
# Principes de base des annuaires & interfaçage applicatif

## Réunion Xstra - 02/03/2006

Alain ZAMBONI

# Plan

- Principes de base
- L'implémentation OpenLDAP
- Interfaçage applicatif
- Conclusion

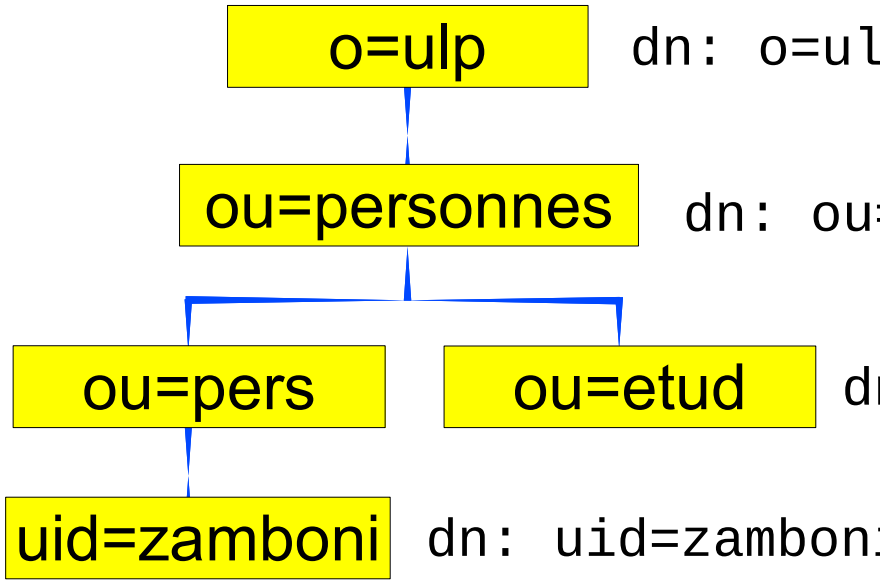
# LDAP : concept

- Lightweight Directory Acces Protocol (RFC 2251)
- Notion d'« annuaire »
  - Conteneur organisé d'information
  - Succède au protocole X.500 de l'OSI
- Définit une méthode d'accès aux données
  - Et non pas le stockage des données
- Souplesse de la représentation des données
- Intégrer la possibilité de réplication
- Accès sécurisé : LDAPS

# Structuration d'un annuaire

- Structure arborescente
- Un noeud = une « **entrée** »
  - Entrée composée d'**attributs**
    - Paire clés/valeurs
  - indexée par un **dn** : « distinguished name » (RFC 2253)
    - Unique dans l'annaire
- 2 type d'attributs :
  - **User** attribute : attributs « normaux », manipulés par l'utilisateur
  - **Opérationnal** attribute : attributs systèmes, manipulés par le serveur

# Structuration d'un annuaire




```
sn: ZAMBONI
givenName: Alain
mail: az@mondomaine.fr
…
createTimestamp: 20051012150802Z
modifyTimestamp: 20051012150802Z
```

# Définition des attributs

- Données informatives ou applicatives sur l'entrée
- Entrée « typée » par une ou plusieurs **objectClass**
  - *STRUCTURAL*, *ABSTRACT* ou *AUXILIARY*
  - Attributs **MUST** (obligatoires) ou **MAY** (facultatifs)
- Attributs aussi définis : type, opérations, multivalué
- Héritage de classes et d'attributs
- Description des attributs et objectClass : **schéma**
  - Schéma standards : RFC 2256
  - Possibilité de définir ses propres schémas

# Branche « Personnes »

```
dn: uid=zamboni,ou=pers,ou=personnes,o=ulp
objectClass: person
sn: ZAMBONI
cn: ZAMBONI Alain
userPassword: {crypt}$1$hoiifla5$efopefn

### fichier core.schema ###
objectclass ( 2.5.6.6 NAME 'person'
   SUP top
   STRUCTURAL
   MUST ( sn $ cn )
   MAY ( userPassword $ telephoneNumber $
           seeAlso $ description ) )
```

# Branche « Personnes »

```
### fichier core.schema ###

attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
   DESC 'RFC2256: last (family) name(s) for which
     the entity is known by'
   SUP name )

attributetype ( 2.5.4.41 NAME 'name'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringsMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.15{32768} )
```

