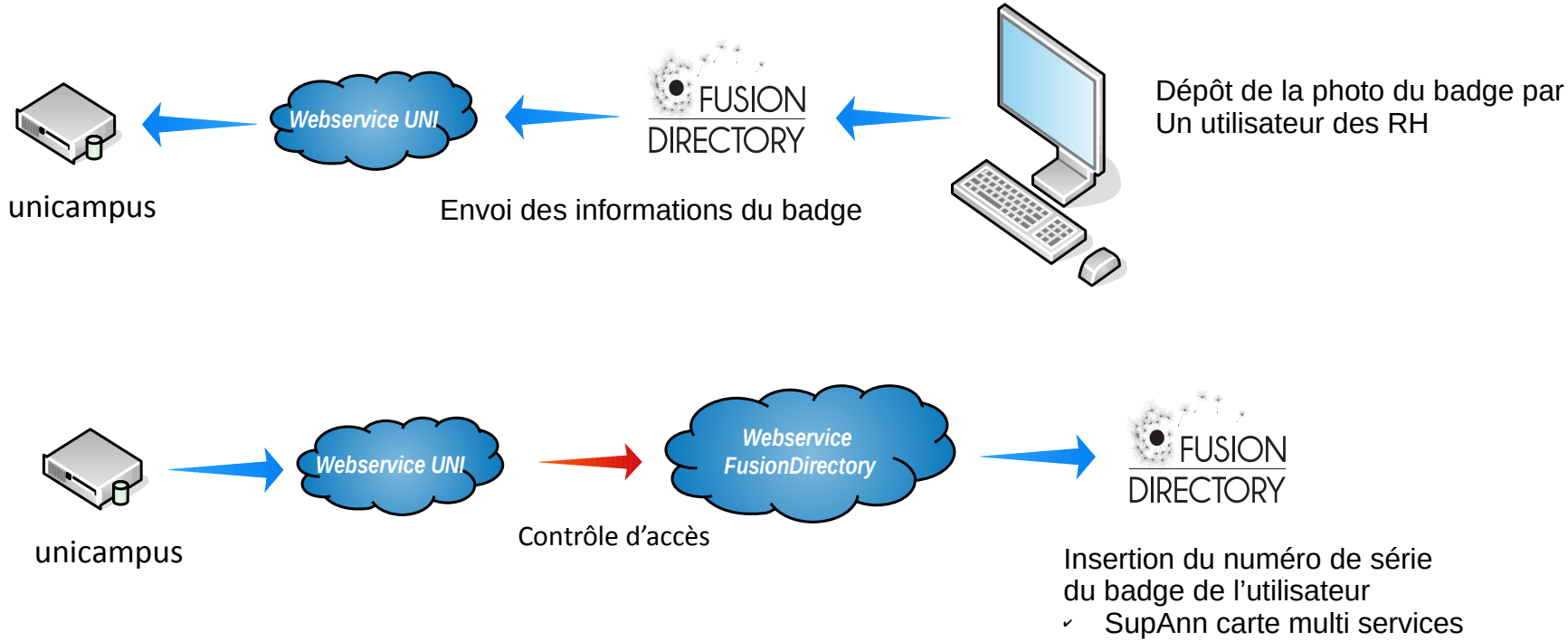
Cas concret : TELECOM SudParis

Création de comptes numériques



Cas concret : TELECOM SudParis

Création de badges d'accès



Développement et customisation : un point fort des logiciels libres !

Développement et customisation !

Un des points forts des logiciels libres sont leur extensibilité et leur customisation.

- FusionDirectory :
  - API de création de plugin : simplePlugin
  - Triggers permettant de scripter apres des opérations bien précises
- LemonLDAP::NG :
  - API d'écriture de plugins
- LSC :
  - Plugin executable
  - Plugin d'exemple pour écrire de nouveaux plugins

Développement : SimplePlugin API FusionDirectory

Historiquement Gosa[2] l'ancêtre de FusionDirectory n'avait pas d'api mais des fonctions disparates, pas prévues pour évoluer et ne formait pas un ensemble complet.

Lors des réflexions autour de la naissance de FusionDirectory, la question d'une API propre s'est imposée.

