LDAP : concepts, déploiement

Laurent Mirtain
C. Claveleira, C. Gross

Journées LDAP - 27 et 28 septembre 2000

Plan

- Concepts
- Déployer un service LDAP
- Les logiciels serveurs
- Les clients LDAP
- Les outils de développement
- Les applications de LDAP aujourd'hui et demain
- Bibliographie

Concepts

- Concepts
  - Qu'est-ce qu'un annuaire ?
  - Historique
  - LDAP
- Déployer un service LDAP
- Les logiciels serveurs
- Les clients LDAP
- Les outils de développement
- Les applications de LDAP aujourd'hui et demain
- Bibliographie

Concepts : qu'est-ce qu'un annuaire ?

- Un conteneur d'informations organisées
- Exemples d'annuaires courants
  - annuaire téléphonique
  - carnet d'adresses
  - catalogue de vente
  - guides télé

Ce sont des annuaires offline

Concepts : qu'est-ce qu'un annuaire ?

- Un service d'annuaire électronique, c'est en plus...
  - un protocole qui permet l'accès au contenu
  - une syntaxe qui permet d'interroger la base
- et aussi
  - un modèle de duplication
  - un modèle de distribution des données

Concepts : qu'est-ce qu'un annuaire ?

Spécificités des annuaires électroniques

- dynamiques (informations changent -> + à jour)
- souples (changement aisé type et organisation des données)
- peuvent être sécurisés (qui voit quoi)
- peuvent être personalisés (façon de présenter les données, action sur ses propres données,...)

Concepts : qu'est-ce qu'un annuaire ?

Caractéristiques comparées des annuaires et base de données

- rapport lecture/écriture (beaucoup) plus élevé pour les annuaires
- annuaires plus facilement extensibles (types de données)
- diffusion à beaucoup plus large échelle
- distribution des données entre serveurs plus facile avec les annuaires
- plus grande duplication des informations des annuaires (+ fiable, +performant, + proche des clients)
- importance des standards -> LDAP
- performances globales des annuaires plus élevées (en lecture)

Concepts : qu'est-ce qu'un annuaire ?

exemple de services d'annuaires que nous utilisons déjà : le DNS

- pour obtenir l'url http://www.sncf.com/ il faut obtenir l'adresse du serveur www.sncf.com -> requête DNS

❑ DNS est un exemple d'un service d'annuaire global

- il est distribué entre des serveurs coopérants
- il a un espace de nommage uniforme

Concepts : à quoi peut servir un annuaire en ligne ?

- chercher (et trouver) des informations mieux et plus vite
- pour des humains ou des applications
- gérer (carnets d'adresses, comptes utilisateurs, profils,...)
- de base de donnée simple
- à stocker et diffuser des certificats dans une PKI

Concepts : ce que n'est pas un annuaire

- approprié à de fréquentes écritures
- destiné à manipuler des données volumineuses
- un substitut à un serveur FTP, un système de fichiers,...

Concepts : historique

Historiquement sont apparus :

- Bases de comptes de systèmes multi-utilisateurs (70-80)
  - Unix `/etc/passwd`,
  - IBM MVS PROFS
  - ...
- Grapevine (Xerox, début 80)
- Internet Domain Name System (84)
  - service de nommage réseau
  - spécifique mais efficace
- WHOIS
  - bases de contacts

Concepts : historique

- Les annuaires dédiés aux applications
  - Lotus cc:Mail, Notes
  - Unix sendmail /etc/aliases
  - Microsoft Exchange
- Les annuaires Internet (offrent de plus en plus un accès LDAP)
  - Bigfoot, Yahoo's Four11, AnyWho (AT&T), Schwitboard
- Les annuaires système-réseau (NOS)
  - Sun NIS, NIS+
  - Novell NetWare Directory Service (93) (proche d'X500)
  - Microsoft Active Directory (natif LDAP)
- Les annuaires multi-usage
  - X.500 (88-93-97)
  - WHOIS++ (93)
  - CSO (PH)
  - LDAP (93)