# Opérations LDAP

- Opérations usuelles
  - Bind : authentification (*bindDn*)
  - Search : recherche d'entrée (*baseDN* + filtre)
  - Add : ajout d'entrée
  - Delete : suppression d'entrée
  - Modify : modifications d'entrée
  - ModifyDN : modification du dn d'une entrée (*rdn* + *newSuperior*)
- Autres moins utiles
  - Unbind, abandon, compare

# Filtres de recherche

- Filtre sur les valeurs des attributs (RFC 2254)
- Opérateurs : =  ~=  >=  <=
- Opérateurs booléens : &  |  !
- Caractères spéciaux :  ()  \  *
- Chaque opération doit être parenthèsée
- Opérations inexistantes :
  - != ⇒ ( ! (attribut=valeur) )
  - > ⇒ ( ! (attribut<=valeur) )
  - < ⇒ ( ! (attribut>=valeur) )

# Filtres de recherche

- ( & (objectclass=person) (telephoneNumber=*) )
  - Personne avec un No de tel renseigné
- ( & (objectClass=person) ( | (sn=*z) (sn=z*) ) )
  - Personne dont le sn commence ou termine par z
- ( & (objectClass=person) ( ! (sn=zamboni) ) ( ! (telephoneNumber=*) ) )
  - Personne avec avec sn != zamboni et sans No de tél.
- ( & (objectclass=person) (cn~=schmit) )
  - Schmit, schmidt, schmitt, schmitteckert, schmutz,...

# Syntaxe LDIF

- LDAP Data Interchange Files (RFC 2849)
- Représentation des attributs des entrées
  - Attribut: valeur
- Opérations LDAP représentée en LDIF
  - dn
  - Changetype : ( add | delete | modify | modrdn )
    - ( replace | add | delete ) : attribut  # pour modify
  - Attribut : valeur
  - Modrdn un peu différent

# Syntaxe LDIF

```
dn: uid=zamboni,ou=pers,ou=personnes,o=ulp
objectClass: person
sn: ZAMBONI
cn: ZAMBONI Alain
userPassword: {crypt}$1$hoiifla5$efopefn
mail: Alain.Zamboni@mondomaine.fr
mail: zamboni@mondomaine.fr
mail: az@mondomaine.fr
objectClass: posixAccount
uidNumber: 1000
gidNumber: 1000
homeDirectory: /home/zamboni
```

# Syntaxe LDIF

```
dn: uid=zamboni,ou=pers,ou=personnes,o=ulp
changetype: add
objectClass: person
uid: zamboni
sn: ZAMBONI
cn: ZAMBONI Alain
userPassword: {crypt}$1$hoiifla5$efopefn
mail: Alain.Zamboni@mondomaine.fr

dn: uid=zamboni,ou=pers,ou=personnes,o=ulp
changeType: modrdn
newrdn: uid=zambono
deleteOldRdn: 1
newSuperior: ou=pers,ou=personnes,o=ulp
```

# Syntaxe LDIF

```
dn: uid=zambono,ou=pers,ou=personnes,o=ulp
changeType: modify
add: mail
mail: alain.Zambono@mondomaine.fr
-
delete: mail
mail: Alain.Zamboni@mondomaine.fr
-
replace: sn
sn: ZAMBONO

dn: uid=zambono,ou=pers,ou=personnes,o=ulp
changetype: delete
```

