Rappels LDAP

Benoit Métrot
benoit.metrot@math.univ-poitiers.fr

UMR 7348 - Laboratoire de Mathématiques et Applications (Poitiers)

ANF Développement logiciel pour l'administration système et réseau
Angers, 23 mai 2012

Plan

Progression

Organisation de l'information

- Structure arborescente
  - Racine
  - Branches
  - Feuille ou entrée
- Types de données spécifiés par un schéma
- Structuration de l'information selon un modèle clé/valeur

Comparaison avec une base de données

- Opérations de lecture largement majoritaires
- Optimisé pour une lecture rapide
- Types de données extensibles
- Diffusion plus large
- Plus grande duplication de l'information

SupAnn

- Groupes de travail dépendant du CRU [1]
- Etudie la construction d'annuaires pour les établissements de l'enseignement supérieur
- Rédaction de recommandations
  - SupAnn v1 (2003)
  - SupAnn 2008
  - SupAnn 2009
- Construction d'un schéma d'annuaire spécifique normalisé
- http ://www.cru.fr/documentation/supann/index

1. CRU : Comité Réseau des Universités intégré à Renater depuis juin 2011

UMR 7348
B. Métrot

Objectifs de SupAnn

- Cohérence dans la mise en oeuvre des annuaires des établissements d'enseignement supérieur
- Favoriser la portabilité des logiciels
- Aboutir à une homogénéité des contenus entre les établissements

Progression

Historique

Norme X500

- Conçu par les opérateurs téléphoniques pour échanger leurs annuaires
- Construit sur le modèle réseau OSI
- Annuaire à grande échelle et distribué
- Elle définit
  - Les règles de nommage des objets et des entités
  - Les protocoles pour fournir le service d'annuaire
  - Un mécanisme d'authentification
- Implémentations lourdes, difficiles et buggés

Historique

LDAP

- Création à l'Université du Michigan en 1993
  - Simplification du protocle DAP (X500)
  - Adaptation à TCP/IP
- Repris par l'IETF pour aboutir à la définition de plusieurs RFC en 1995
  - *Lightweight Directory Access Protocol* (RFC 1777)
  - *The String Representation of Standard Attribute Syntaxes* (RFC 1778)
  - *A String Representation of Distinguished Names* (RFC 1779)
- Apparition de LDAPv3 en 2002 avec la RFC 3377 *Lightweight Directory Access Protocol (v3) : Technical Specification*

Modèle d'information

Fournit les structures et les types de données

- Directory Information Tree (DIT) $\rightarrow$ Arbre d'information LDAP
- L'entrée, élément de base de l'annuaire
  - Instance d'une ou plusieurs classes d'objet (*objectClass*)
  - Attributs facultatifs ou obligatoires
  - Une ou plusieurs valeurs par attribut
- Schéma d'annuaire
  - Type de données des attributs
  - Règles de codage
  - Règles de comparaison

Modèle de nommage

RDN et DN

→ Comment référencer de façon unique les entrées et les données du DIT

RDN : Relative Distinguished Name

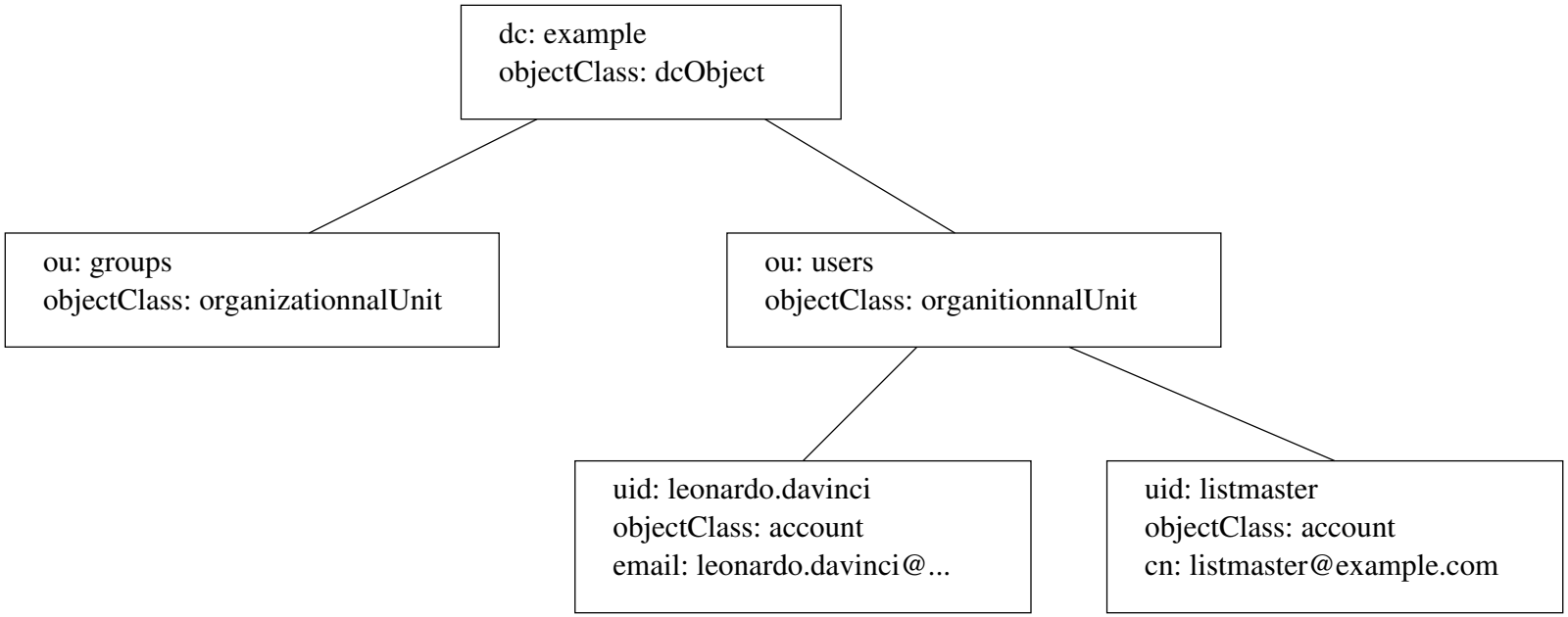
Attribut unique qui distingue une entrées des autres issues d'un même parent.

DN : Distinguished Name

Identifie de façon unique les entrées de l'arbre. S'obtient par agrégation des RDN suivant le chemin de l'entrée voulue jusqu'à la racine.