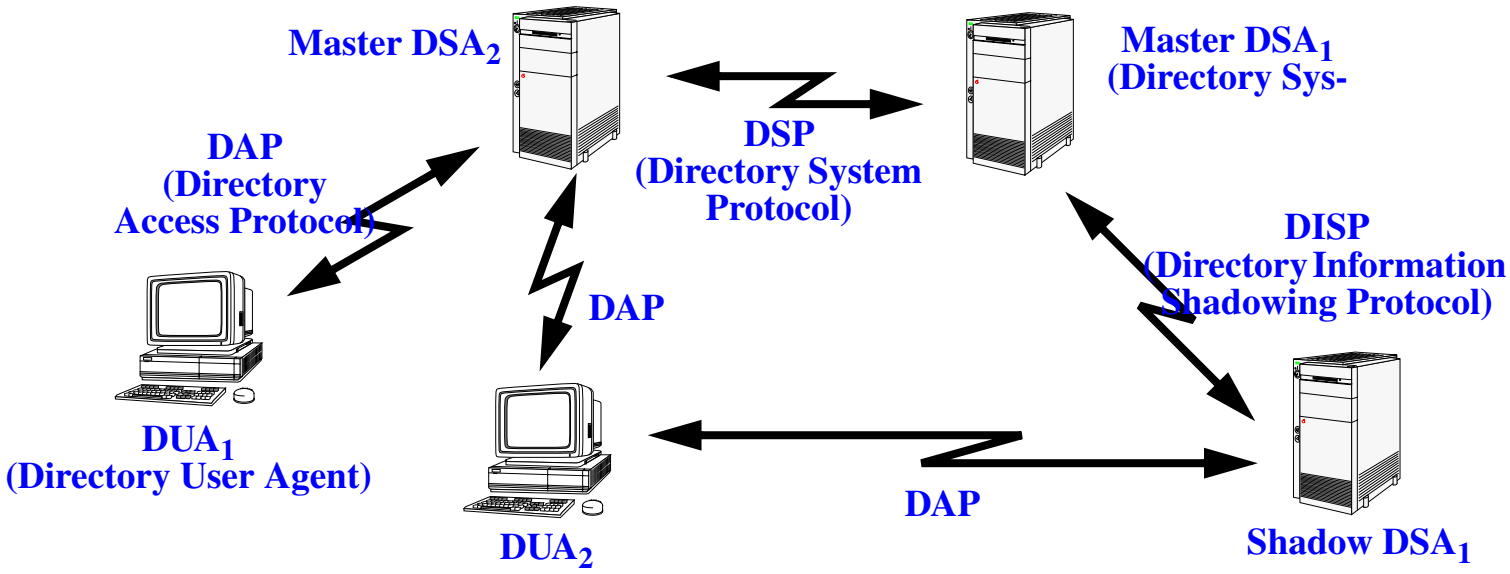
Concepts : historique : X.500

- Standard conçu par les opérateurs télécom pour interconnecter leurs annuaires téléphoniques.
- Destiné à devenir LE service d'annuaire GLOBAL distribué, normalisé et fédérateur.
- Mais conçu aussi pour répondre à tout type de besoin d'annuaire grâce à un modèle de données de type objet et extensible.

Concepts : historique : X.500

❑ X.500 définit :

- les règles pour nommer les objets et les entités
- les protocoles pour fournir le service d'annuaire
- un mécanisme d'authentification.



X500 = suite (conséquente) : X501, X509, X511, X518, X519, X520, X521, X525

Concepts : historique : X.500

- Exemple d'annuaire X.500 :
  - NameFlow Paradise (Piloting An inteRnationAl DIrectory SErvice),
  - SURFNET (nl)...
- Logiciels DSA X.500
  - ISODE Consortium/Quipu,
  - NeXor/XT-Quipu,
  - Control Data/Rialto Global Directory Server

Concepts : historique : X.500

Qualités et défauts d'X500 :

- Atouts d'X500 :
  - scalability, fonctions de recherche évoluées, distribué (données et administration), ouvert
- Défauts d'X500 :
  - implémentations (très) lourdes, buggées et difficilement interopérables, basé sur les protocoles ISO, contraire à la culture internet
- Echec : les ambitions d'X500 n'ont pas été atteintes

Concepts : historique : LDAP

- En 1993 Lightweight Directory Access Protocol (LDAP) est né de l'adaptation et du dégraissage de X.500 DAP au protocole TCP/IP.
- Deux groupes de travail aboutissent à 2 produits fonctionnant comme frontal X.500 :
  - Directory Assistance Service (DAS) : RFC 1202
  - Directory Interface to X.500 Implemented Efficiently (DIXIE) : RFC 1249

qui convergent finalement vers le standard IETF LDAP.

- LDAPv1 : RFC 1487
- LDAPv2 : RFC 1777
- LDAPv3 : RFC 2251

LDAP garde beaucoup d'aspects de X.500 dans les grandes lignes, mais va dans le sens de la simplification et de la performance

Concepts : historique : LDAP

- LDAP est initialement un frontal d'accès à des bases d'annuaires X.500 (translateur LDAP/DAP).
- Devient un annuaire natif (standalone LDAP) utilisant sa propre base de données, sous l'impulsion d'une équipe de l'Université du Michigan (U-M LDAP 3.2 en 95).

(Wengyik Yeong, Steve Kille, Colin Robbins, Tim Howes, Marc Wahl).

- En 96, apparaissent les premier serveurs commerciaux.

Concepts : historique, autres annuaires

- WHOIS++ (93)

Whois utilisait une seule base, Whois++ introduit la notion de bases réparties reliées par le Whois++ index service.

- CSO

Annuaire d'adresses électroniques créé par l'université de l'Illinois, plus connu sous le nom de PH.

Popularisé par Eudora.

- NETFIND, SOLO (Simple Object LOok)...

Des clients capables d'interroger différents types de Directory Servers (notion de Meta-Directory Service).

Concepts : LDAP

- Concepts
  - Qu'est-ce qu'un annuaire ?
  - Historique
  - LDAP
    - Protocole
    - Modèle d'information
    - Modèle de nommage
    - Modèle fonctionnel
    - Modèle de sécurité
    - Modèle de duplication
    - APIs
    - LDIF

Concepts : LDAP

LDAP définit :

- le protocole d'accès -- comment accéder à l'information contenue dans l'annuaire,
- un modèle d'information -- le type d'informations contenues dans l'annuaire,
- un modèle de nommage -- comment l'information est organisée et référencée,
- un modèle fonctionnel -- comment on accède et met à jour l'information,
- un modèle de sécurité -- comment données et accès sont protégés,
- un modèle de duplication -- comment la base est répartie entre serveurs,
- des API -- pour développer des applications clientes,
- LDIF -- un format d'échange de données.

Concepts : LDAP, le protocole

Le protocole définit :

- Comment s'établit la communication client-serveur :
  - commandes pour se connecter ou se déconnecter, pour rechercher, comparer, créer, modifier ou effacer des entrées.
- Comment s'établit la communication serveur-serveur :
  - échanger leur contenu et le synchroniser (replication service)
  - créer des liens permettant de relier des annuaires les uns aux autres (referral service).
- Le format de transport de données :
  - pas l'ASCII (comme pour http, smtp...) mais le Basic Encoding Rules (BER), sous une forme allégée (appelée LBER : Lightweight BER)

Concepts : LDAP, le protocole

Le protocole définit (suite) :

