

Grouper à Paris1

Présentation

- grouper 2.2.1
- en production depuis janvier 2013
- 2670 groupes (13600 membres pour le plus gros, 50 membres en moyenne) dont 900 groupes synchronisés
- grouper UI shibbolethisé
- petites adaptations sur l'interface :
 - i18n (de Lille1)
 - correctif pour la recherche des noms de groupe [1]
 - remplacement de la recherche des personnes par le webwidget <https://github.com/prigaux/wsgroups> (voir aussi <http://wsgroups.univ-paris1.fr/web-widget/test-search-user.html>) pour contourner l'absence de gestion des homonymes. Cf [patches](#) et [README_univ_modification.md](#)
 - intégration dans l'ENT, avec :
 - suppression de la colonne de gauche pour simplifier l'interface
 - mise en valeur du bouton "+ Ajouter des membres"



- les groupes sont utilisées uniquement via LDAP
 - psp pour synchroniser grouper vers OpenLDAP
 - avec gestion supannGroupeAdminDn et supannGroupeLecteurDn
 - avec gestion businessCategory, seeAlso, labeledURI (pour les groupes créés automatiquement par GSHcreator.py) suite à des pbs avec la synchro incrémentale PspChangeLogConsumer [2] toutes les 10 secondes, remplacement par un petit script plus puissant [3] appelant :

```
gsh -psp -sync <noms de groupes mentionnés dans la table grouper_change_log_type>
```

- grouper-loader pour synchroniser
 - des groupes de bases de données variées vers grouper (apogee, sof, sifac, sesame, harpege)
 - des groupes de LDAP vers grouper (groupes créés automatiquement avec le script : <https://subversion.cru.fr/esup-tools/PAGScreator/trunk/GSHcreator.py>)
- invitation de personnes hors Paris1

Invitation de personnes hors Paris1

Pour gérer les personnes hors Paris1, nous avons mis en place les [external subjects](#).

Nous synchronisons ensuite ces personnes dans une branche de notre LDAP [4]. Voilà une vue pas à pas de ces fonctionnalités :

- Dans l'interface d'admin d'un groupe, on peut faire "Invite external people"

Current location is:

Root: Applications: Media-webdav: podcast-admins

<u>Name</u>	podcast-admins
<u>Path</u>	Applications:Media-webdav:podcast-admins
<u>Description</u>	Acces a https://media-webdav.univ-paris1.fr/podcast/
<u>ID</u>	podcast-admins
<u>ID Path</u>	applications:media-webdav:podcast-admins
<u>Alternate ID Path</u>	
<u>UUID</u>	b080788256de4646b0358c8c5f42ff7b
<u>Types</u>	base members List field

Show entities with ADMIN privilege

[Delete](#) [Add to Group workspace](#) [Edit group](#) [Manage members](#) [Add members](#)

[Manage members lite](#) [Invite external people](#) [Move group](#) [Copy group](#)

[Audit log](#) [View entity details](#)

- On peut ensuite inviter des personnes en donnant leur adresse mail

Invite external people to participate in groups ⓘ

[Lite UI](#)

[Admin UI](#)

* indicates a required field

Enter the invitation information

Email addresses of people to invite *

pascal@rigaux.org

Email subject

invitation à rejoindre le groupe "podcast admins"

Message to users

Bonjour,
Je vous invite à rejoindre le groupe
"podcast admins". Pour cela veuillez
cliquer sur le lien suivant :
\$inviteLink\$

Email addresses to notify when registered

Groups to assign to new users

Enter search text to find a group

Applications:Media-webdav:podcast-admins

- Voici le mail reçu

from **Grouper Paris1** <no-reply@univ-paris1.fr> ★
subject **invitation à rejoindre le groupe "podcast admins"** 10:08
to pascal@rigaux.org ★ other actions ▼

Bonjour,

Je vous invite à rejoindre le groupe "podcast admins". Pour cela veuillez cliquer sur le lien suivant :

<https://grouper.univ-paris1.fr/grouper/grouperExternal/appHtml/grouper.html?operation=ExternalSubjectSelfRegister.externalSubjectSelfRegister&externalSubjectInviteId=adec30da36a245d89e7fed8545e7fd4f>

Cordialement,

--

Pascal Rigaux

- La personne clique sur le lien, se logue en shibboleth, arrive sur la page suivante (nécessite le patch [5])

Self registration for people external to this institution ⓘ

* indicates a required field

Edit your information	
<u>Login ID</u>	2372297@sac.cru.fr
<u>Name</u> *	Pascal Rémi Rigaux
<u>Email</u>	pascal@rigaux.org
Prenom	Pascal Rémi
Nom complet *	
Nom de famille *	Rigaux
<input type="submit" value="Submit"/>	

- Puis

Success: your invitation from Rigaux Pascal has been processed. You were added to the roles: podcast-admins.