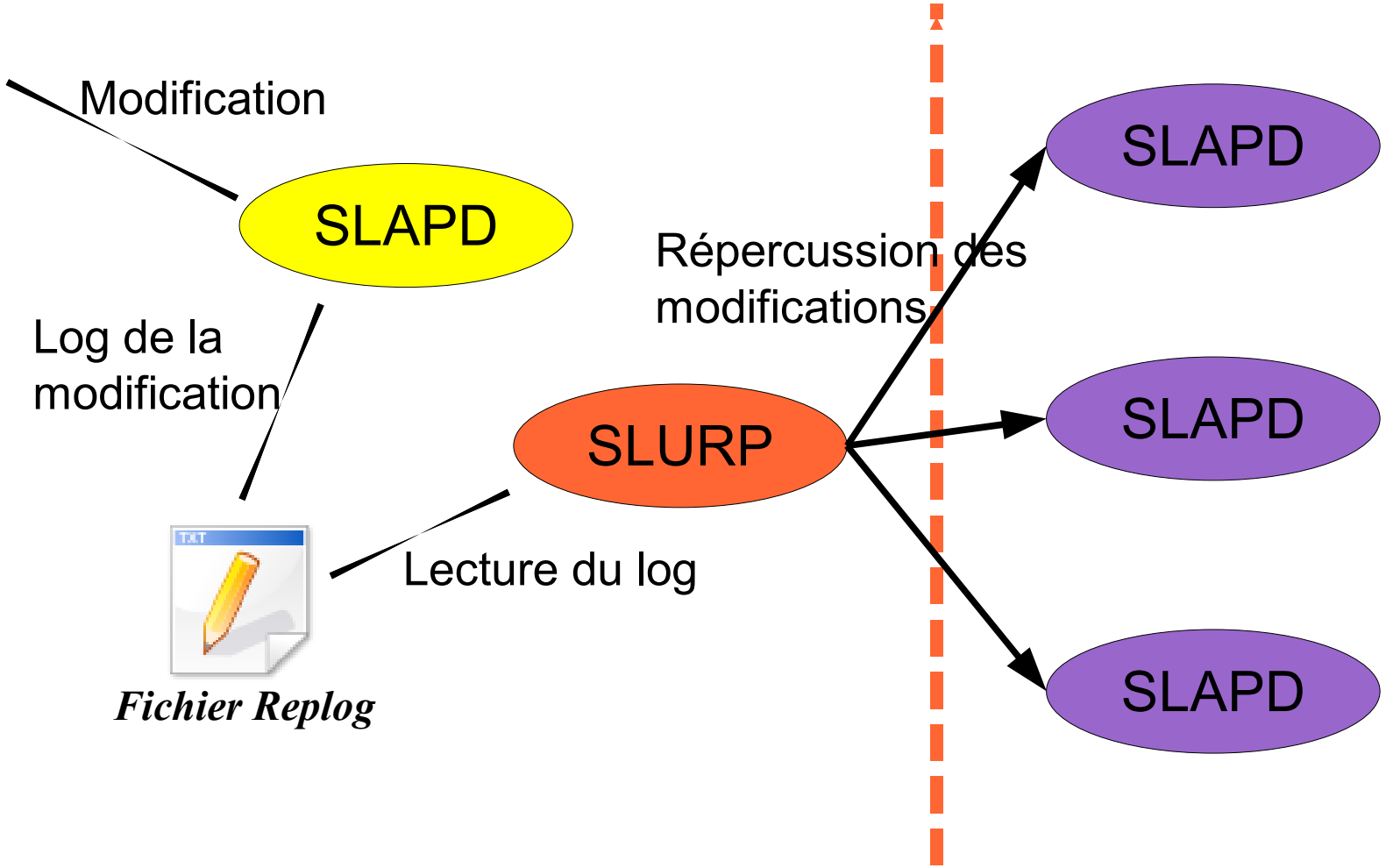
# Plan

- Principes de base
- L'implémentation OpenLDAP
- Interfaçage applicatif
- Conclusion

# Openldap

- Implémentation libre de serveur LDAP
- Une instance peut héberger plusieurs arborescences
  - Une base = un **backend**
- Plusieurs format de stockage (type de backend) possible
  - Bdb, ldbm, hdb, sql
  - Fausses bases : perl, passwd, monitor
- Possibilité de faire de la réplication
  - Slurpd
  - Syncrepl

# Réplication avec Slurpd



# LDAP Sync replication engine

- Principe inverse de slurpd
  - L'esclave demande les mises à jour
  - Configuration sur l'esclave
- Deux modes de fonctionnement :
  - RefreshOnly
    - plusieurs requêtes à un intervalle de temps configuré
  - RefreshAndPersist
    - une seule requête « persistante »
    - Le maître renvoie les modifications au fur et à mesure

# Access list

- Filtrage souple et puissant
- Parcours séquenciel des ACL

**Access to** *<objet>* **by** *<sujet>* *<action>* *<contrôle>*

<objet> : dn + filtre + liste d'attribut
<sujet> : dn, adresse ip, nom de domaine, users, *,...
<action> : none | auth | compare | search | read | write
<contrôle> : stop | break | continue

- Attention à l'ordre

# Performances

- Minimiser l'impact du volume de données sur les performances
- Indexation dans la configuration de slapd
  - Indexation des attributs sur type de comparaison
    - pres,eq,approx,sub,none
- Configuration du backend
  - Directives d'optimisation dans la conf de slapd
  - Fichier de configuration du backend (DB_CONFIG)
  - Taille des caches, checkpoint, syncho immédiate...