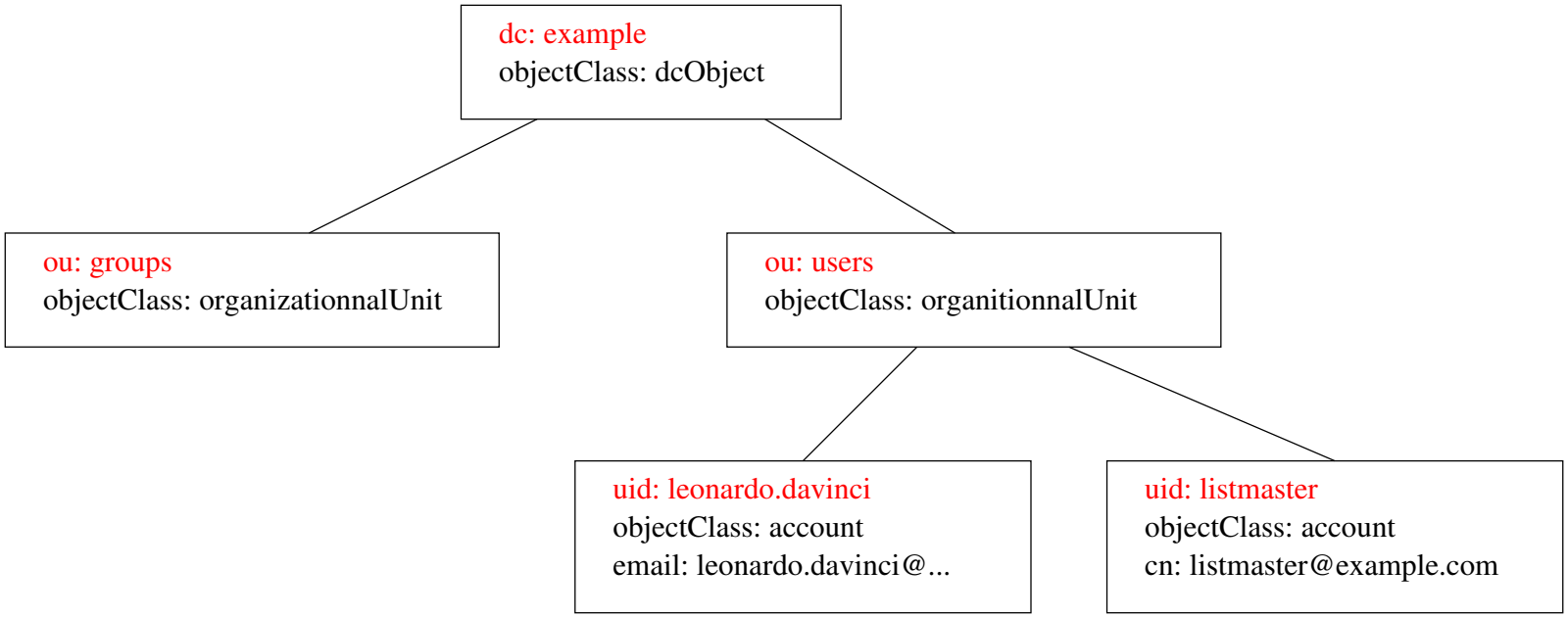
Modèle de nommage

Exemple



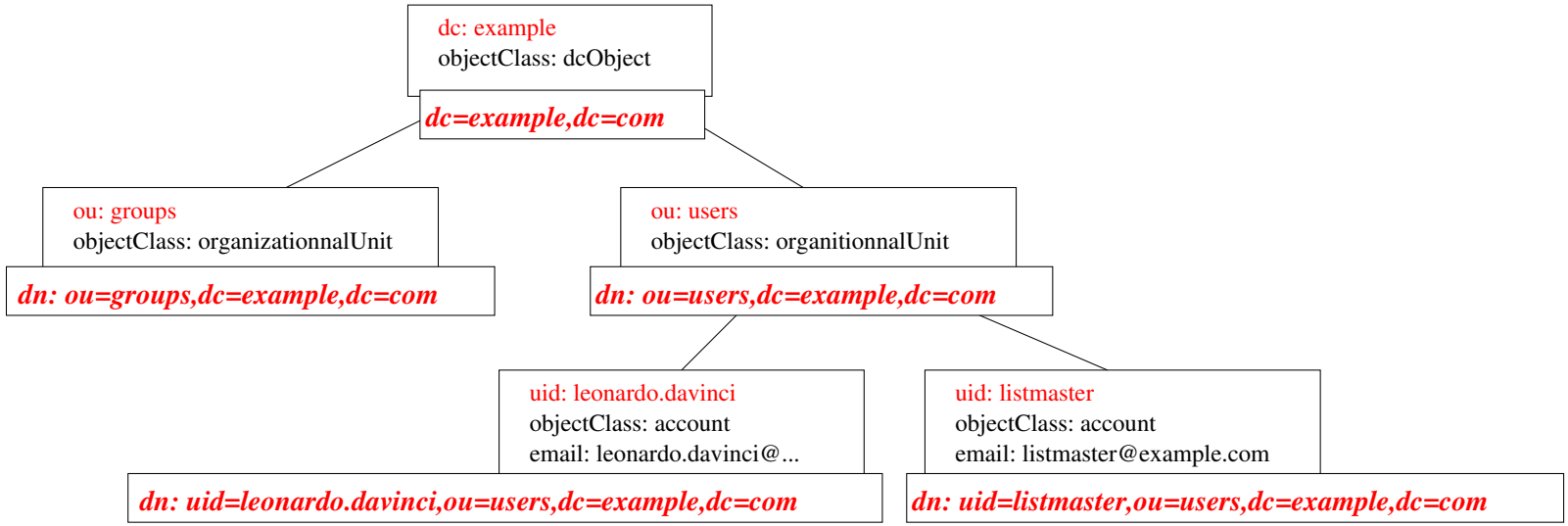
Modèle de nommage

RDN



Modèle de nommage

DN



Modèle fonctionnel

Protocole d'accès à l'annuaire

- Mécanisme d'authentification
- Opérations sur les entrées
  - Recherche
  - Ajout
  - Suppresion
  - Modification
  - ...

Modèle de sécurité

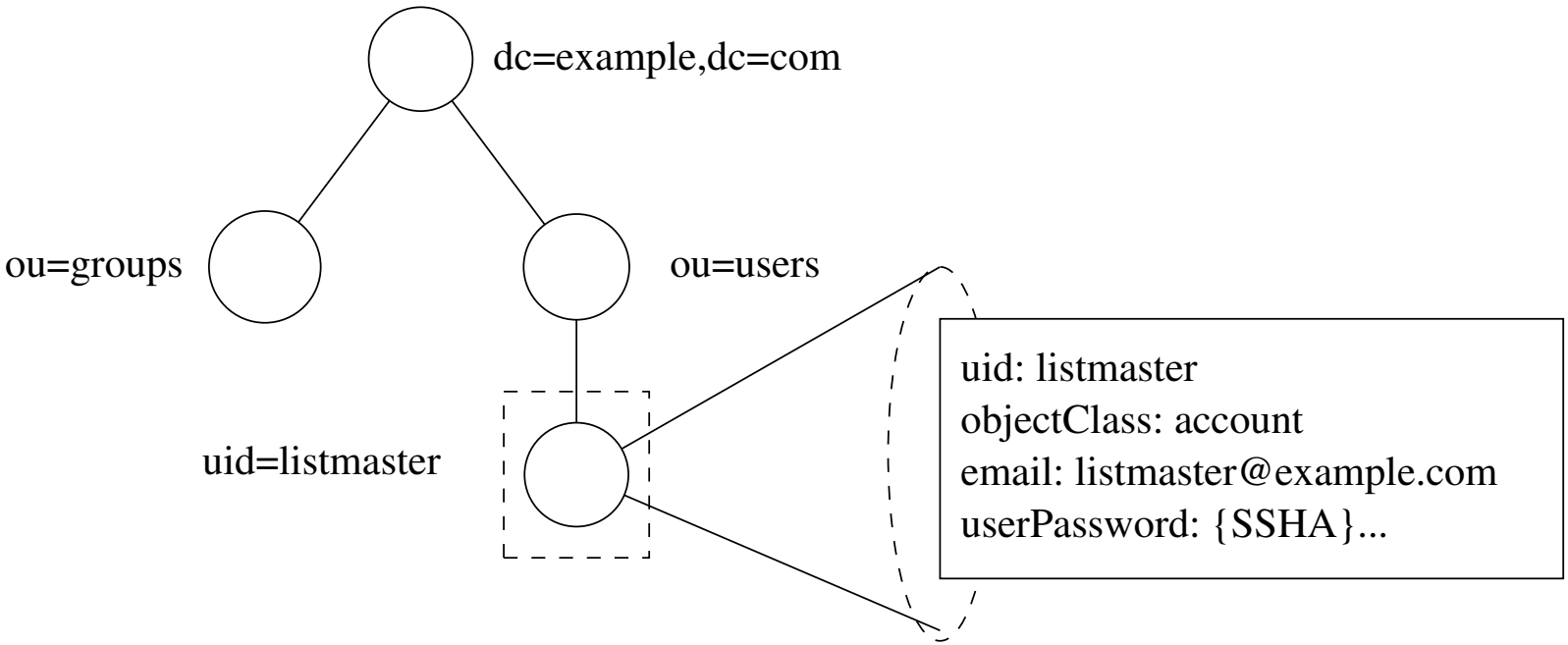
- Authentifier les clients (preuve d'identité)
- Controler l'accès aux données pour les clients authentifiés

Objets LDAP

Arbre

Directory Information Tree (DIT)

- Classification hierarchique des entrées
- Chaque noeud est une entrée de l'annuaire
- La racine de l'arbre caractérise l'annuaire (suffixe ou base LDAP)



Objets LDAP

Classe d'objet

- Détermine le type d'objet enregistré dans une entrée
- Définit les attributs de l'entrée
- Trois natures de classes :
  - Classe abstraite
  - Classe structurelle
  - Classe auxiliaire
- Une entrée peut instancier au plus une classe structurelle et plusieurs classes auxiliaires

Objets LDAP

Une entrée, des attributs

- Modelise un objet stocké dans l'annuaire
- Les classes d'objet associées à l'entrée définissent ses attributs
- Attributs obligatoires ou facultatifs
- Attributs mono-valués (une seule valeur) ou multi-valués (plusieurs valeurs)