- Voici le résultat dans SimpleMembershipUpdate

Gestion des groupes (Grouper)

Mettre à jour les membres d'un groupe ⓘ

Gérer les groupes

Vous êtes actuellement sur: Applications: Media-webdav: podcast-admins

[Trouver un autre groupe](#)

Nom	podcast-admins
Adresse longue	Applications:Media-webdav:podcast-admins
Description	Acces a https://media-webdav.univ-paris1.fr/podcast/

[Plus de choix...](#)

Ajouter un membre

Saisir les premiers caractères pour trouver un membre à ajouter - Merci de patienter car la recherche dans l'annuaire peut mettre un certain

Ajouter un membre

Liste des membres

Les membres du groupe: 1-8 of 8 page si

Sort by: Name ▼

- ☐ ~~Pascal Rigaux~~ ▼
- ☐ ~~Thomas (April 2018)~~ ▼
- ☐ ~~David (April 2018)~~ ▼
- ☐ ~~Philippe (April 2018)~~ ▼
- ☐ ~~Thomas (April 2018)~~ ▼
- ☐ Pascal Rémi Rigaux (2372297@sac.cru.fr) ▼
- ☐ Pascal Rigaux ▼
- ☐ ~~David (April 2018)~~ ▼

Page de résultat: 1

- Le résultat dans LDAP

```
dn: cn=applications.media-webdav.podcast-admins,ou=groups,dc=univ-paris1,dc=fr
objectClass: groupOfNames
objectClass: supannGroupe
objectClass: top
cn: applications.media-webdav.podcast-admins
ou: Applications:Media-webdav:podcast-admins
description: Acces a https://xxx.univ-paris1.fr/podcast/
owner: cn=grouper,ou=admin,dc=univ-paris1,dc=fr
member: uid=xxxxx,ou=people,dc=univ-paris1,dc=fr
member: uid=xxxxxxxx,ou=people,dc=univ-paris1,dc=fr
member: uid=xxxxxxxx,ou=people,dc=univ-paris1,dc=fr
member: uid=xxxxxxxx,ou=people,dc=univ-paris1,dc=fr
member: uid=xxxxxx,ou=people,dc=univ-paris1,dc=fr
member: uid=prigaux,ou=people,dc=univ-paris1,dc=fr
member: uid=xxxxx,ou=people,dc=univ-paris1,dc=fr
member: eduPersonPrincipalName=2372297@sac.cru.fr,ou=externalPeople,dc=univ-paris1,dc=fr
supannGroupeAdminDN: uid=xxxxxxxx,ou=people,dc=univ-paris1,dc=fr
supannGroupeAdminDN: uid=prigaux,ou=people,dc=univ-paris1,dc=fr
```

et

```
dn: eduPersonPrincipalName=2372297@sac.cru.fr,ou=externalPeople,dc=univ-paris1,dc=fr
objectClass: eduPerson
objectClass: inetOrgPerson
objectClass: supannPerson
eduPersonPrincipalName: 2372297@sac.cru.fr
displayName: Pascal Rémi Rigaux
sn: Rigaux
cn: Rigaux R Pascal
givenName: Pascal Rémi
supannMailPerso: pascal@rigaux.org
supannParrairainDN: cn=grouper,ou=admin,dc=univ-paris1,dc=fr
```

Les applications que nous avons configuré ou testé avec des member ou=externalPeople :