# Outils clients

- Ldapsearch : recherche
- Ldapmodify : modification
- Ldapadd : ajout
- Ldapdelete : suppression
- Slapcat : dump direct de la base
- Slapadd : ajout direct dans la base
- Slappasswd : chiffrement mot de passe

OSIRIS

# Plan

- Principes de base
- L'implémentation OpenLDAP
- Interfaçage applicatif
- Conclusion

# Interfaçage applicatif

- Annuaire : stockage d'information pour les applications
- Accessible par un protocole standard : LDAP
- Beaucoup de possibilités :
  - Annuaire bien sûr
  - Authentification
  - Gestion de droits
  - Et beaucoup d'autres
- Utilisation de schémas standards ou spécifiques

# Configuration des applications

- Principe globalement identique pour toutes
- Informations d'accès à l'annuaire
  - Adresse IP et port du service LDAP
    - Ldap : 389 , ldaps : 636
  - BaseDN : suffixe de l'annuaire
  - BindDn : login d'accès
  - Password : mot de passe du bindDn
- Correspondance des attributs
  - Quel est l'attribut pour quel information ?
  - Ex : nom de l'attribut contenant le login, le nom

# Exemples d'applications

- Serveur web Apache (2.2)
  - mod_authnz_ldap
  - **`AuthLDAPUrl ldap://host:port/basedn?attribute?scope?filter`**
  - **`Require (valid-user | ldap-user | ldap-dn | ...)`**
  - http://httpd.apache.org/
- Serveur FTP (ex: PureFTP)
  - Authentification