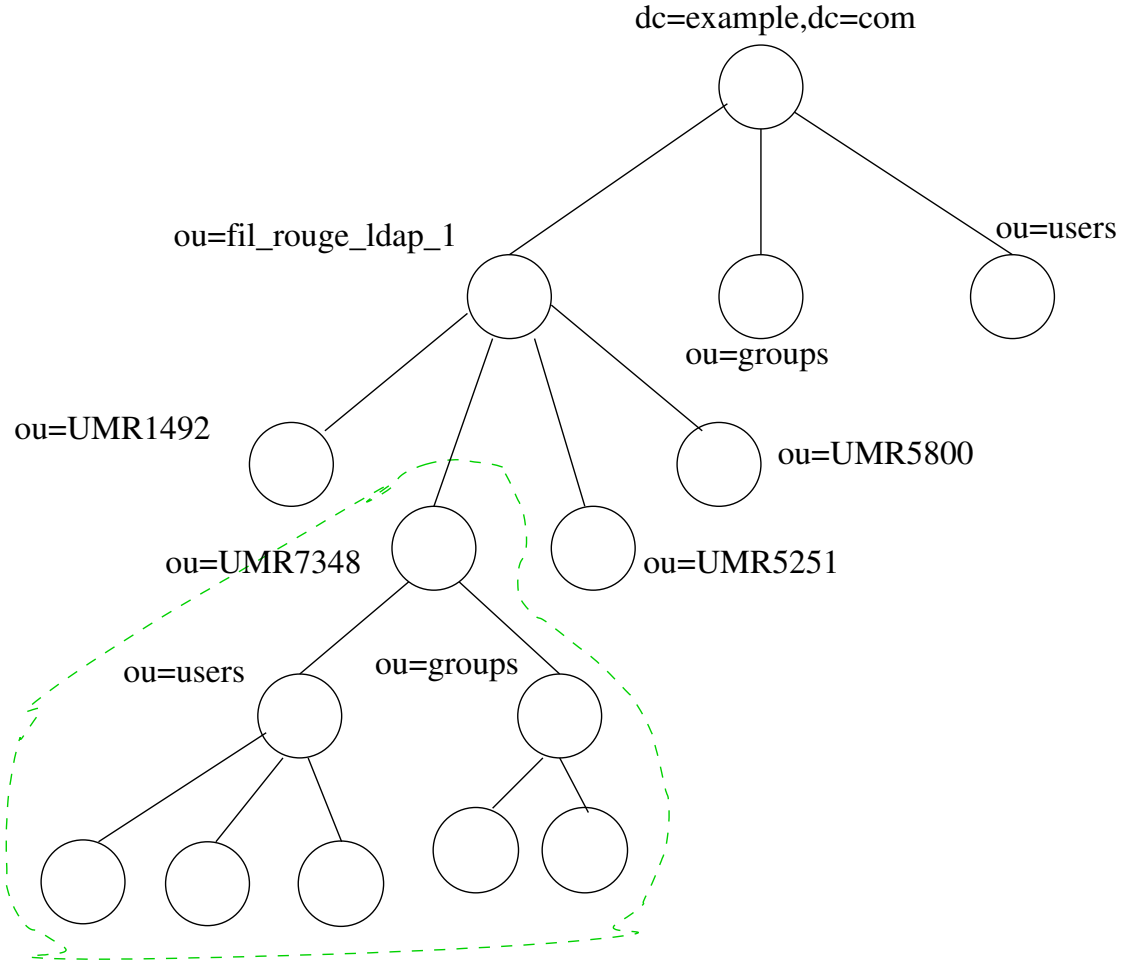
Session LDAP

1. Etablissement de la connexion TCP (négociation SSL ou TLS)
2. Authentification (bind) avec un DN ou anonyme
3. Envoi de la requete par le client (opération de recherche, mise à jour...)
4. Envoi des résultats des la requête par le serveur au client (ensemble d'entrées)
5. Fermeture de la connexion