- nuxeo [\[6\]](#)
- uportal [\[7\]](#)
- sympa

Notes

1. fix search in lite-ui SimpleMembershipUpdate: it was searching on "name" and displaying "display-name". Make it search on name, display-name and description (as a side effect, it searches on each words separately)

```
--- a/java/src/edu/internet2/middleware/grouper/grouperUi/serviceLogic/SimpleMembershipUpdateFilter.java
+++ b/java/src/edu/internet2/middleware/grouper/grouperUi/serviceLogic/SimpleMembershipUpdateFilter.java
@@ -108,8 +108,8 @@ public class SimpleMembershipUpdateFilter {
        GrouperUiUtils.message("simpleMembershipUpdate.errorNotEnoughGroupChars", false),
        "bullet_error.png");
    } else {
        queryOptions = new QueryOptions().paging(TagUtils.mediaResourceInt("simpleMembershipUpdate.
groupComboboxResultSize", 200), 1, true).sortAsc("theGroup.displayNameDb");
        - groups = GrouperDAOFactory.getFactory().getGroup().getAllGroupsSecure("%" + searchTerm + "%",
grouperSession, loggedInSubject,
        - GrouperUtil.toSet(AccessPrivilege.ADMIN, AccessPrivilege.UPDATE), queryOptions, TypeOfGroup.
GROUP_OR_ROLE_SET);
        + groups = GrouperDAOFactory.getFactory().getGroup().getAllGroupsSplitScopeSecure(searchTerm,
grouperSession, loggedInSubject,
        + GrouperUtil.toSet
(AccessPrivilege.ADMIN, AccessPrivilege.UPDATE), queryOptions, TypeOfGroup.GROUP_OR_ROLE_SET);
        if (GrouperUtil.length(groups) == 0) {
            GrouperUiUtils.dhtmlxOptionAppend(xmlBuilder, "",
```

```
2. grouper_error.log.2013-02-05:2013-02-06 10:18:10,019: [DefaultQuartzScheduler_Worker-10] ERROR
ChangeLogHelper.processRecords(281) - - Did not get all the way through the batch! 18848 != 18948
```

3. PspChangeLogConsumer ne gère que l'ajout/suppression de membres, pas la création/modification de groupes...
Le script de synchro incrémentale est disponible temporairement ici : [export-modified-groups-to-LDAP](#).
4. Il eut peut-être été plus intelligent de configurer externalSubjects.storage.ExternalSubjectStorable pour lire/écrire directement dans LDAP au lieu de synchroniser grouper vers LDAP.
5. La shibbolethisation de grouper est très limitée : les attributs shibboleth ne sont jamais utilisés (en dehors du REMOTE_USER). Le patch suivant ajoute la récupération des attributs shibboleth dans l'interface ExternalSubjectSelfRegister.

```
commit 74951b9d8ec4ea4e44028d66d35e4beba61a91d7
Author: Pascal Rigaux
Date: Mon Apr 22 16:39:15 2013 +0200
```

lite UI externalUsers: get shibboleth attributes if available + make those values read-only + allow to mark additional attrs "required" in UI