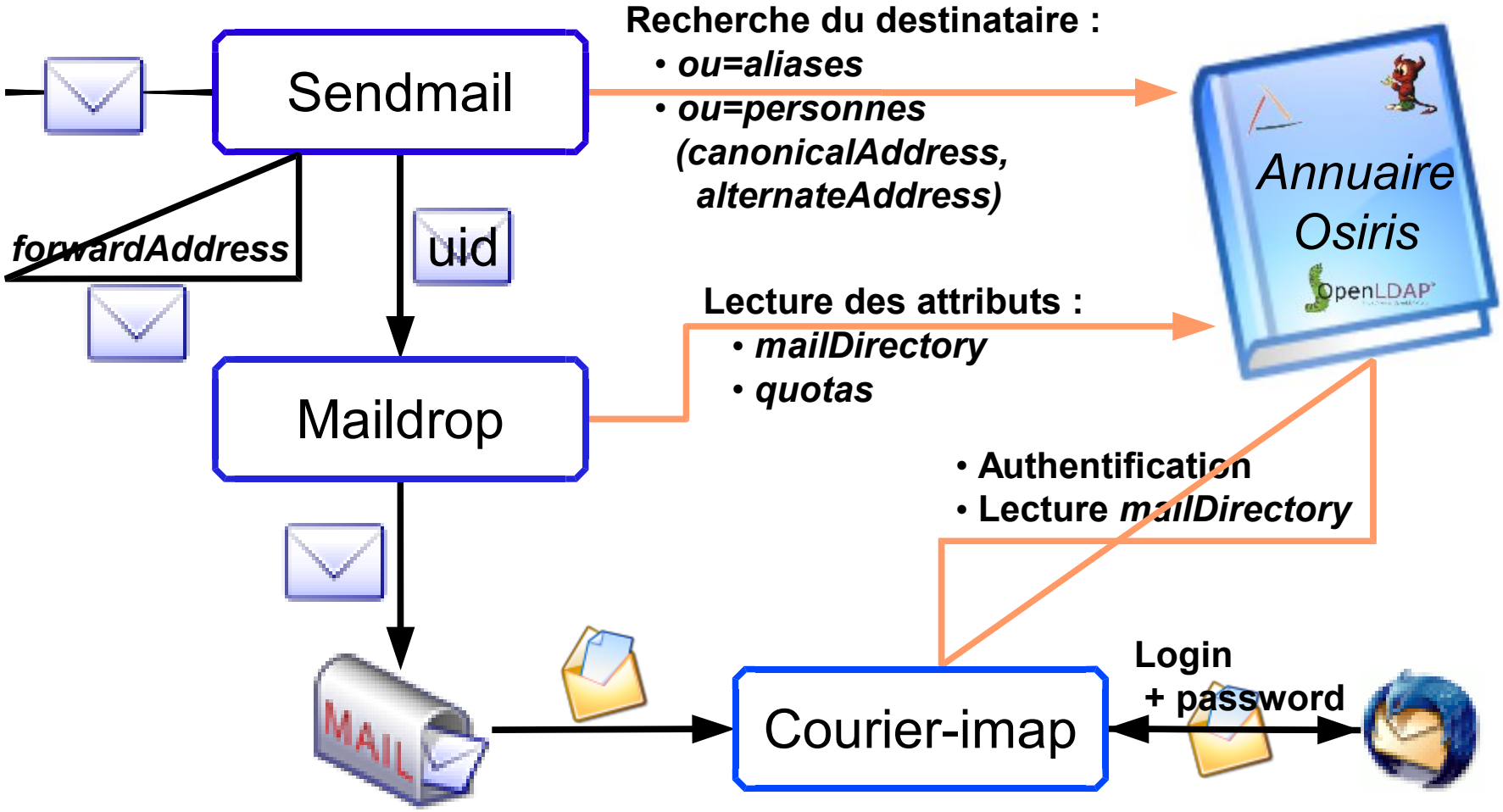
# Exemples d'applications

- Librairie pam_ldap
  - Toutes les applications pouvant s'appuyer sur pam_ldap
    - Login, ssh
- Freeradius : authentification et autorisation
- Name service switch (nsswitch.conf)
  - Identique à NIS pour des machines clientes UNIX
  - Lecture user, group, hosts, services,...

# Exemples d'applications

- Samba
  - Authentification
  - Données utilisateurs : groupes, chemin du profil,...
- Liste de diffusions : sympa
  - Authentification
  - Données recherchées grâce à des filtres
    - Abonnés, modérateurs
- Clients de messagerie
  - Carnet d'adresses LDAP (thunderbird)

# Applications de messagerie



# Librairies de développement

- Librairies disponibles dans de nombreux langages
  - Perl : Net::LDAP (perl-ldap-033)
  - C/C++ : libldap
    - librairie Openldap
  - Java : LDAP Class Librairies (JLDAP)
    - http://www.openldap.org/jldap/
  - PHP
    - Nécessite l'installation de la librairie Openldap
    - Option compilation de PHP : --with-ldap
  - TCL : module LDAP
    - inclus dans la tcllib

# Plan

- Principes de base
- L'implémentation OpenLDAP
- Interfaçage applicatif
- Conclusion

# Avantages

- Souplesse et simplicité
  - Peu de contraintes
  - Structure de données simple
    - Ex: Attributs multivalués
- Distributivité
  - références sur d'autres annuaire
- Réplication
  - Facile et fiable
  - Répartition des charges
- Universel
  - Beaucoup d'applications parlent LDAP

# Inconvénients

- Opérations de bas niveau
  - Pas de contraintes d'intégrité
    - Beaucoup de tests à faire au niveau applicatif
  - Pas de transactions
- Requêtes complexes irréalisables
  - Pas de jointures !!!
- Des doutes sur l'implémentation openLDAP ...
  - Modifications de schéma = recharger la base
  - Documentation faible

# Docs et références

- http://www.openldap.org
- http://ldapbook.labs.libre-entreprise.org
  - Index des RFC
- http://www.zytrax.com/books/ldap/
- ftp://kalamazoolinux.org/pub/pdf/ldapv3-00-28Oct2002-1-printer_friendly.pdf
- http://ldap.akbkhome.com
  - Browser des schémas standard