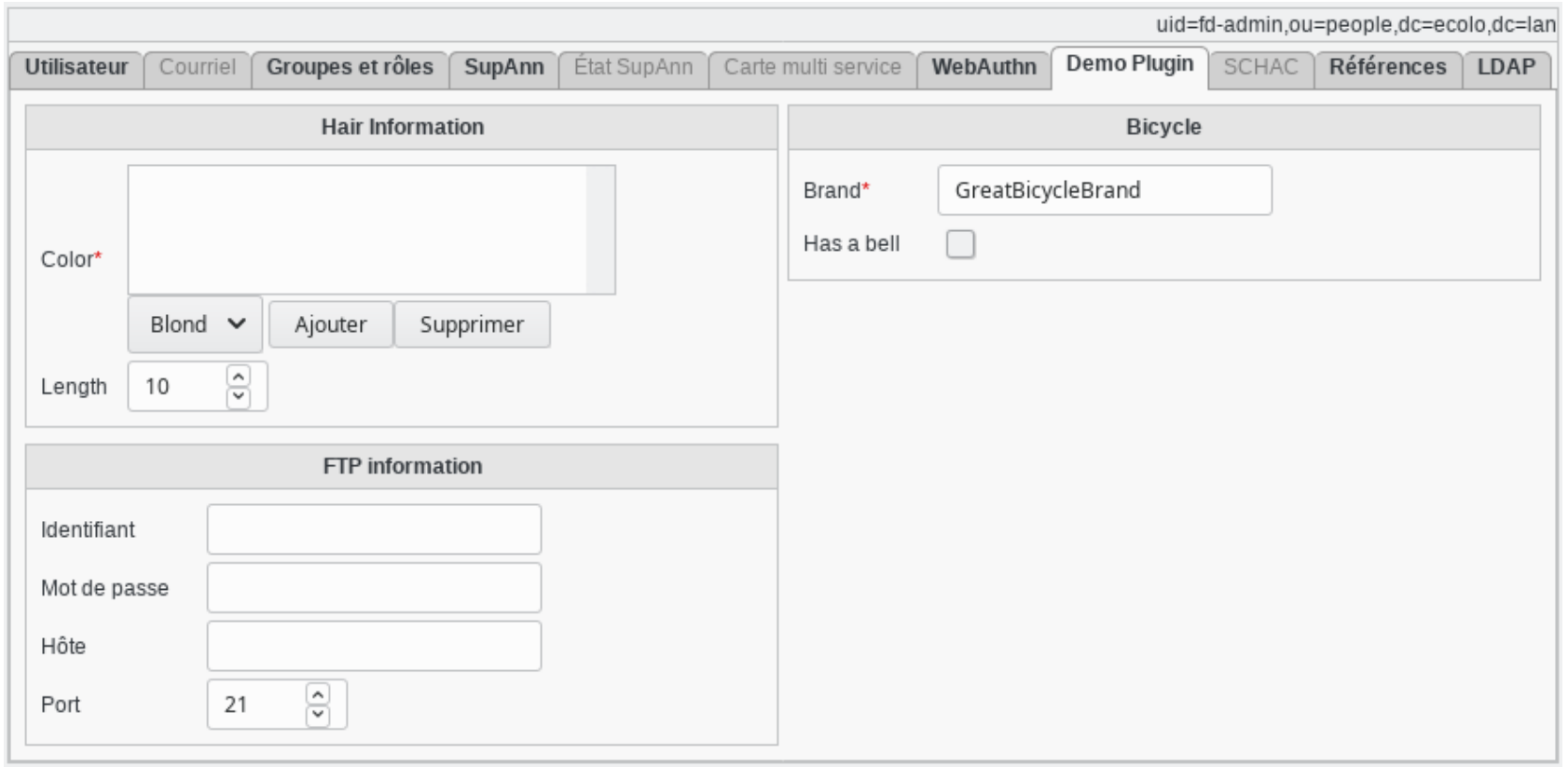
Parmi ses fonctionnalités les plus importantes on retrouve :

- Faciliter via une couche d'abstraction le stockage dans un annuaire LDAP
- Construire automatiquement l'interface graphique de manière simple et automatique
- Gérer automatiquement les acls de FusionDirectory sans écrire de code supplémentaire
- Fournir un ensemble d'attributs pour simplifier l'écriture de plugins gérant des données complexes