NB: when you set values read-only, they are omitted. Since we want them, ExternalSubjectSelfRegister need patching

```
diff --git a/conf/resources/grouper/nav.properties b/conf/resources/grouper/nav.properties
index b299b31..ba6c3fa 100644
--- a/conf/resources/grouper/nav.properties
+++ b/conf/resources/grouper/nav.properties
@@ -1608,24 +1608,31 @@ externalSubjectSelfRegister.register.field.identifie
r.tooltip=Login ID is detect

externalSubjectSelfRegister.register.field.name.label=Name
externalSubjectSelfRegister.register.field.name.tooltip=First and last name to show up in pick lists
+externalSubjectSelfRegister.register.field.name.shibAttr=displayName

externalSubjectSelfRegister.register.field.institution.label=Institution
externalSubjectSelfRegister.register.field.institution.tooltip=Name of the institution that identifies
you in pick lists

externalSubjectSelfRegister.register.field.email.label=Email
externalSubjectSelfRegister.register.field.email.tooltip=Email address where notifications will be sent
+externalSubjectSelfRegister.register.field.email.shibAttr=mail

externalSubjectSelfRegister.register.field.jabber.label=Jabber ID
externalSubjectSelfRegister.register.field.jabber.tooltip=Jabber ID for online chat

externalSubjectSelfRegister.register.field.givenname.label=Prenom
externalSubjectSelfRegister.register.field.givenname.tooltip=
+externalSubjectSelfRegister.register.field.givenname.shibAttr=givenName

externalSubjectSelfRegister.register.field.sn.label=Nom de famille
externalSubjectSelfRegister.register.field.sn.tooltip=
+externalSubjectSelfRegister.register.field.sn.required=true
+externalSubjectSelfRegister.register.field.sn.shibAttr=sn

externalSubjectSelfRegister.register.field.cn.label=Nom complet
externalSubjectSelfRegister.register.field.cn.tooltip=
+externalSubjectSelfRegister.register.field.cn.required=true
+externalSubjectSelfRegister.register.field.cn.shibAttr=cn

externalSubjectSelfRegister.submitButtonText=Submit
externalSubjectSelfRegister.submitButtonTooltip=Submit and save your information
diff --git a/java/src/edu/internet2/middleware/grouper/grouperUi/beans/externalSubjectSelfRegister
/ExternalRegisterContainer.java b/java/src/edu/internet2/middleware/grouper/grouperUi/beans
/externalSubjectSelfRegister/ExternalRegisterContainer.java
index a59d737..3da1583 100644
--- a/java/src/edu/internet2/middleware/grouper/grouperUi/beans/externalSubjectSelfRegister
/ExternalRegisterContainer.java
+++ b/java/src/edu/internet2/middleware/grouper/grouperUi/beans/externalSubjectSelfRegister
/ExternalRegisterContainer.java
@@ -159,10 +159,15 @@ public class ExternalRegisterContainer implements Serializable {

String label = GrouperUiUtils.message("externalSubjectSelfRegister.register.field.name.
```

```

label", true);
    String tooltip = GrouperUiUtils.message("externalSubjectSelfRegister.register.field.name.
tooltip", true);
+    String shib_val = getShibbolethValue("externalSubjectSelfRegister.register.field.name.
shibAttr");

    registerField.setLabel(label);
    registerField.setTooltip(tooltip);
-
+
+    if (shib_val != null) {
+        registerField.setValue(shib_val);
+        registerField.setReadOnly(true);
+    } else
        if (externalSubject != null) {
            registerField.setValue(externalSubject.getName());
        }
@@ -211,11 +216,16 @@ public class ExternalRegisterContainer implements Serializable {

    String label = GrouperUiUtils.message("externalSubjectSelfRegister.register.field.email.
label", true);
    String tooltip = GrouperUiUtils.message("externalSubjectSelfRegister.register.field.email.
tooltip", true);
+    String shib_val = getShibbolethValue("externalSubjectSelfRegister.register.field.email.
shibAttr");

    registerField.setLabel(label);
    registerField.setTooltip(tooltip);
    registerField.setParamName("param_email");
-
+
+    if (shib_val != null) {
+        registerField.setValue(shib_val);
+        registerField.setReadOnly(true);
+    } else
        if (externalSubject != null) {
            registerField.setValue(externalSubject.getEmail());
        }
@@ -234,10 +244,17 @@ public class ExternalRegisterContainer implements Serializable {

    String label = GrouperUiUtils.message("externalSubjectSelfRegister.register.field." +
registerField.getSystemName() + ".label", true);
    String tooltip = GrouperUiUtils.message("externalSubjectSelfRegister.register.field." +
registerField.getSystemName() + ".tooltip", true);
+    String required = getNavStringOrNull("externalSubjectSelfRegister.register.field." +
registerField.getSystemName() + ".required");
+    String shib_val = getShibbolethValue("externalSubjectSelfRegister.register.field." +
registerField.getSystemName() + ".shibAttr");

    registerField.setLabel(label);
    registerField.setTooltip(tooltip);
+    registerField.setRequired("true".equals(required));

+    if (shib_val != null) {
+        registerField.setValue(shib_val);
+        registerField.setReadOnly(true);
+    } else
        if (externalSubject != null) {
            ExternalSubjectAttribute externalSubjectAttribute = externalSubject.retrieveAttribute
(externalSubjectAttributeConfigBean.getSystemName(), false);
            if (externalSubjectAttribute != null) {
@@ -258,6 +275,38 @@ public class ExternalRegisterContainer implements Serializable {
    }
}

+    /* https://wiki.shibboleth.net/confluence/display/SHIB2
+NativeSPAttributeAccess#NativeSPAttributeAccess-Tool-SpecificExamples for java says: */
+    /* Shibboleth attributes are by default UTF-8 encoded.
+    However, depending on the servlet container configuration they are interpreted as ISO-8859-1
values.
+    This causes problems with non-ASCII characters. */