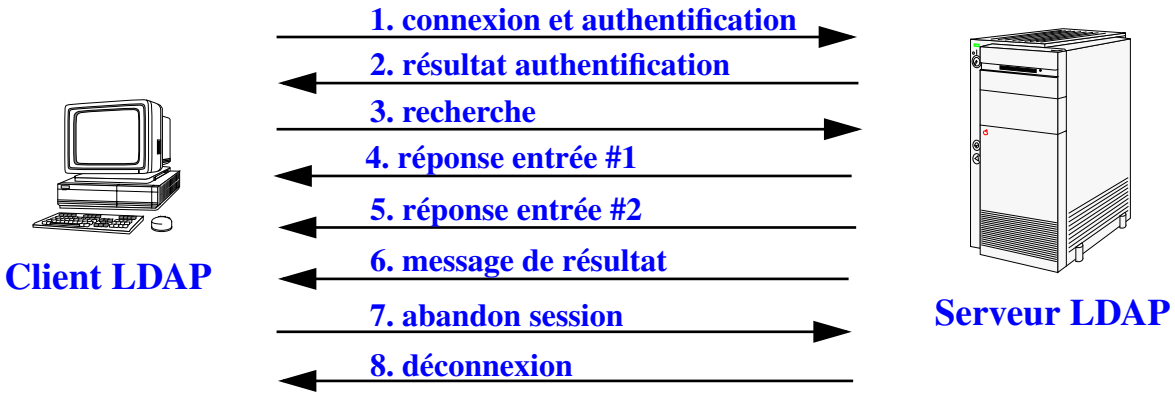
- Les mécanismes de sécurité :
  - méthodes de chiffrement et d'authentification
  - mécanismes de règles d'accès aux données.
- Les opérations de base:
  - interrogation : `search, compare`
  - mise à jour : `add, delete, modify, rename`
  - connexion au service : `bind, unbind, abandon`

Concepts : LDAP, le protocole

- Communication client-serveur :
  - normalisée par l'IETF : la version actuelle est LDAPv3 (RFC2251).
- Communication serveur-serveur :
  - le referral service est définit par LDAPv3,
  - le replication service est encore en cours de normalisation sous la dénomination LDAP Duplication Protocol (LDUP)

Concepts : LDAP, le protocole

- possibilité d'avoir une seule connexion pour passer plusieurs requêtes



Concepts : LDAP, le protocole

- LDAPv3 est conçu pour être extensible sans avoir à modifier la norme grâce à 3 concepts :
  - LDAP extented operations : rajouter une opération, en plus des neuf opérations de base.
  - LDAP controls : paramètres supplémentaires associés à une opération qui en modifient le comportement.
  - Simple Authentification and Security Layer : couche supplémentaire permettant à LDAP d'utiliser des méthodes d'authentification externes.

Concepts : LDAP, modèle d'information

- Le modèle d'information définit le type de données pouvant être stockées dans l'annuaire.
  - L'entrée (Entry) = élement de base de l'annuaire. Elle contient les informations sur un objet de l'annuaire.
  - Ces informations sont représentées sous la forme d'attributs décrivant les caractéristiques de l'objet.
  - Toute sorte de classe d'objet (réel ou abstrait) peut être représentée.
  - Le schéma de l'annuaire définit la liste des classes d'objets qu'il connaît.

Concepts : LDAP, modèle d'information

Schéma

- Le Directory schema est l'ensemble des définitions relatives aux objets qu'il sait gérer (~typedef).
- Le schéma décrit les classes d'objets, les types des attributs et leur syntaxe.
- Chaque entrée de l'annuaire fait obligatoirement référence à une classe d'objet du schéma et ne doit contenir que des attributs qui sont rattachés au type d'objet en question.

Concepts : LDAP, modèle d'information

❑ Attributs

Un type d'attribut (ou attribut) est caractérisé par :

- Un nom, qui l'identifie
- Un Object Identifier (OID), qui l'identifie également
- S'il est mono ou multi-valué
- Une syntaxe et des règles de comparaison (matching rules)
- Un format ou une limite de taille de valeur qui lui est associée

Tableau 1 : Exemple d'attributs d'une entrée

type d'attribut	valeur d'attribut
cn:	Barnabé Dupond
uid:	bdupond
telephonenumber:	+33 (0)1 2345 6789
mail:	Barnabe.Dupond@acme.com
roomnumber:	C105

Concepts : LDAP, modèle d'information

Les types d'attributs ont une syntaxe qui sert à décrire le format de données et comment l'annuaire compare ces valeurs lors d'une recherche sur critère.

Tableau 2 : Exemple de syntaxes d'attributs

syntaxe LDAP	syntaxe X.500	description
cis	caseIgnoreMatch	texte, la casse n'est pas prise en compte
ces	caseExactMach	texte, la casse intervient
tel	telephoneNumberMatch	texte représentant un numéro de tel
int	integerMatch	nombre entier, comparaison numérique
dn	distinguishedNameMatch	nom d'entrée, règles spécifiques
bin	octetStringMatch	données binaires, comparaison byte/byte

Concepts : LDAP, modèle d'information

- 2 catégories d'attributs :
  - ***User attributes*** : attributs « normaux » manipulés par les utilisateurs (`givenname`, `telephoneNumber`),
  - ***Operational*** attributes : atttributs « systèmes » utilisé par le serveur (`modifiersname`)
- Certains serveurs LDAP respectent les standards X.500 de hiérarchisation des attributs :
  - permettent de décrire un attribut comme étant un sous-type d'un attribut super-type et d'hériter ainsi de ses caractéristiques.

Exemple : cn, sn, givenname sont des sous-types de l'attribut super-type name

Concepts : LDAP, modèle d'information

❑ Classes d'objets

Les classes d'objets modélisent des objets réels ou abstraits en les caractérisant par une liste d'attributs optionnels ou obligatoires. Une classe d'objet est définie par :

- Un nom, qui l'identifie
- Un OID, qui l'identifie également
- Des attributs obligatoires
- Des attributs optionnels
- Un type (structurel, auxiliaire ou abstrait)

Exemples de classes d'objet :

- une organisation (`o`),
- ses départements (`ou`),
- son personnel (`organizationalPerson`),
- ses imprimantes (`device`),
- ses groupes de travail (`groupofnames`).