Recherche

Base de recherche

Désigne le sous-arbre de l'annuaire à considérer

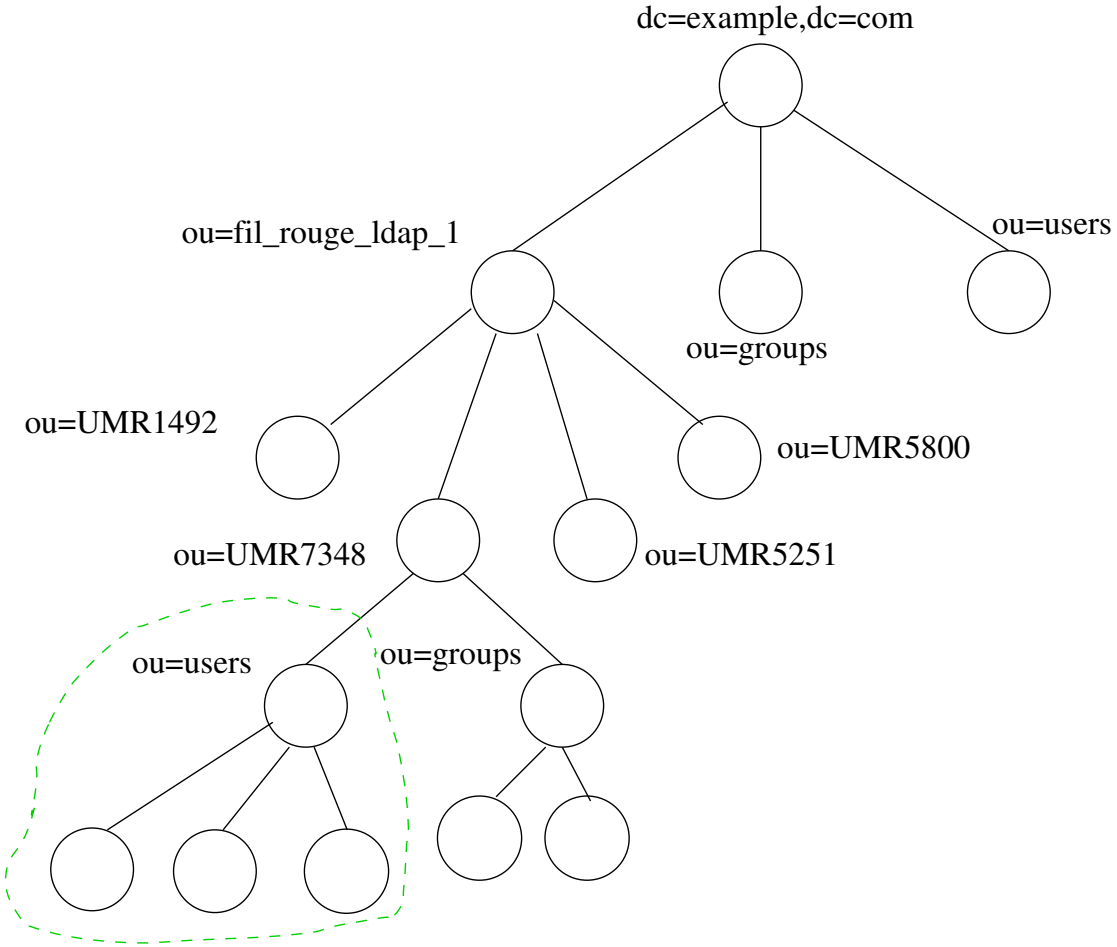


dn: ou=UMR7348,ou=fil_rouge_ldap_1,dc=example,dc=com

Recherche

Base de recherche

Désigne le sous-arbre de l'annuaire à considérer

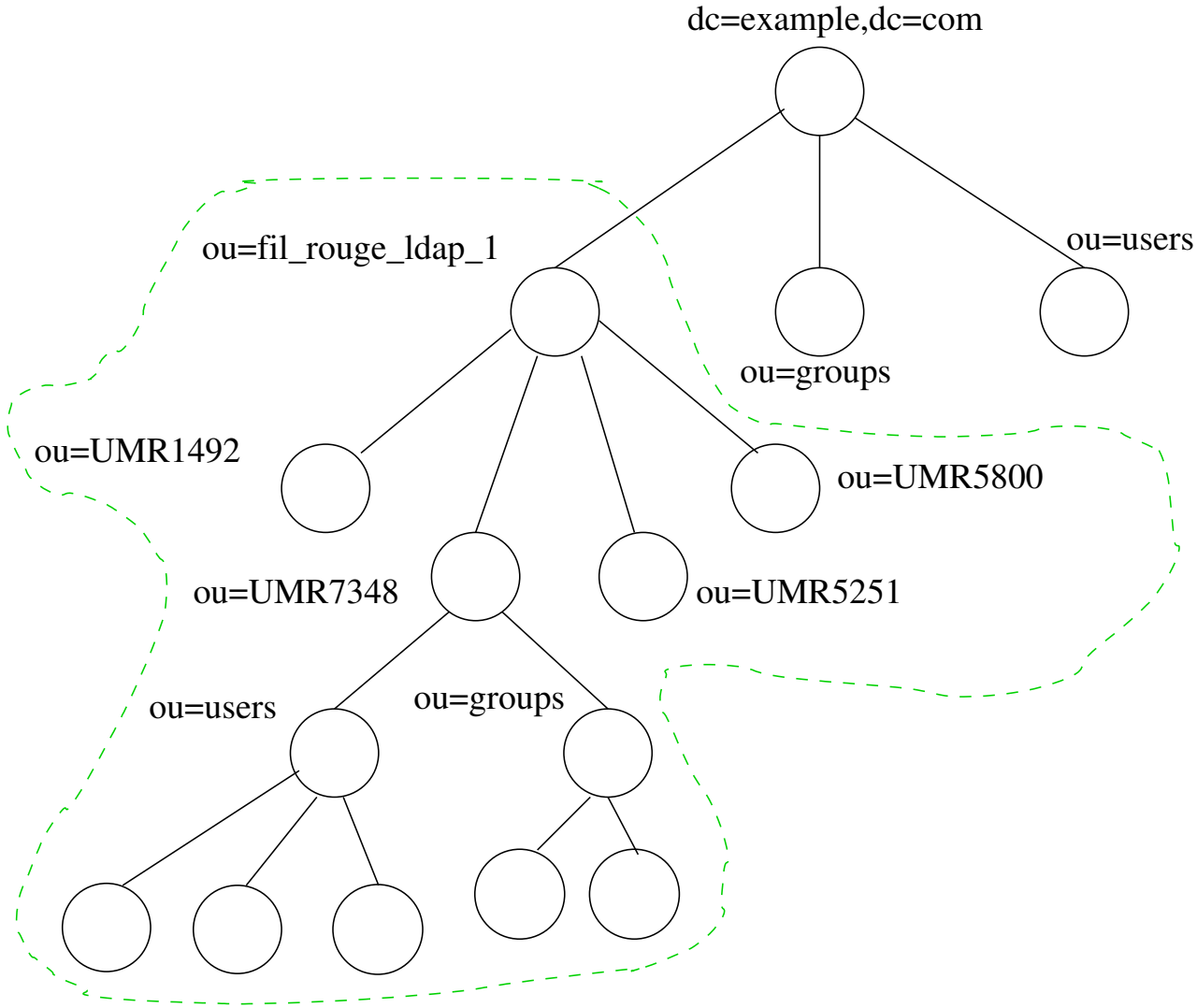


dn: ou=users,ou=UMR7348,ou=fil_rouge_ldap_1,dc=example,dc=com

Recherche

Base de recherche

Désigne le sous-arbre de l'annuaire à considérer

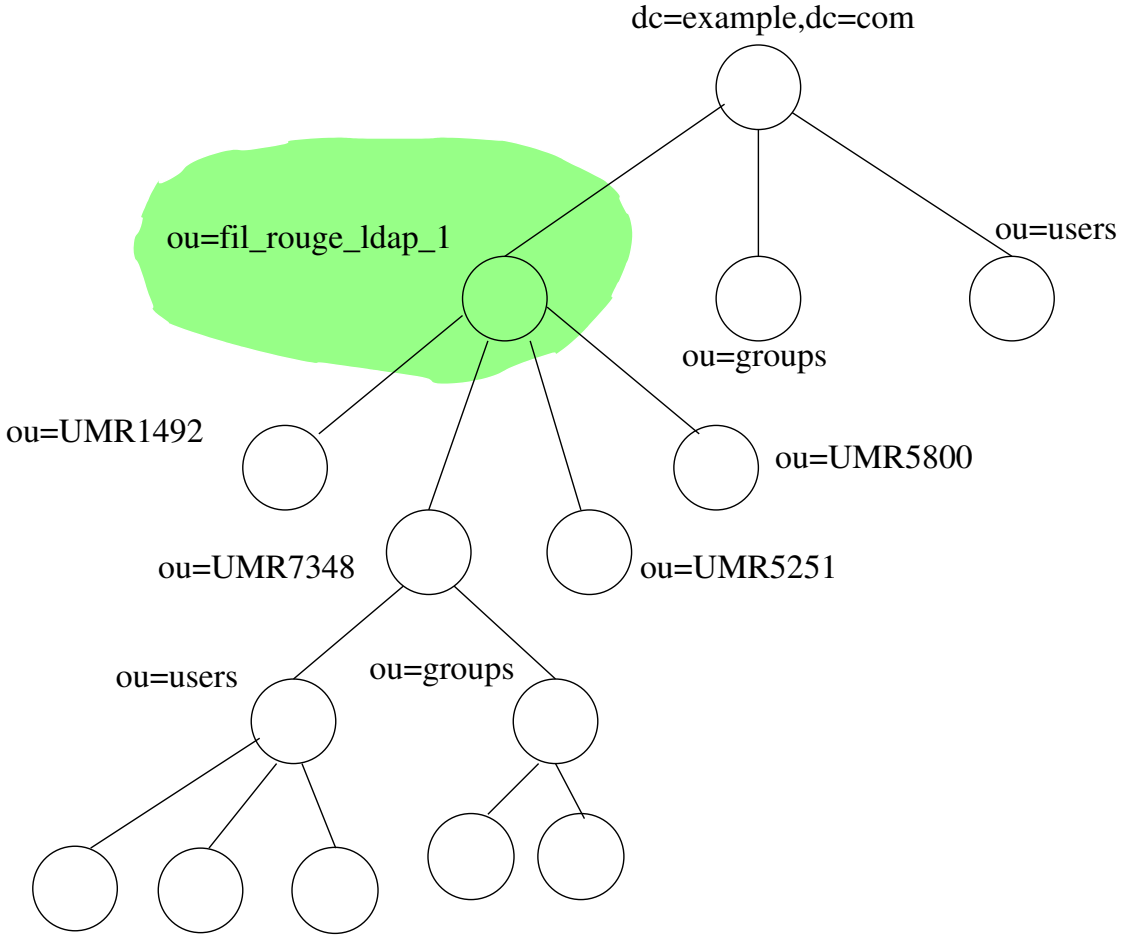


dn: ou=fil_rouge_ldap_1,dc=example,dc=com

Portée de recherche

base

Détermine la profondeur de recherche dans le sous-arbre

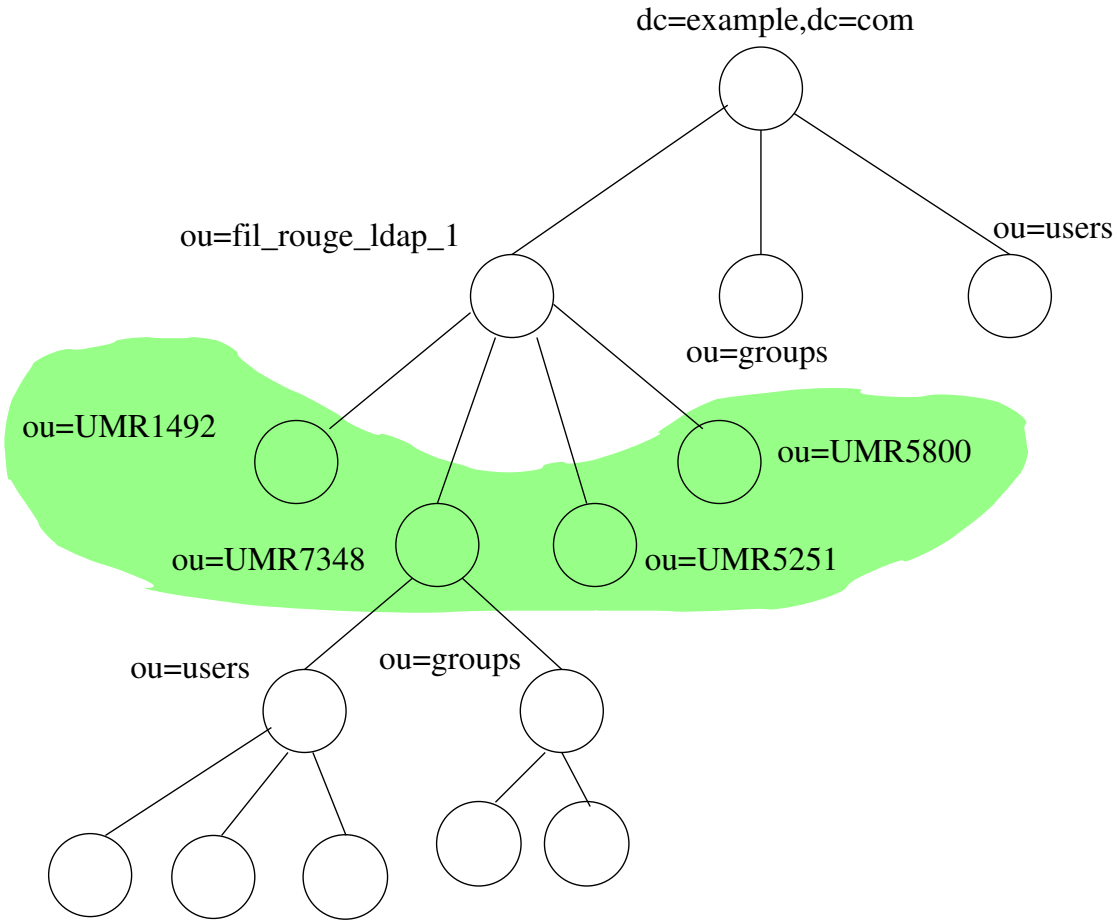


Base de recherche : ou=fil_rouge_ldap_1,dc=example,dc=com