```

```

+     private String safe_interpret_as_UTF8(final String value) {
+         if (value == null) return null;
+         try {
+             return new String(value.getBytes("ISO-8859-1"), "UTF-8");
+         } catch (java.io.UnsupportedEncodingException e) {
+             return value;
+         }
+     }
+
+     private String getShibAttribute(String attrName) {
+         HttpServletRequest httpRequest = GrouperUiFilter.retrieveHttpRequest();
+         Object val = httpRequest.getAttribute(attrName);
+         return val != null ? safe_interpret_as_UTF8((String) val) : null;
+     }
+
+     private String getShibbolethValue(String key) {
+         String attrName = getNavStringOrNull(key);
+         return attrName == null ? null : getShibAttribute(attrName);
+     }
+
+     private String getNavStringOrNull(String key) {
+         try {
+             return GrouperUiUtils.message(key);
+         } catch (java.util.MissingResourceException e) {
+             return null;
+         }
+     }
+
+ /**
+  * get the identifier of the user logged in
+  * @return the identifier
diff --git a/java/src/edu/internet2/middleware/grouper/grouperUi/serviceLogic
/ExternalSubjectSelfRegister.java b/java/src/edu/internet2/middleware/grouper/grouperUi/serviceLogic
/ExternalSubjectSelfRegister.java
index 78eeaf3..8191631 100644
--- a/java/src/edu/internet2/middleware/grouper/grouperUi/serviceLogic/ExternalSubjectSelfRegister.java
+++ b/java/src/edu/internet2/middleware/grouper/grouperUi/serviceLogic/ExternalSubjectSelfRegister.java
@@ -416,12 +416,7 @@ public class ExternalSubjectSelfRegister {
     for (RegisterField registerField : externalRegisterContainer.getRegisterFields()) {

        String paramName = registerField.getParamName();
-       String paramValue = StringUtils.trimToNull(request.getParameter(paramName));
-
-       //skip readonly fields
-       if (registerField.isReadonly()) {
-           continue;
-       }
+       String paramValue = registerField.isReadonly() ? registerField.getValue() :
        StringUtils.trimToNull(request.getParameter(paramName));

        if (registerField.isRequired() && StringUtils.isBlank(paramValue)) {
            String theMessage = TagUtils.navResourceString("externalSubjectSelfRegister.
            fieldRequiredError");

```

6. Paramétrage de nuxeo pour gérer les externalPeople

```

--- a/templates/custom/config/default-ldap-groups-directory-bundle.xml
+++ b/templates/custom/config/default-ldap-groups-directory-bundle.xml
@@ -68,5 +68,62 @@
-->
    </references>
  </directory>
+ <directory name="ldapExternalGroupDirectory">
+
+     <!-- On utilise la connexion que l'on a definie dans default-ldap-users-bundle.xml -->
+     <server>default</server>
+
+     <!-- schema correspondant dans nuxeo, et identifiant des groupes (dans nuxeo pas dans
+     l'annuaire !) -->

```

```

+         <schema>group</schema>
+         <idField>groupname</idField>
+
+         <!-- branche dans laquelle sont les groupes -->
+         <searchBaseDn>ou=groups,dc=univ-paris1,dc=fr</searchBaseDn>
+
+         <!-- filtre de recherche -->
+         <searchFilter>(objectClass=groupOfNames)</searchFilter>
+
+         <!-- portee de la recherche -->
+         <searchScope>onelevel</searchScope>
+
+         <!-- Type de recherches possibles :
+             subinitial : toto => recherche sur toto*
+             subfinal   : toto => recherche sur *toto
+             subany      : toto => recherche sur *toto*
+         Par défaut la recherche est en subinitial -->
+         <substringMatchType>subany</substringMatchType>
+
+         <!-- si readOnly a false et connexion a l'annuaire avec des droits d'écriture, possibilite de
+         creation de groupes dans l'annuaire depuis nuxeo -->
+         <!-- <readOnly>true</readOnly>-->
+
+         <!-- cache en seconde -->
+         <cacheTimeout>3600</cacheTimeout>
+
+         <!-- nombre maximal d'entrees à mettre en cache -->
+         <cacheMaxSize>1000</cacheMaxSize>
+
+         <!-- utilisé si création de groupes dans l'annuaire depuis nuxeo -->
+         <creationBaseDn>ou=groups,dc=univ-paris1,dc=fr</creationBaseDn>
+         <creationClass>top</creationClass>
+         <creationClass>groupOfNames</creationClass>
+         <rdnAttribute>eduPersonPrincipalName</rdnAttribute>
+
+         <!-- mapping entre les attributs du schema groupe dans nuxeo et les attributs ldap -->
+         <fieldMapping name="groupname">cn</fieldMapping>
+         <fieldMapping name="grouplabel">description</fieldMapping>
+
+         <references>
+         <!-- LDAP reference resolve DNS embedded in uniqueMember attributes
+             If the target directory has no specific filtering policy, it is most
+             of the time not necessary to enable the 'forceDnConsistencyCheck' policy.
+             Enabling this option will fetch each reference entry to ensure its
+             existence in the target directory. -->
+
+             <ldapReference field="members" directory="ldapExternalUserDirectory"
+                 forceDnConsistencyCheck="false"
+                 staticAttributeId="member"
+             />
+         </references>
+         </directory>
+     </extension>
+ </component>
+ --- a/templates/custom/config/default-ldap-users-directory-bundle.xml
+ +++ b/templates/custom/config/default-ldap-users-directory-bundle.xml
+ @@ -77,7 +77,7 @@
+         <creationClass>person</creationClass>
+         <creationClass>organizationalPerson</creationClass>
+         <creationClass>inetOrgPerson</creationClass>
+ -         <rdnAttribute>eduPersonPrincipalName</rdnAttribute>
+ +         <rdnAttribute>uid</rdnAttribute>
+
+         <!-- Mapping entre le nom des champs dans le schema user de nuxeo et les attributs de l'annuaire -->
+         <fieldMapping name="username">eduPersonPrincipalName</fieldMapping>
+ @@ -90,7 +90,68 @@
+         <references>
+             <inverseReference field="groups" directory="groupDirectory" dualReferenceField="members" />
+         </references>
+     </directory>
+ <!--poure retrouver les externe de Paris 1 -->