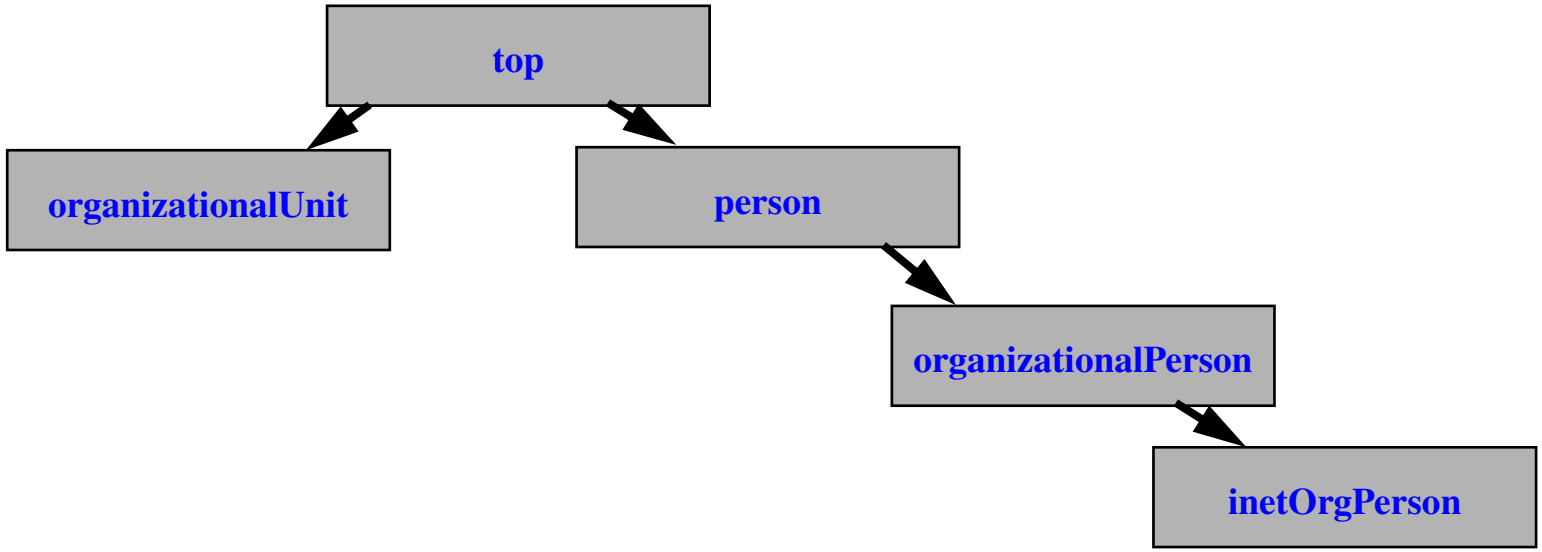
Concepts : LDAP, modèle d'information

Le type d'une classe est lié à la nature des attributs qu'elle utilise :

- Une classe structurelle correspond à la description d'objets courants de l'annuaire : les personnes, les groupes, les unités organisationnelles... Une entrée appartient toujours à une classe d'objet structurelle.
- Une classe auxiliaire permet de rajouter des informations complémentaires à des objets structurels.
- Une classe abstraite désigne des objets basiques du schéma (top, alias).

Concepts : LDAP, modèle d'information

Les classes d'objets forment une hiérarchie, au sommet de laquelle se trouve l'objet `top`.



- Chaque objet hérite des propriétés (attributs) de l'objet dont il est le fils.
- On précise la classe d'objet d'une entrée à l'aide de l'attribut `objectClass`.
- Il faut obligatoirement indiquer la parenté de la classe d'objet en partant de l'objet `top` et en passant par chaque ancêtre de l'objet.

Concepts : LDAP, modèle d'information

Par exemple, l'objet `inetOrgPerson` à la filiation suivante :

```
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
```

L'objet person **a comme attributs :** commonName, surname, description, seeAlso, telephoneNumber, userPassword

L'objet fils organizationalPerson **ajoute des attributs comme :** organizationUnitName, title, postalAddress**...**

L'objet petit-fils inetOrgPerson **lui rajoute des attributs comme :** mail, labeledURI, uid (userID), photo**...**

Une entrée peut appartenir à un nombre non limité de classes d'objets.

Les attributs obligatoires sont la réunion des attributs obligatoires de chaque classe.

Concepts : LDAP, modèle d'information

Exemple de classes d'objets :

Entry Type	Required Attributes	Optional Attributes
inetOrgPerson (defines entries for a person)	•commonName (cn) •surname (sn) •objectClass	•businessCategory •carLicense •departmentNumber • description •employeeNumber •facsimileTelephone •Number •givenName •mail •manager •mobile •organizationalUnit (ou) •pager •postalAddress •roomNumber •secretary •seeAlso •telephoneNumber •title •labeledURI •uid
organizationalUnit (defines entries for organizational units)	•ou •objectClass	•businessCategory •description •facsimileTelephoneNumber •location (l) •postalAddress •seeAlso •telephoneNumber

Concepts : LDAP, modèle d'information

❑ OIDs

Les classes d'objets et les attributs

- sont normalisés par le RFC2256 afin de garantir l'interopérabilité entre logiciels.

- Sont référencées par un object identifier (OID) unique dont la liste est tenue à jour par l'Internet Assigned Numbers Authority (IANA).

Un OID est une séquence de nombres entiers séparés par des points. Les OIDs sont alloués de manière hiérarchique :

- seule, l'autorité qui a délégation sur la hiérarchie « 1.2.3 » peut définir la signification de l'objet « 1.2.3.4 ». Par exemple :

```
2.5                - fait référence au service X.500
2.5.4              - est la définition des types d'attributs
2.5.6              - est la définition des classes d'objets
1.3.6.1            - Internet OID
1.3.6.1.4.1        - IANA-assigned company OIDs, utilisé pour entreprises privées
1.3.6.1.4.1.4203   - OpenLDAP
1.3.6.1.4.1.7135   - pour le CRU
```

Concepts : LDAP, modèle d'information

❑ Définition des schémas

Les schémas existants sont issus de X.500, plus des ajouts de LDAP ou d'autres consortium industriels.