Portée de recherche

one

Détermine la profondeur de recherche dans le sous-arbre

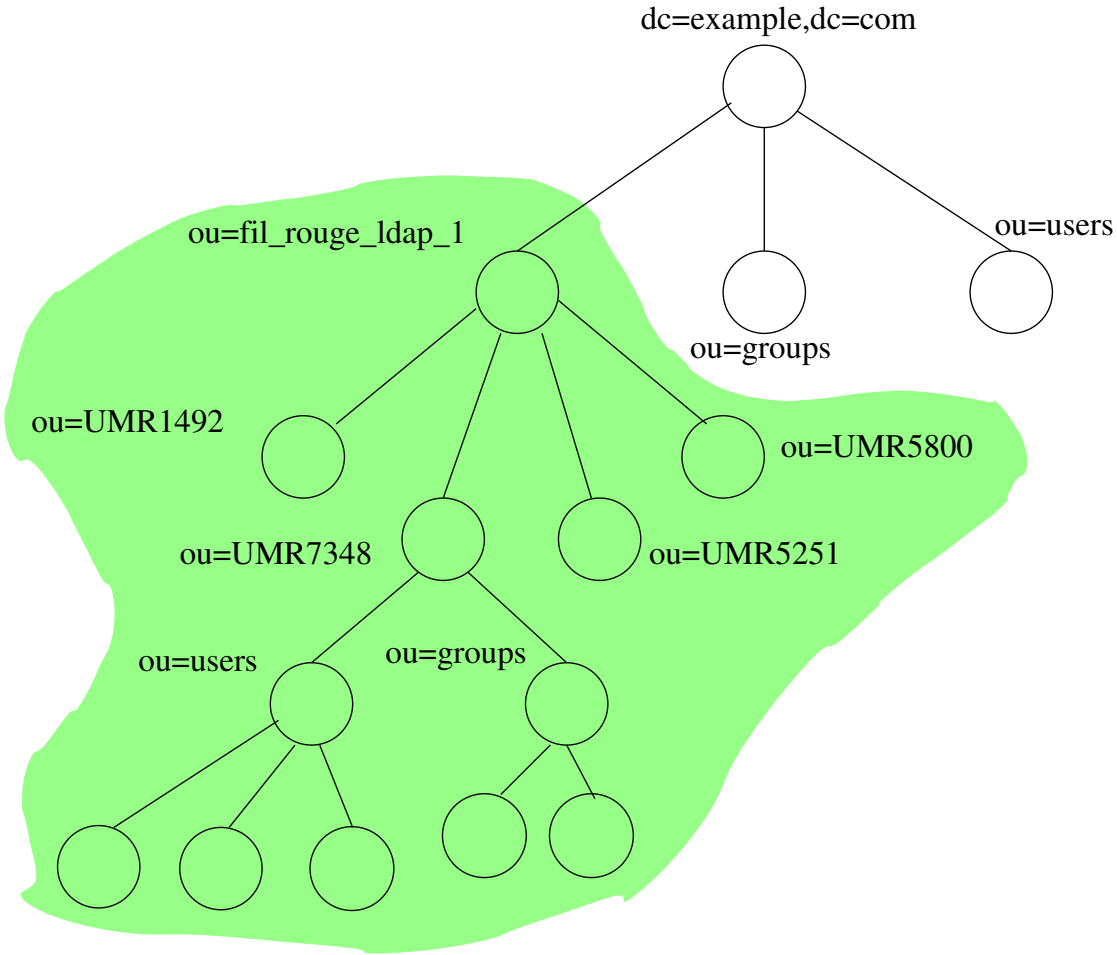


Base de recherche : ou=fil_rouge_ldap_1,dc=example,dc=com

Portée de recherche

sub

Détermine la profondeur de recherche dans le sous-arbre

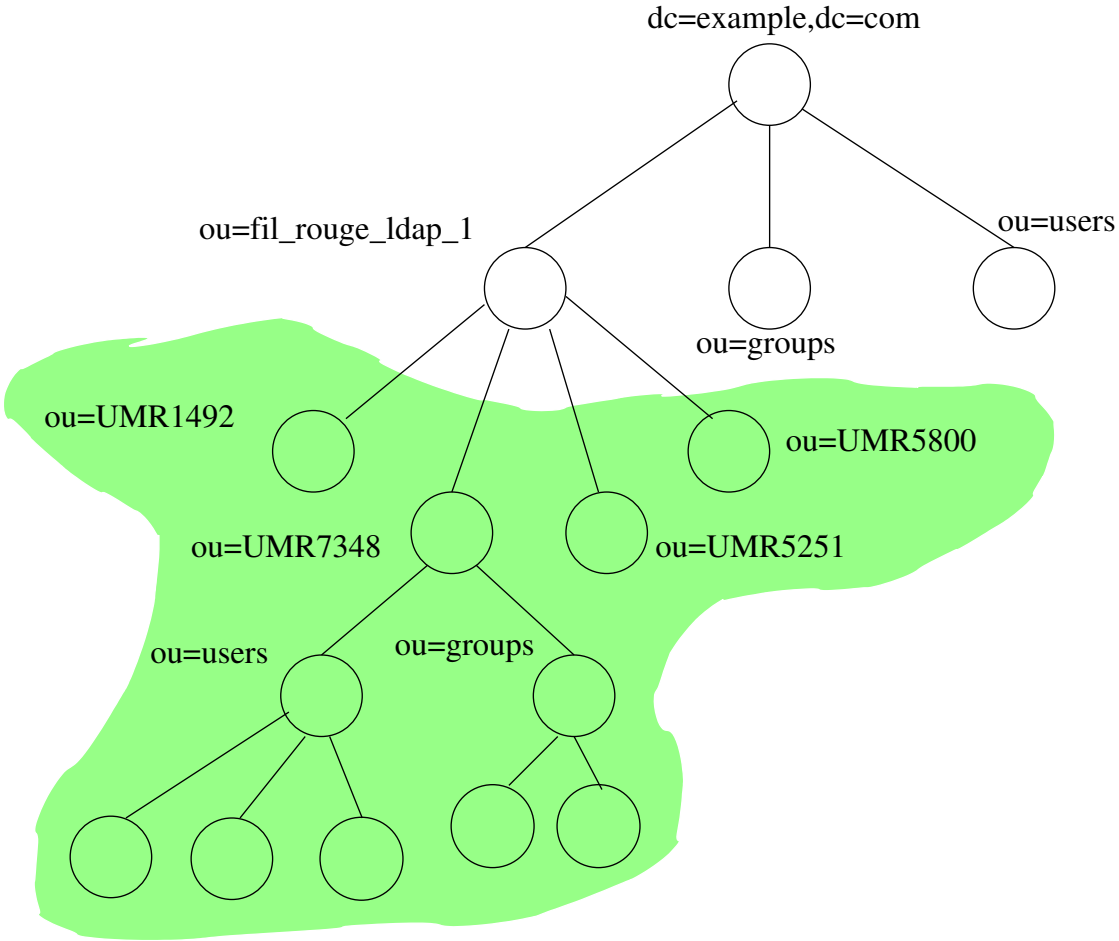


Base de recherche : ou=fil_rouge_ldap_1,dc=example,dc=com

Portée de recherche

children

Détermine la profondeur de recherche dans le sous-arbre



Base de recherche : ou=fil_rouge_ldap_1,dc=example,dc=com

Filtre de recherche

- Chaque critère est de la forme *(attribut<cmp>valeur)*
- Les critères *(<op><critère1><critère2>)* s'assemblent avec des opérateurs logiques