```

```

+ <directory name="ldapExternalUserDirectory">
+
+ <!-- on s'appuie sur la connexion qu'on vient de définir -->
+ <server>default</server>
+
+ <!-- schema nuxeo utilisé : par défaut user -->
+ <schema>user</schema>
+
+ <!-- identifiant/mdp des personnes (dans nuxeo) -->
+ <idField>username</idField>
+ <passwordField>password</passwordField>
+
+ <!-- branche ldap dans laquelle sont situés les utilisateurs -->
+ <searchBaseDn>ou=externalPeople,dc=univ-paris1,dc=fr</searchBaseDn>
+
+ <!-- ObjectClass à rechercher => ajouté au filtre de recherche -->
+
+ <searchClass>eduPerson</searchClass>
+
+ <!-- Portée de la recherche -->
+ <searchScope>onelevel</searchScope>
+
+ <!-- Type de recherches possibles :
+         subinitial : toto => recherche sur toto*
+         subfinal   : toto => recherche sur *toto
+         subany      : toto => recherche sur *toto*
+         Par défaut la recherche est en subinitial -->
+ <substringMatchType>subinitial</substringMatchType>
+
+ <!-- Si False avec un binddn ayant des accès en écriture sur l'annuaire, proposera
+ d'ajouter des utilisateurs dans l'annuaire -->
+ <readOnly>true</readOnly>
+
+ <!-- cache timeout en secondes -->
+ <cacheTimeout>3600</cacheTimeout>
+
+ <!-- nombre maximal d'entrées en cache -->
+ <cacheMaxSize>1000</cacheMaxSize>
+
+ <!-- Maximum number of entries returned by the search -->
+ <querySizeLimit>20</querySizeLimit>
+
+ <!-- utilisé pour éventuellement créer des utilisateurs depuis nuxeo ... -->
+ <creationBaseDn>ou=externalPeople,dc=univ-paris1,dc=fr</creationBaseDn>
+ <creationClass>top</creationClass>
+ <creationClass>eduPerson</creationClass>
+ <creationClass>inetOrgPerson</creationClass>
+ <rdnAttribute>eduPersonPrincipalName</rdnAttribute>
+
+ <!-- Mapping entre le nom des champs dans le schema user de nuxeo et les attributs de l'annuaire -->
+ <fieldMapping name="username">eduPersonPrincipalName</fieldMapping>
+ <fieldMapping name="firstName">givenName</fieldMapping>
+ <fieldMapping name="lastName">sn</fieldMapping>
+ <fieldMapping name="company">supannetablissement</fieldMapping>
+ <fieldMapping name="email">mail</fieldMapping>
+
+ <!-- référence aux groupes, cf. default-ldap-groups-directory-bundle.xml -->
+ <references>
+   <inverseReference field="groups" directory="externalGroupDirectory" dualReferenceField="
members" />
+ </references>
+ </directory>
+ </extension>
+ </component>
+ --- a/templates/custom/config/default-virtual-groups-bundle.xml
+++ b/templates/custom/config/default-virtual-groups-bundle.xml
@@ -34,6 +34,7 @@
+ <groups>
+   <!-- définition du répertoire des groupes -->
+   <directory>groupDirectory</directory>
+   <directory>externalGroupDirectory</directory>