Développement : SimplePlugin API FusionDirectory

```php
<?php
// The main function : information about attributes
static function getAttributesInfo ()
{
  return [
    // Attributes are grouped by section
    'section1' => [
      'name'  => _('Hair Information'),
      'attrs' => [
        new SetAttribute(                  // This attribute is multi-valuated
          new SelectAttribute (
            _('Color'),                    // Label of the attribute
            _('Color of the hair'),        // Description
            'hairColor',                   // LDAP name
            TRUE,                          // Mandatory
            ['blond','black','brown'],     // [SelectAttribute] Choices
            '', // We don't set any default value, it will be the first one
            ['Blond','Black','Brown']      // [SelectAttribute] Output choices
          )
        ),
```

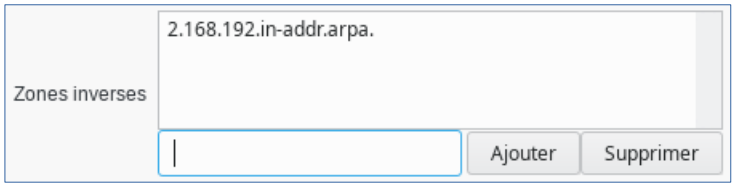
Développement : SimplePlugin API FusionDirectory



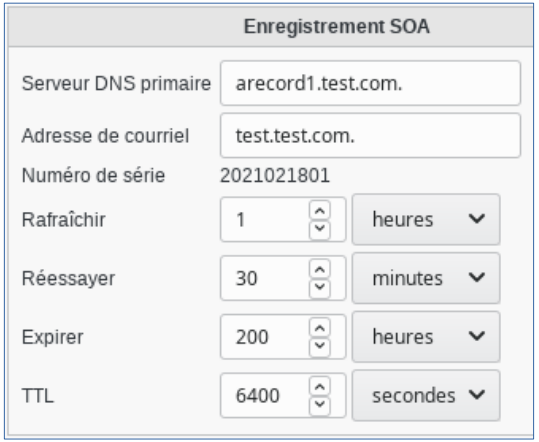
Développement : SimplePlugin API FusionDirectory

- Le SetAttribute englobe un attribut pour le rendre multivalué.
- Le CompositeAttribute permet de découper un seul attribut LDAP en plusieurs attributs dans l'interface.
- Un ObjectsAttribute permet de choisir des objets via une fenêtre de gestion.