Il existe plusieurs formats pour décrire un schéma LDAP :

- `slapd.conf` : fichier de configuration utilisé par U-M slapd, OpenLDAP et Netscape Directory.
- `ASN.1` : grammaire utilisée dans les documents décrivant les standards LDAP et X.500.
- `LDAPv3` : LDAPv3 introduit l'obligation pour un serveur de publier son schéma via LDAP en le stockant dans l'entrée `subschema`.

Concepts : LDAP, modèle d'information

- Exemple de syntaxe slapd.conf :

```
attribute NAME [ALIASES] [OID] SYNTAXID [OPTIONS]

attribute cn commonName 2.5.4.3 cis

objectclass NAME [oid OID] [superior SUP] [requires REQATTRS] [allows ALLOWATTTRS]

objectclass person
   oid 2.5.6.6
   superior top
   requires
      sn,
      cn
   allows
      description,
      seeAlso,
      telephoneNumber,
      userPassword
```

Concepts : LDAP, modèle d'information

- Exemple de syntaxe ASN.1 :

```
ub-common-name INTEGER ::= 64
commonName ATTRIBUTE
   WITH ATTRIBUTE-SYNTAX caseIgnoreStringSyntax
   (SIZE (1..ub-common-name))
   ::= {attributeType 3}

person OBJECT-CLASS ::= {
   SUBCLASS OF top
   MUST CONTAIN {
      commonName,
      surname}
   MAY CONTAIN {
      description,
      seeAlso,
      telephoneNumber,
      userPassword}
   ::= {objectClass 6}
```

Concepts : LDAP, modèle d'information

- Exemple de syntaxe LDAPv3 (attribut `cn` et objet `person`)

```
attributetypes: (2.5.4.3 NAME 'cn' DESC 'commonName Standard'
Attribute' SYNTAX 1.3.5.1.4.1.1466.115.121.1.15)

objectclass: (2.5.6.6 NAME 'person' DESC 'standard person'
   Object Class' SUP 'top'
   MUST (objectclass $ sn $ cn )
   MAY ( description $ seealso $ telephonenumber $ userpassword )
)
```

Concepts : LDAP, modèle d'information

- Schema checking

Quand une entrée est créée, le serveur vérifie si sa syntaxe est conforme à sa classe ou ses classes d'appartenance : c'est le processus de Schema Checking.

Concepts : LDAP, modèle de nommage

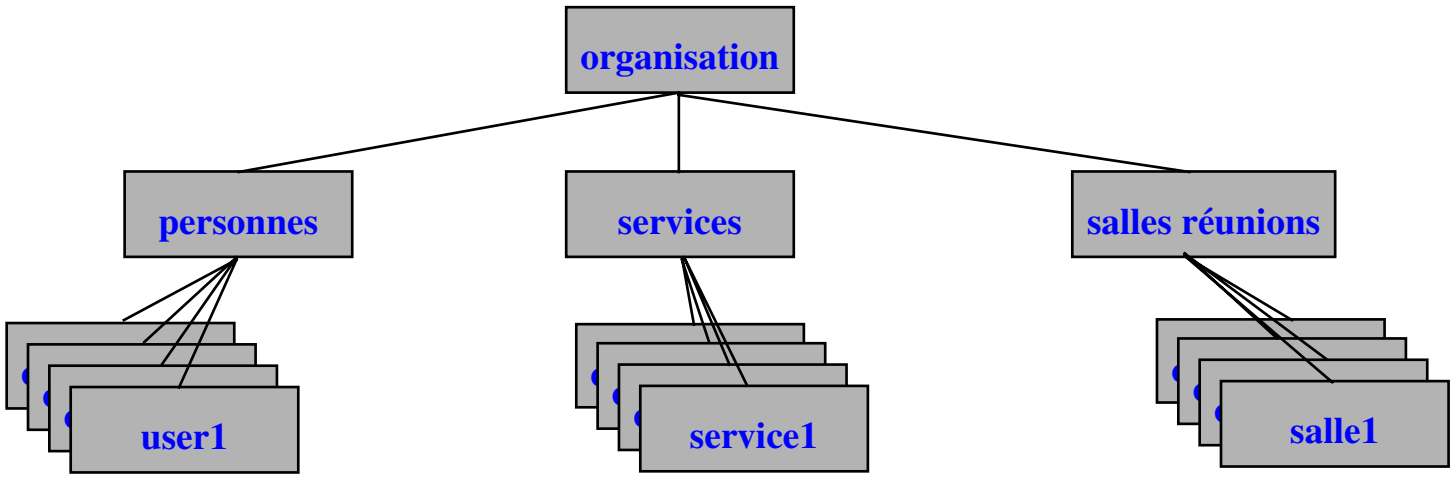
- Le modèle de nommage définit comment sont organisées les entrées de l'annuaire et comment elles sont référencées.
- Les entrées représentent des objets.
- L'organisation de ces objets se fait suivant une structure logique hiérarchique : le Directory Information Tree (DIT).
- Au sein de ce DIT, l'identification d'une entrée se fait à l'aide d'un nom, le Distinguish Name (DN).

Concepts : LDAP, modèle de nommage

- Le Directory Information Tree (DIT)

Classification des entrées dans une arborescence hiérarchique (comparable au système de fichier Unix).

Exemple de modélisation d'une organisation



Chaque nœud de l'arbre correspond à une entrée de l'annuaire ou directory specific entry (DSE).

Au sommet de l'arbre se trouve l'entrée Suffix ou Root Entry ou BaseDN, qui caractérise une base LDAP.

Concepts : LDAP, modèle de nommage

Le suffix définit l'espace de nommage dont le serveur a la gestion.

Un serveur peut gérer plusieurs arbres (donc plusieurs suffixes).