Comparaison *<cmp>*	
=	Egalité
≅	Approximation
<=	Inférieur ou égal
>=	Supérieur ou égal

Op. logique *<op>*	
&	Et
|	Ou
!	Négation

Exemple : toutes les entrées

`(objectClass=*)`

Filtre de recherche

- Chaque critère est de la forme *(attribut<cmp>valeur)*
- Les critères *(<op><critère1><critère2>)* s'assemblent avec des opérateurs logiques

Comparaison *<cmp>*	
=	Egalité
≅	Approximation
<=	Inférieur ou égal
>=	Supérieur ou égal

Op. logique *<op>*	
&	Et
|	Ou
!	Négation

Exemple : les entrées de comptes POSIX avec une adresse mail

```
(&(objectClass=posixAcount)(mail=*))
```

Filtre de recherche

- Chaque critère est de la forme *(attribut<cmp>valeur)*
- Les critères *(<op><critère1><critère2>)* s'assemblent avec des opérateurs logiques

Comparaison *<cmp>*	
=	Egalité
≅	Approximation
<=	Inférieur ou égal
>=	Supérieur ou égal

Op. logique *<op>*	
&	Et
|	Ou
!	Négation

Exemple : les entrées dont le nom complet (cn) commence par M ou T

```
(|(cn=M*)(cn=T*))
```

Filtre de recherche

- Chaque critère est de la forme *(attribut<cmp>valeur)*
- Les critères *(<op><critère1><critère2>)* s'assemblent avec des opérateurs logiques

Comparaison *<cmp>*	
=	Egalité
≅	Approximation
<=	Inférieur ou égal
>=	Supérieur ou égal

Op. logique *<op>*	
&	Et
|	Ou
!	Négation

Exemple : les entrées dont le prénom ne contient pas ben

```
(!(givenName=*ben*))
```

LDIF

- Fichier texte (utf-8) pour l'échange d'entrées LDAP
- Ligne de commentaire commence par #
- Les entrées sont séparées par une ligne vide
- Chaque ligne représente un attribut avec sa valeur
- Un attribut peut s'écrire sur plusieurs lignes à condition qu'elles commancent par un et un seul espace

```
dn: ou=people,dc=example,dc=com
ou: people
objectClass: organizationalUnit

dn: ou=groups,dc=example,dc=com
ou: groups
objectClass: organizationalUnit
```

Opérations sur les entrées

Ajout

```
dn: cn=user.no_1,ou=users,ou=UMR7348,
 ou=fil_rouge_ldap_1,dc=example,dc=com
changetype: add
objectClass: organizationalPerson
objectClass: person
objectClass: inetOrgPerson
cn: user.no_1
sn: user.no_1
mail: user.no_1@example.com
uid: user.no_1
userPassword: {SSHA}LeHashDuPassword
```

Opérations sur les entrées

Suppression

```
dn: cn=user.no_1,ou=users,ou=UMR7348,
 ou=fil_rouge_ldap_1,dc=example,dc=com
changetype: delete
```

Opérations sur les attributs

Ajout

```
dn: cn=user.no_1,ou=users,ou=UMR7348,
 ou=fil_rouge_ldap_1,dc=example,dc=com
changetype: modify
add: telephoneNumber
telephoneNumber: +33 5 49 49 49 49
```

Opérations sur les attributs

Modification

```
dn: cn=user.no_1,ou=users,ou=UMR7348,
 ou=fil_rouge_ldap_1,dc=example,dc=com
changetype: modify
replace: telephoneNumber
telephoneNumber: +33 5 94 94 94 94
```

Opérations sur les attributs

Suppression

```
dn: cn=user.no_1,ou=users,ou=UMR7348,
 ou=fil_rouge_ldap_1,dc=example,dc=com
changetype: modify
delete: telephoneNumber
```

Progression

Coté serveur

- OpenLDAP (slapd)
- Apache Directory Server

Coté client

- Comandes OpenLDAP pour intéragir avec l'annuaire
  - *ldapadd*
  - *ldapmodify*
  - *ldapsearch*
  - ...
- Bibliothèque dynamique cliente configurable via */etc/ldap/ldap.conf*
- LdapVI
- GQ
- Apache Directory Studio

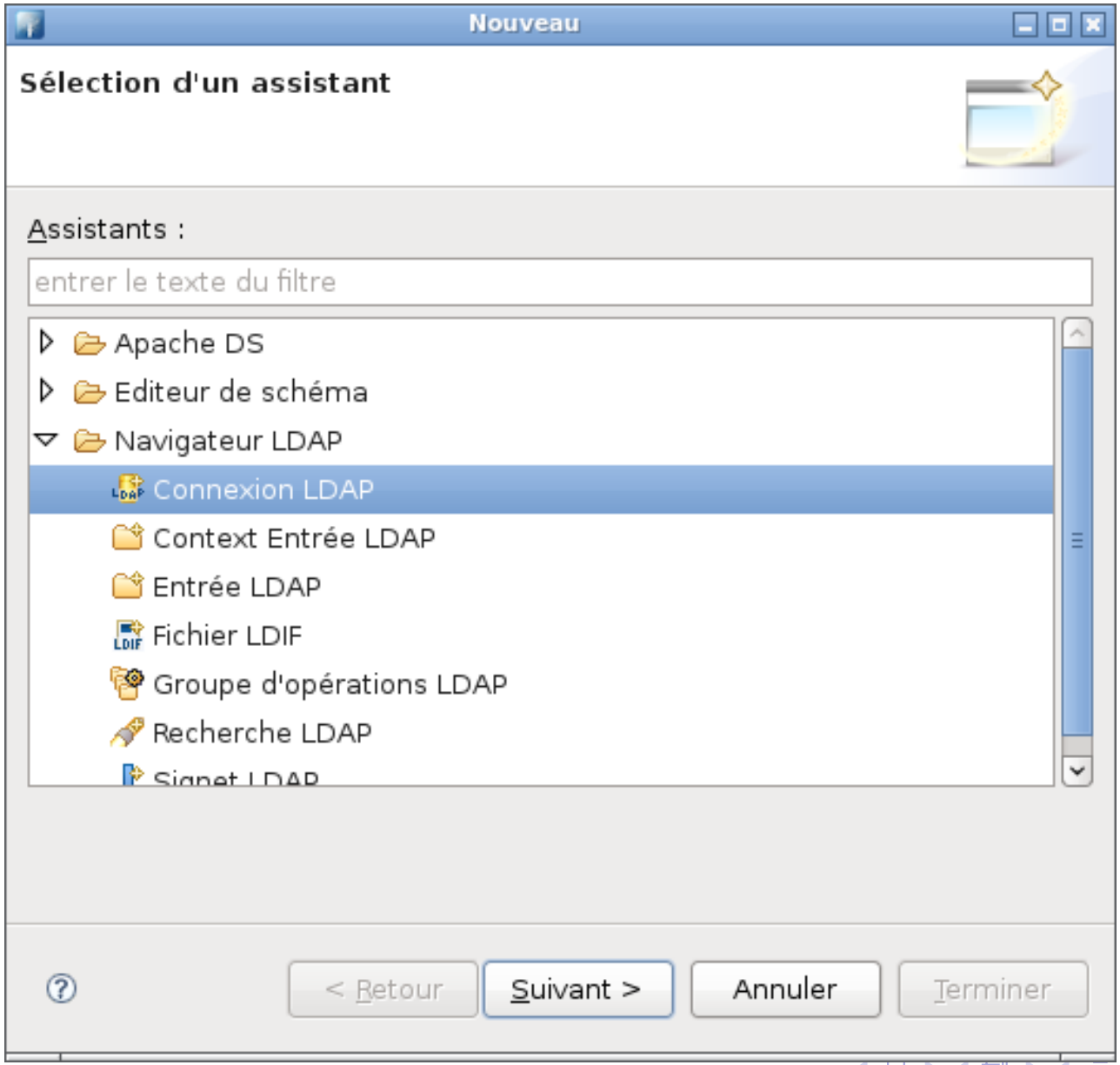
Méthode d'authentification

- Annonyme
- Simple $\rightarrow$ Identifiant + Mot de passe
- SASL / GSSAPI $\rightarrow$ Kerberos