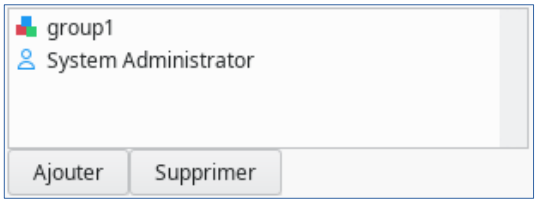
SetAttribute



CompositeAttribute

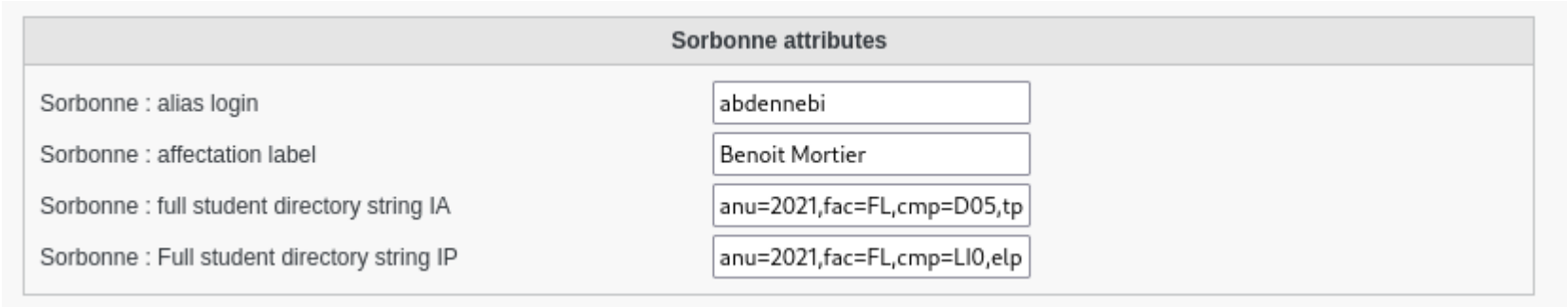


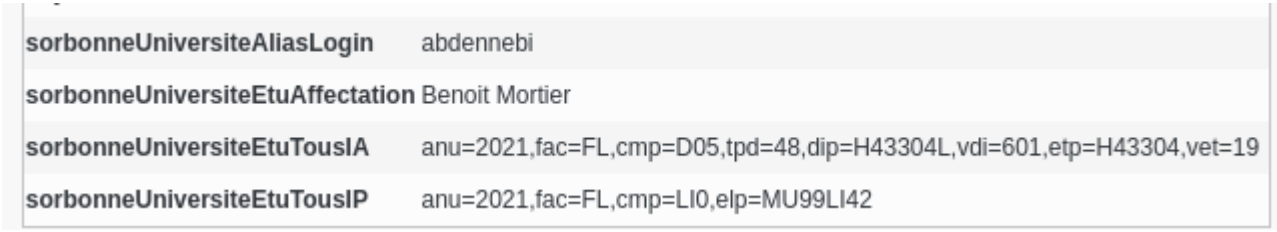
ObjectsAttribute



Développement : plugin personalisé

- La facilite d'utilisation de simplePlugin permet d'ecrire rapidement un plugin pour rajouter des données propres a l'etablissement qu'il l'utilise
- Ca permet de bénéficier de l'ensemble des fonctionnalités de FusionDirectory





Customisation : Triggers

- Il existe de nombreux cas où on désire déclencher des actions après la création, la modification ou la vérification de données
- Dans le cas de FusionDirectory, une liberté totale est laissée en ce qui concerne l'écriture des triggers
- Cela peut être utilisé pour appeler d'autres webservices ou déclencher des processus de synchronisation avec d'autres applicatifs

Étape 1 : Création du compte

Étape 2 : Exécution des triggers

Étape 3 : insertion dans ActiveDirectory

Développement : Plugin LemonLDAP::NG

Les plugins de portail vous permettent de personnaliser le comportement de LemonLDAP::NG.

Les cas d'utilisation courants des plugins sont :

- Recherche d'informations de session dans un backend supplémentaire
- Mise en œuvre de contrôles ou d'étapes supplémentaires lors de la connexion
- Ajustement du comportement des protocoles SAML, OIDC ou CAS pour contourner les bugs applicatifs

Développement : Plugin LemonLDAP::NG

Les plugins de portail vous permettent de personnaliser le comportement de LemonLDAP::NG.

Les cas d'utilisation courants des plugins sont :

- Recherche d'informations de session dans un backend supplémentaire
- Mise en œuvre de contrôles ou d'étapes supplémentaires lors de la connexion
- Ajustement du comportement des protocoles SAML, OIDC ou CAS pour contourner les bugs applicatifs

Développement : LSC executable plugin

Le plugin exécutable permet à tout administrateur système d'utiliser LSC sur une source ou une destination personnalisée sans écrire de code Java, uniquement en enveloppant chaque méthode via une commande de script.

- Peut être écrit dans n’importe quel language
- Les scripts doivent renvoyer sur la sortie standard le contenu attendu.
- Le flux d'erreur est réservé aux messages qui doivent être traités comme un message d'erreur.
- Un code retour non nul indique que le script a rencontré une erreur.

Développement : LSC executable plugin

- Wrapper qui permet d'être utiliser comme sources ou destination
- Source
  - Script pour lister les objets (list)
  - Script pour récupérer un objet (get)
- Destination
  - Script pour ajouter un objet (add)
  - Script pour mettre à jour un objet (update)
  - Script pour supprimer un object (delete)
  - Script pour renommer un objet (modrdn)

Développement : LSC executable plugin

```
<pluginConnection>
  <name>executable</name>
  <url>fake</url>
  <username>fake</username>
  <password>fake</password>
</pluginConnection>
```

```
<pluginSourceService implementationClass="org.lsc.plugins.connectors.executable.ExecutableLdifSourceService">
  <name>user-src-service</name>
  <connection reference="executable" />
  <exec:executableLdifSourceServiceSettings>
    <name>user-src-service-exec</name>
    <connection reference="executable" />
    <exec:listScript>path/to/listscript</exec:listScript>
    <exec:getScript>path/to/getscript</exec:getScript>
    <exec:variables>
      <entry><key>key</key><value>value</value></entry>
      <entry><key>key2</key><value>value2</value></entry>
    </exec:variables>
  </exec:executableLdifSourceServiceSettings>
</pluginSourceService>
```