Il possède une entrée spéciale, appelée root DSA Specific Entry (rootDSE) qui contient la description du DIT (V3).

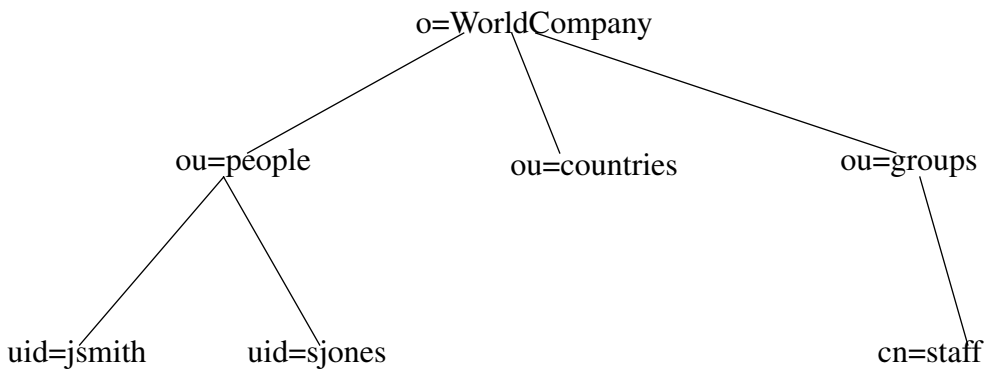
- Avec LDAP, vous êtes libres d'organiser vos données comme bon vous semble (design du DIT) (≠X500).
- Des contraintes (performance, gestion...) impliqueront de choisir tel ou tel type de modèle (cf § déploiement).

Concepts : LDAP, modèle de nommage

❑ Le Distinguished name (DN)

Référence de manière unique une entrée du DIT (⇔ path fichier UNIX).

Formé de la suite des noms des entrées, en partant de l'entrée et en remontant vers le suffix, séparé par des ",".



Ex : le DN de l'entrée **`jsmith`** vaut :

`uid=jsmith, ou=people, o=WorldCompany`

Chaque composant du DN est appelé Relative Distinguished Name (RDN).

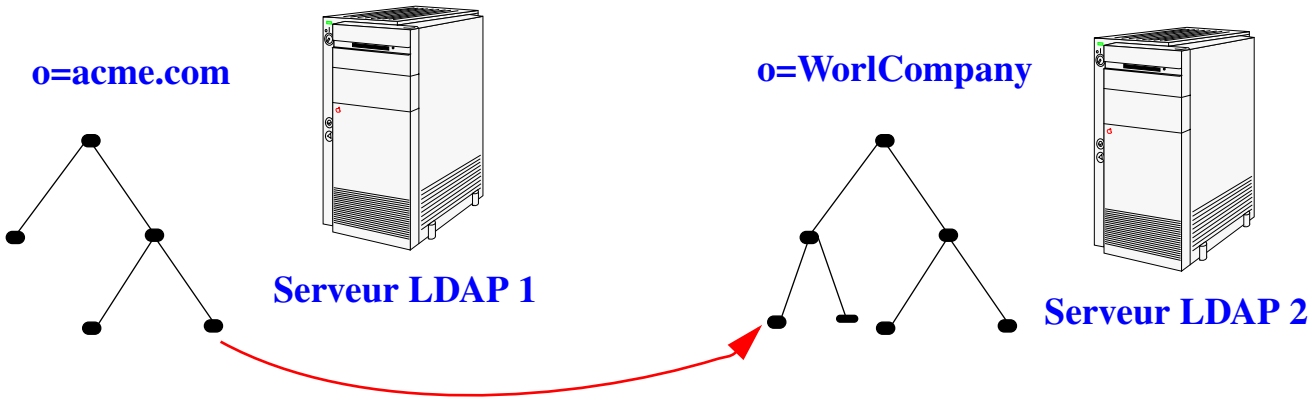
Le RDN est constitué d'un des attributs de l'entrée (et de sa valeur). Le choix de cet attribut doit assurer que 2 entrées du DIT n'aient pas le même DN.

Concepts : LDAP, modèle de nommage

❑ Alias et referral

Deux objets abstraits particuliers : les aliases et les referrals

→ permettent à une entrée de l'annuaire de pointer vers une autre entrée du même ou d'un autre annuaire.



→ L'attribut `aliasObjectName` de l'objet `alias` a pour valeur le DN de l'entrée pointée.

→ L'attribut `ref` de l'objet `referral` a pour valeur l'URL LDAP de l'entrée désignée.

→ Les referrals sont traités au niveau du serveur en LDAP V2, par le client en V3

Concepts : LDAP, modèle fonctionnel

- Le modèle fonctionnel décrit le moyen d'accéder aux données et les opérations qu'on peut leur appliquer.

Le modèle définit :

- Les opérations d'interrogation.
- Les opérations de comparaison.
- Les opérations de mise à jour.
- Les opérations d'authentification et de contrôle.
- Les opérations étendus (V3)

Concepts : LDAP, modèle fonctionnel

❑ Interrogation

LDAP ne fournit pas d'opération de lecture d'entrée.

Pour connaître le contenu d'une entrée, il faut écrire une requête qui pointe sur cette entrée.