```

```

        </groups>

        <!-- uid ldap de l'administrateur -->
--- a/templates/custom/config/multi-directory-config.xml
+++ b/templates/custom/config/multi-directory-config.xml
@@ -18,6 +18,12 @@
    <source name="ldapUserDirectory">
      <subDirectory name="ldapUserDirectory"/>
    </source>
+    <!-- déclaration de la source ldap pour les externes -->
+
+    <source name="ldapExternalUserDirectory">
+      <subDirectory name="ldapExternalUserDirectory"/>
+    </source>
+
    <!-- declaration de la source locale que nous allons definir dans un autre point d'extension -->
    <source name="sqlUserDirectory" creation="true">
@@ -42,11 +48,22 @@
    <source name="ldapGroupDirectory">
      <subDirectory name="ldapGroupDirectory"/>
    </source>
-
    <!-- declaration de la source locale definie dans un autre fichier -->
    <source name="sqlGroupDirectory" creation="true">
      <subDirectory name="sqlGroupDirectory"/>
    </source>
  </directory>
+
+  <directory name="externalGroupDirectory">
+    <!-- declaration de la source ldap externe pour les groupes -->
+    <source name="ldapExternalGroupDirectory">
+      <subDirectory name="ldapExternalGroupDirectory"/>
+    </source>
+    <!-- declaration de la source locale definie dans un autre fichier -->
+    <source name="sqlGroupDirectory" creation="true">
+      <subDirectory name="sqlGroupDirectory"/>
+    </source>
+  </directory>
+
  </extension>
</component>

```

7. Paramétrage de uportal pour gérer les externalPeople

```

--- a/update/uPortal/uportal-impl/src/main/resources/properties/contexts/personDirectoryContext.xml
+++ b/update/uPortal/uportal-impl/src/main/resources/properties/contexts/personDirectoryContext.xml
@@ -172,6 +172,7 @@
    @bg.use.db.persondirs@<ref bean="cachinguPortalJdbcAttributeSource"/>@end.use.
db.persondirs@
    <!-- <ref bean="cachinguPortalJdbcUserSource"/> -->
    @bg.use.ldap.persondirs@<ref bean="cachinguPortalLdapUserSource"/>@end.use.ldap.
persondirs@
+    <ref bean="cachinguPortalLdapExternalUserSource"/>
+    </list>
+    </property>
+    </bean>
@@ -377,6 +378,36 @@
+    </bean>
+    </property>
+    </bean>
+
+    <bean id="cachinguPortalLdapExternalUserSource" class="org.jasig.services.persondir.support.
CachingPersonAttributeDaoImpl">
+        <property name="usernameAttributeProvider" ref="usernameAttributeProvider" />
+        <property name="cacheNullResults" value="true" />
+        <property name="userInfoCache">
+            <bean class="org.jasig.portal.utils.cache.MapCacheFactoryBean">
+                <property name="cacheFactory" ref="cacheFactory" />
+                <property name="cacheName" value="org.jasig.services.persondir.USER_INFO.
ldap_external_person_dir" />
+            </bean>
+        </property>
+        <property name="cacheKeyGenerator" ref="userAttributeCacheKeyGenerator" />
+        <property name="cachedPersonAttributesDao" >
+            <bean class="org.jasig.services.persondir.support.ldap.LdapPersonAttributeDao">
+                <property name="contextSource" ref="legacyLdapContext" />
+                <property name="baseDN" value="ou=externalPeople,dc=univ-paris1,dc=fr" />
+                <property name="queryAttributeMapping">
+                    <map>
+                        <entry key="username" value="eduPersonPrincipalName" />
+                    </map>
+                </property>
+                <property name="resultAttributeMapping">
+                    <map>
+                        <entry key="memberOf" value="memberOf" />
+                    </map>
+                </property>
+            </bean>
+        </property>
+    </bean>
+
+    @end.use.ldap.persondirs@
    <bean id="userAttributeCacheKeyGenerator" class="org.jasig.services.persondir.support.
AttributeBasedCacheKeyGenerator">
        <property name="useAllAttributes" value="true" />

```