UMR 7348
B. Métrot

Apache Directory Studio

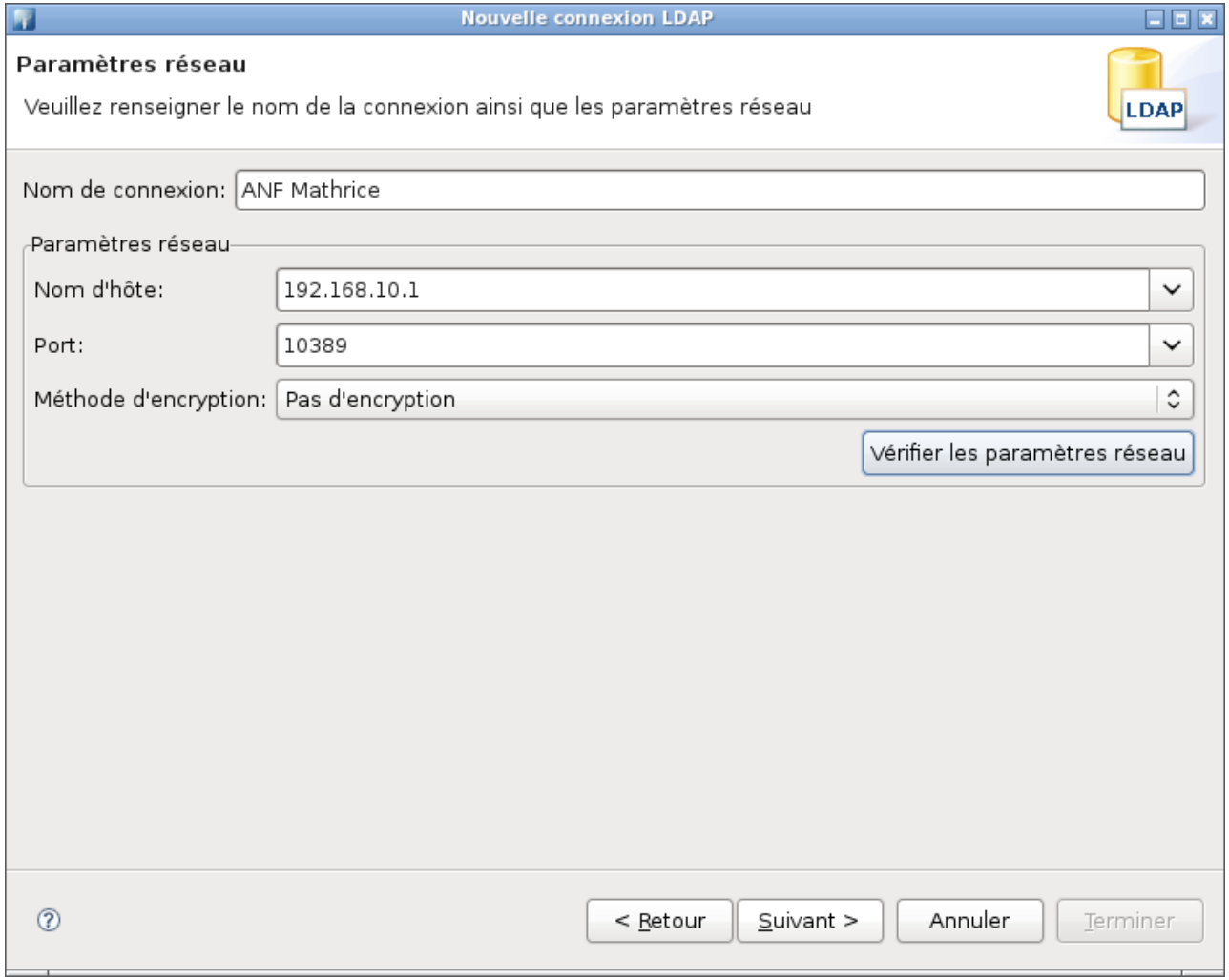
Nouvelle connexion d'annuaire



UMR 7348
B. Métrot

Apache Directory Studio

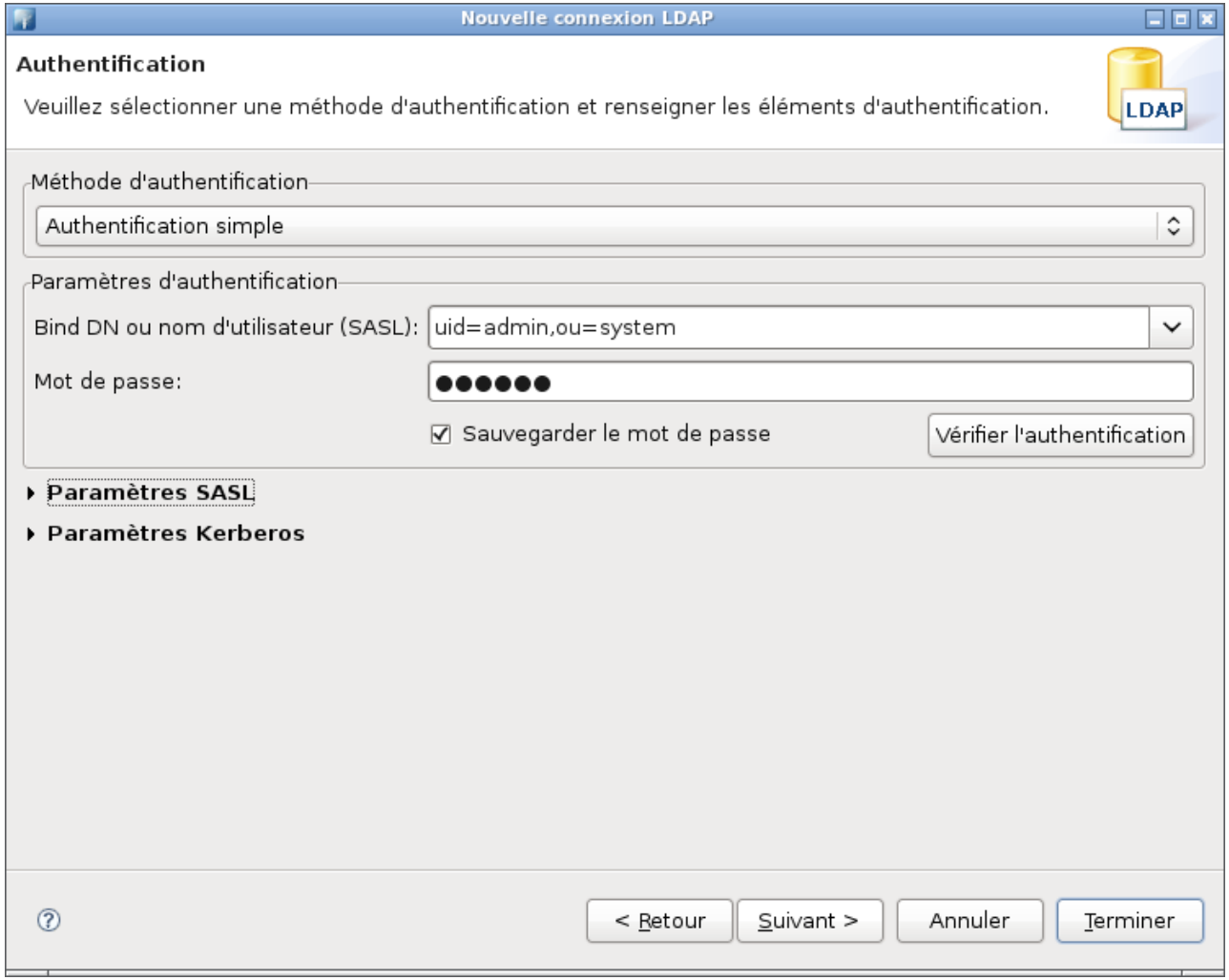
Nouvelle connexion d'annuaire



UMR 7348
B. Métrot

Apache Directory Studio

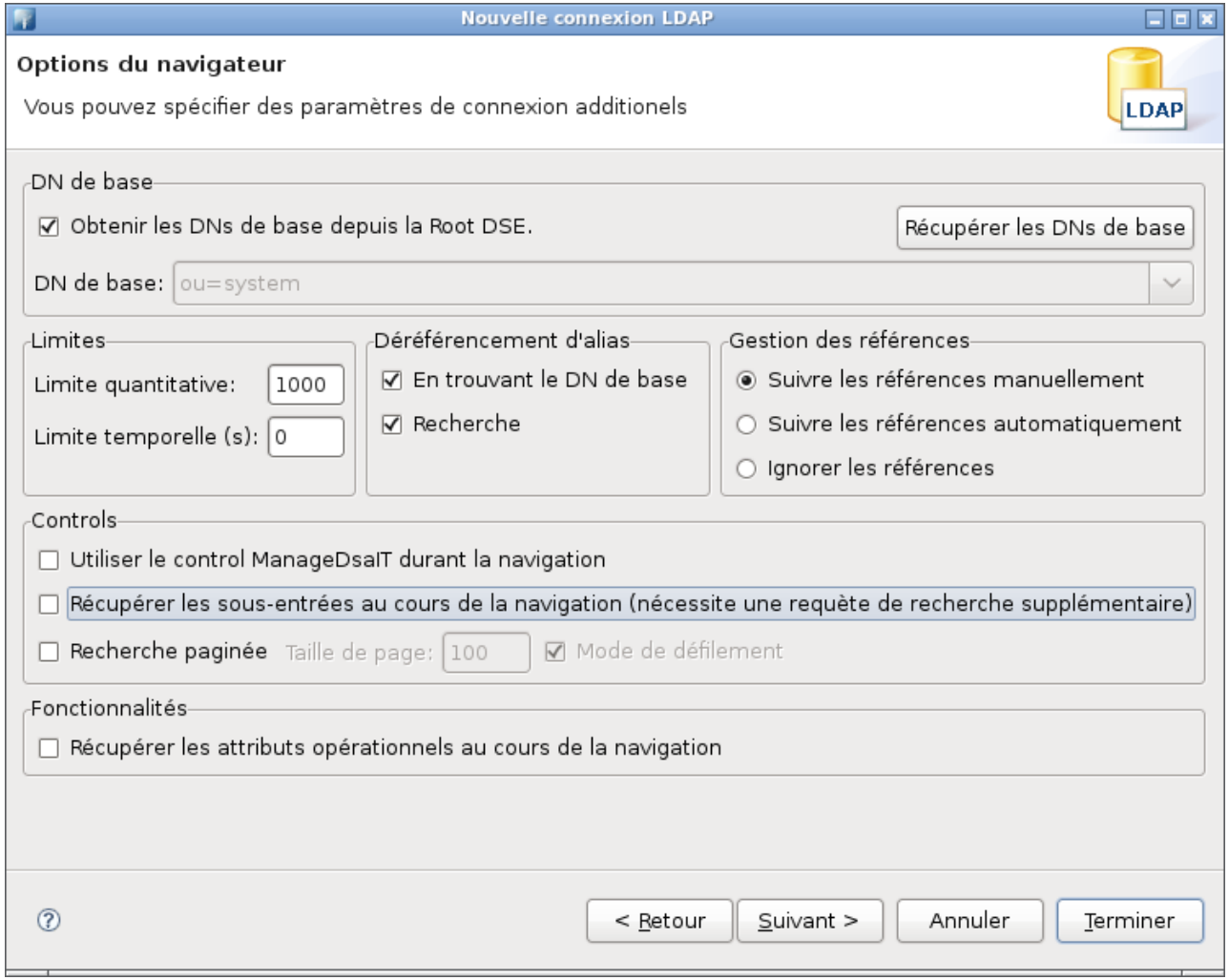
