

Proposition d'organisation des groupes

Attention : Ceci est juste une piste exploratoire ! Document en cours d'élaboration

Symboles utilisés :

- + : dossier
- : groupe alimenté de manière automatique depuis le SI
- = : groupe alimenté manuellement
- > : inclusion de groupe

Branches (dossiers de premier niveau rattachés à la racine de grouper)

branche 'admin' :

Les administrateurs grouper ; un seul groupe : "grouper" (on avait mis comme nom de groupe 'globaux', 'grouper' me semble plus approprié)
Un dossier "applis" permettent de déclarer les administrateurs ayant le droit de gérer des droits d'accès aux applications.
Ce dossier contient différents groupes : un groupe par application (décrite dans la branche "applis").
Les membres d'un groupe ont le droit de gérer les membres du ou des groupes liés à l'appli concernée.
en pratique, seuls les membres du groupe 'admin:grouper' peuvent gérer (manuellement) les groupes et dossiers de cette branche.

branche 'applis' :

Contient des dossiers représentant chacun une appli. Les groupes d'un dossier permet de gérer des privilèges (roles) sur l'appli concernée.

branche 'etab' :

Contient l'ensemble des groupes institutionnels d'un établissement

branche 'attributs' :

Cette branche va contenir des groupes issus d'informations élémentaires ; il ne seront pas utilisés directement, mais vont permettre de construire des groupes grouper 'composites', issus de combinaison de groupes de cette branche.

branche 'collab' :

Pour des groupes collaboratifs ; à creuser.

Structure d'une instance grouper d'un établissement

```

+ admin
  = grouper          (admin:grouper)
  + applis
    = appli1        (admin:applis:appli1)
    = appli2        (admin:applis:appli2)
+ applis             # les privilèges (roles) sur les applis
+ ldap_compat
  + appli1
    = role1         (applis:ldap_compat:appli1:role1)
    = role2         (applis:ldap_compat:appli1:role2)
  + appli2
    = tous          (applis:ldap_compat:appli2:tous)
+ grouper compat    # éviter de multiplier les entrées de grosses populations
+ portail
  + profil
+ etab
  - tous            (etab:tous) #contient (etab:pers:tous), (etab:etud:tous), ...
                    # est-ce utile et nécessaire ? Faut-il publier dans LDAP ?
+ pers
  - tous            (etab:pers:tous)
                    -> attribs:employeeType:member
                    # faut-il publier dans LDAP ? Les applis ayant accès LDAP n'ont-elle pas plus facile à
                    # utiliser une requete LDAP sur attributs ?
+ comp
  + comp1
    - tous          (etab:pers:comp:comp1:tous)
    + comp1.1       # ex : ur1:pers:comp:cri:proxi
    - tous          (etab:pers:comp:comp1:comp1.1:tous)
    + comp1.1.1     # ex : ur1:pers:comp:cri:proxi:centre
    - tous
    ....
    + comp2
+ equipes-rech     # les équipes de recherche
  - eqr1
  - eqr2

+ etud
  - tous
    -> attribs:employeeType:student
+ comp
  - comp1
    -> vet1 + vet5
  - comp2
    -> vet1 + vet7
+ formations
  - vet1 (etudiants licence 1 maths formation initiale)
  - vet2
+ attributs
  + employeeType
    - empT1
    - empT2
+ collab
....

```