Développement : LSC executable plugin

```
<pluginDestinationService implementationClass="org.lsc.plugins.connectors.executable.ExecutableLdifDestinationService">
  <name>user-dst-service</name>
  <connection reference="executable" />
  <exec:executableLdifDestinationServiceSettings>
    <name>user-dst-service-exec</name>
    <connection reference="executable" />
    <exec:listScript>path/to/listscript</exec:listScript>
    <exec:getScript>path/to/getscript</exec:getScript>
    <exec:addScript>path/to/addscript</exec:addScript>
    <exec:updateScript>path/to/updatescript</exec:updateScript>
    <exec:removeScript>path/to/removescript</exec:removeScript>
    <exec:renameScript>path/to/renamescript</exec:renameScript>
    <exec:variables>
      <entry><key>key</key><value>value</value></entry>
      <entry><key>key2</key><value>value2</value></entry>
    </exec:variables>
    <exec:fetchedAttributes>
      <string>uid</string>
      <string>nom</string>
      <string>prenom</string>
    </exec:fetchedAttributes>
  </exec:executableLdifDestinationServiceSettings>
</pluginDestinationService>
```

Développement : LSC Création d'un plugin

Un plugin LSC est défini comme un composant tiers qui peut être :

- un service source
- un service à destination
- un support de langage de script
- une implémentation des options de synchronisation

Développement : LSC Création d'un plugin

Un plugin de service comprend :

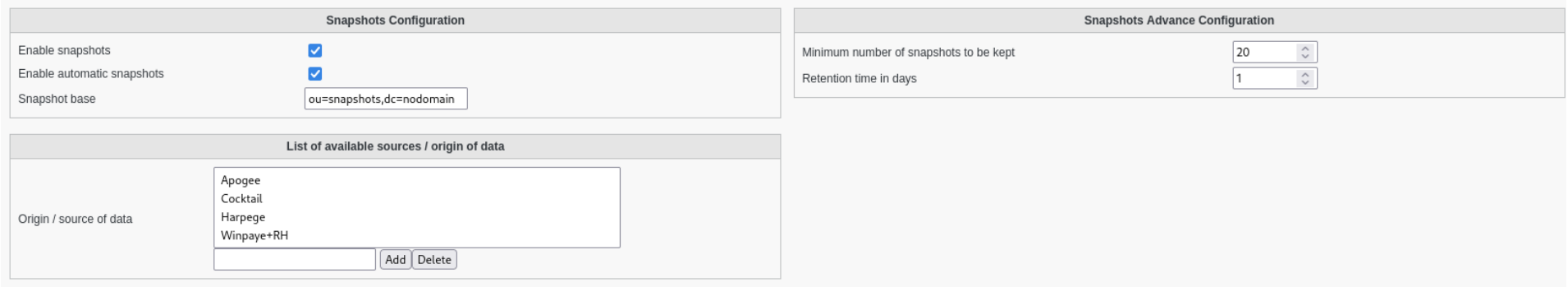
- Une extension XSD pour définir les paramètres du service de plugin
- Un service de plugin qui doit implémenter
  - soit IService pour un service source
  - soit IWritableService pour un service de destination
- une définition de connexion si elle n'est pas déjà définie dans LSC

C

Des pistes pour aller plus loin

Améliorations Futures

- La demande pour des workflow de création de comptes avec approbation a différents niveaux
- Un forte demande apparaît sur les concept d'agrégation / désintégration de comptes numériques
- Un historique de toutes les modifications effectue sur une fiche avec possibilité de revenir en arrière pour rejouer les modifications



Questions / Réponses

Références

- FusionDirectory : https://www.fusiondirectory.org/
- Documentation FusionDirectory: https://fusiondirectory-user-manual.readthedocs.io/en/1.4/index.html
- FusionDirectory API REST : https://rest-api.fusiondirectory.org/
- Forge logicielle FusionDirectory : https://gitlab.fusiondirectory.org/
- LemonLDAP : https://lemonldap-ng.org/
- Forge logicielle LemonLDAP : https://gitlab.ow2.org/lemonldap-ng
- LemonLDAP API Plugin : https://lemonldap-ng.org/documentation/latest/plugincustom.html#write-a-custom-plugin
- LSC : https://lsc-project.org/index.html
- LSC plugin API : https://lsc-project.org/doku.php/documentation/latest/development/addingplugin
- LSC plugin executable : https://lsc-project.org/doku.php/documentation/plugins/executable