Nouvelle connexion d'annuaire



UMR 7348
B. Métrot

Apache Directory Studio

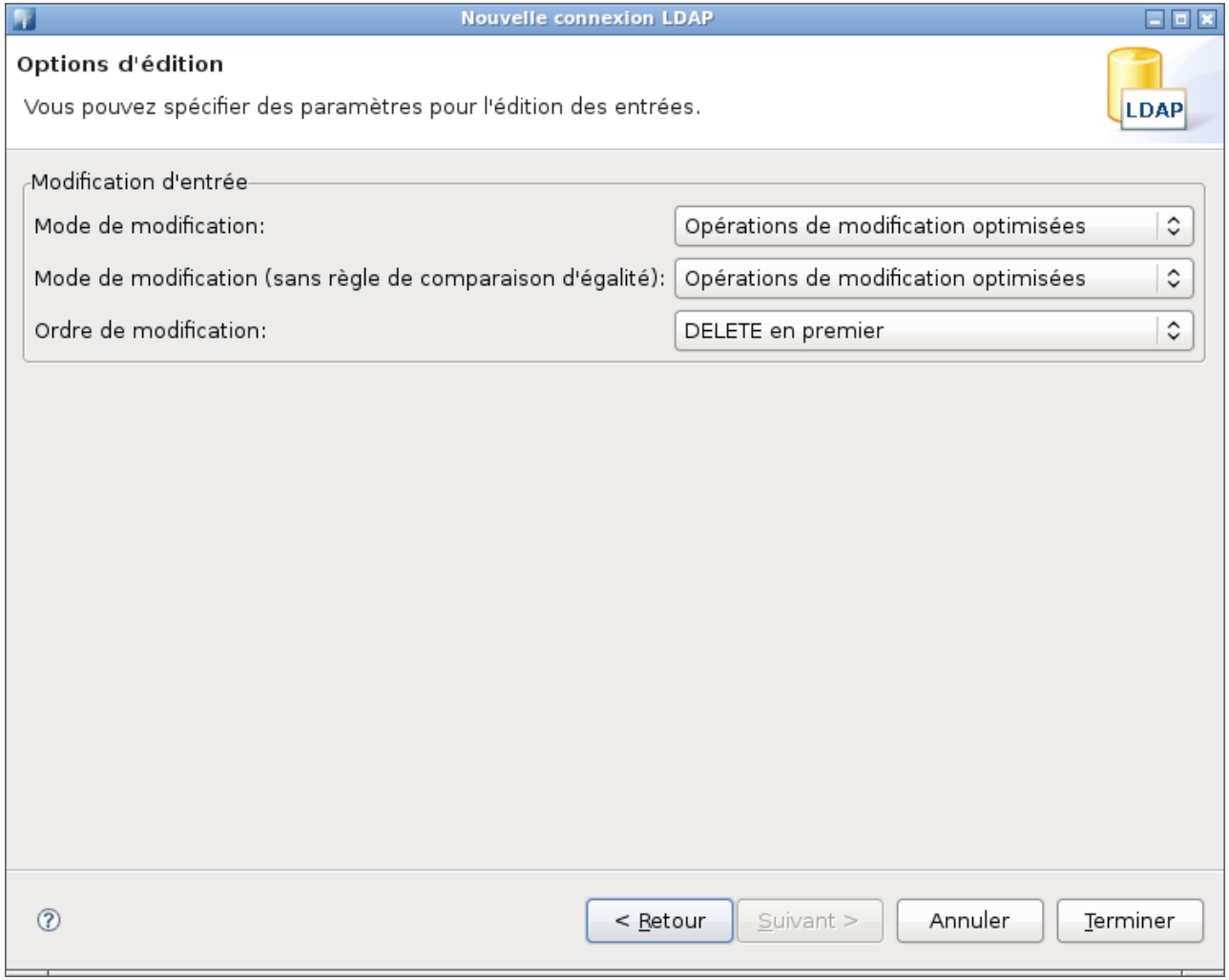
Nouvelle connexion d'annuaire



UMR 7348
B. Métrot

Apache Directory Studio

Nouvelle connexion d'annuaire



UMR 7348
B. Métrot

Apache Directory Studio

Fenêtre principale