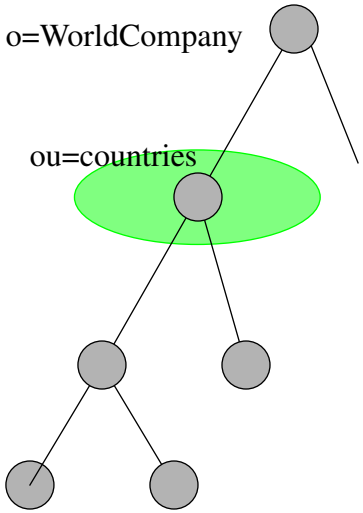
Une requête est composée de 8 paramètres :

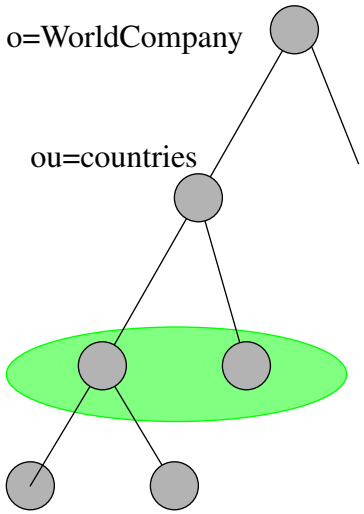
Tableau 3 :

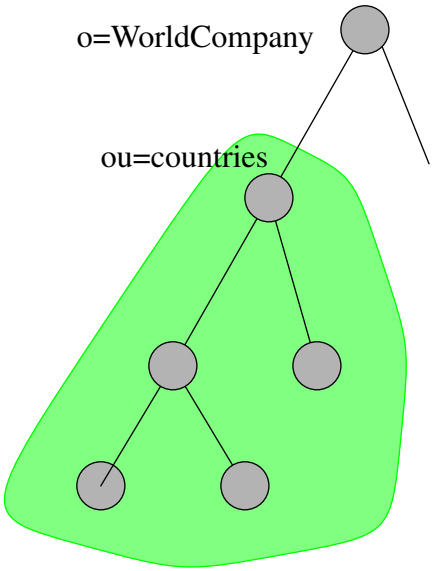
`base object`	l'endroit de l'arbre où doit commencer la recherche
`scope`	la profondeur de la recherche
`derefAliases`	si on suit les liens ou pas
`size limit`	nombre de réponses limite
`time limit`	temps maxi alloué pour la recherche
`attrOnly`	renvoie ou pas la valeur des attributs en plus de leur type
`search filter`	le filtre de recherche
`list of attributes`	la liste des attributs que l'on souhaite connaître

Concepts : LDAP, modèle fonctionnel

❑ Le scope







search base = "ou=countries,o=WorldCompany"

search scope = base	search scope = onelevel	search scope = subtree

Concepts : LDAP, modèle fonctionnel

❑ Les filtres de recherche (RFC 2254)

(<operator>(<search operation>)(<search operation>)...))

Tableau 4 : Exemples de filtres de recherche

(cn=Norbert Durand)	égalité	Nom vaut "Norbert Durand"
(cn=*Mart*)	sous-chaîne	Nom contient "Mart"
(cn~=martin)	approximation	Nom sonne comme "martin"
(employeenumber>=100)	comparaison	Numéro supérieur à 100
(sn=*)	existance	Tous les noms propres
(&(sn=Durand)(l=paris))	ET	Nom vaut "Durand" ET localisation vaut paris
(|(ou=gens)(ou=groupes))	OU	ou vaut gens ou groupes
(!(tel=*))	NON	Toutes les entrées sans attribut téléphone

Ex :

(&(objectclass=inetOrgPerson)(!(mail=*)))Toutes les entrées de type utilisateur sans adresse mail

Concepts : LDAP, modèle fonctionnel : comparaison

- Héritage de X.500 : vérifier si l'attribut d'une entrée contient bien une valeur spécifiée. Le serveur répond vrai ou faux.
- Equivalent à une recherche qui renvoie l'entrée si vrai et ne renvoie rien dans deux cas :
  - si l'attribut ne contient pas cette valeur,
  - si l'attribut n'existe pas

alors que la comparaison renvoie dans ce 2ème cas, un code d'erreur.

Concepts : LDAP, modèle fonctionnel : mise à jour

- 4 opérations : `add, delete, rename, modify`

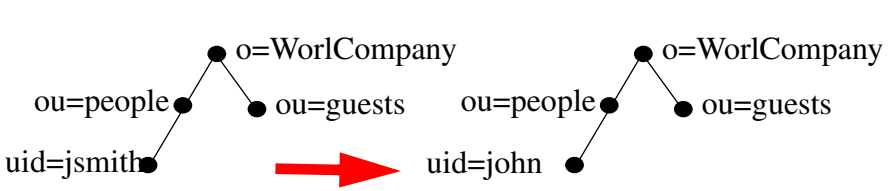
Ces quatre opérations nécessitent les droits d'accès appropriés et des prérequis :

- `add, rename` : entrée ne doit pas déjà exister, entrée doit avoir un parent existant
- `add, modify` : les attributs doivent être conformes au schéma
- `delete` : entrée ne doit pas avoir d'enfant

Concepts : LDAP, modèle fonctionnel : mise à jour

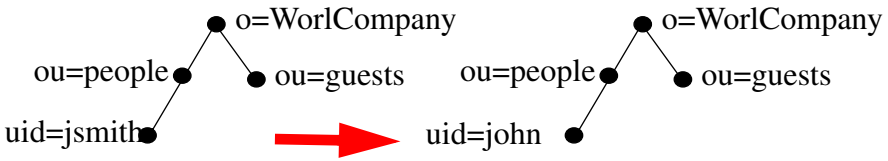
❑ `rename` = `modifyRDN` plus `modifyDN(v3)`

- changer le RDN sans bouger de place
- changer le RDN sans bouger de place, en gardant l'ancien RDN en attribut
- déplacer l'entrée dans l'arbre en gardant le même RDN
- déplacer l'entrée dans l'arbre en changeant le RDN



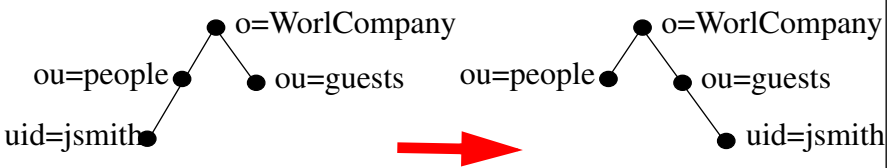
dn: uid=jsmith,ou=people,o=WorlCompany
uid=jsmith

dn: uid=john,ou=people,o=WorlCompany
uid=john



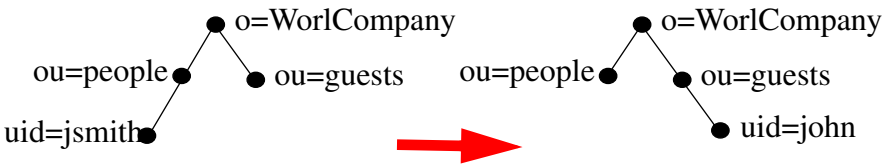
dn: uid=jsmith,ou=people,o=WorlCompany
uid=jsmith

dn: uid=john,ou=people,o=WorlCompany
uid=jsmith



dn: uid=jsmith,ou=people,o=WorlCompany

dn: uid=jsmith,ou=guests,o=WorlCompany



dn: uid=jsmith,ou=people,o=WorlCompany

dn: uid=john,ou=guests,o=WorlCompany

Concepts : LDAP, modèle fonctionnel : Authentification

- Authentification et contrôle
- 3 opérations : `bind, unbind, abandon`
  - `bind` = connexion.
  - `unbind` = déconnexion
  - `abandon`= le client indique au serveur qu'il laisse tomber la requête qu'il avait envoyé. Celui-ci abandonne alors le process.

Concepts : LDAP, modèle de sécurité

- Le modèle de sécurité décrit le moyen de protéger les données de l'annuaire des accès non autorisés.
- La sécurité se fait à plusieurs niveaux :
  - par l'authentification pour se connecter au service,
  - par un modèle de contrôle d'accès aux données,
  - par le chiffrement des transactions entre clients et serveurs ou entre serveurs.

L'authentification

LDAP est un protocole avec connexion : l'ouverture de session (`bind`) s'accompagne d'une identification et, éventuellement, d'un mot de passe (optionnel en V3).

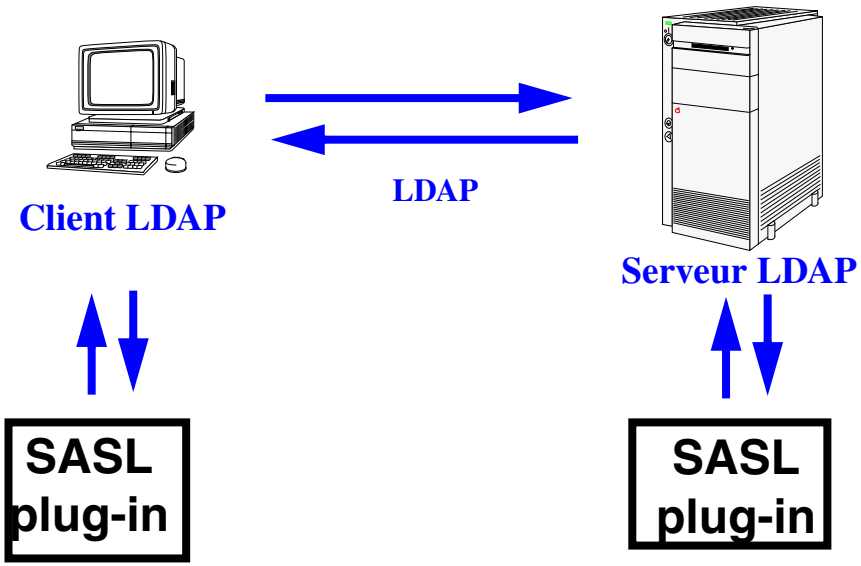
- Anonymous authentification - accès sans authentification permettant d'atteindre les données sans restrictions d'accès (V2, V3).
- Root DN authentification - accès administrateur (tous les droits) (V2, V3).
- Mot de passe en clair - un DN plus un password qui transite en clair sur le réseau (V2, V3).
- Kerberos V4 (V2)
- Mot de passe + SSL (LDAPS) ou TLS - la session est chiffrée et le mot de passe ne transite plus en clair.
- Certificats sur SSL - échange de certificats SSL (clefs publiques/privées).
- Simple Authentification and Security Layer (SASL) - mécanisme externe d'authentification (V3).

Concepts : LDAP, modèle de sécurité

❑ SASL

Simple Authentification and Security Layer (SASL) est défini par le RFC 2222 et permet d'ajouter des mécanismes d'authentification à des protocoles orientés connexion (~ plug-in). SASL est implanté dans LDAPv3.

Les mécanismes supportés par SASL sont Kerberos, S/Key, GSSAPI entre autres.



Le contrôle d'accès

Le serveur attribue à l'utilisateur identifié, des droits d'accès aux données (lecture, écriture, recherche et comparaison), qui lui ont été définis par l'administrateur sous la forme d'ACLs.

Pas encore normalisé par l'IETF donc non compatibles entre serveurs.

- Netscape Directory : sous la forme d'un attribut Access Control Items (`aci`)
- OpenLDAP : sous la forme de directives de contrôle d'accès dans `slapd.conf`

Concepts : LDAP, modèle de sécurité

- Le contrôle d'accès (suite)

Les ACLs peuvent être "placées" au niveau des entrées, au sommet de l'arbre ou sur un sous-arbre.

Elles agissent sur les entrées ou certains de leurs attributs.

Elles s'appliquent à des individus ou à des groupes, mais aussi suivant les adresses IP ou les noms de domaine des clients ou les jours et heures.

Le placement et la portée des ACLs dépendent des capacités du logiciel.

Concepts : LDAP, modèle de sécurité

- Expression générique des ACLs :

```
<quoi> <qui> <comment>
      <quoi> : point d'entrée de l'annuaire auquel s'applique la règle
      <qui> : à qui s'appliquent ces droits
      <comment> : opérations autorisées/refusées
```

<comment>	<qui>
Read	Tout le monde
Write	Un utilisateur
Search	Un groupe d'utilisateur
Compare	Une machine
Selfwrite	
Add	
Delete	

Exemple openldap :

```
access to * by self write
       by * read
```

Le chiffrement

LDAPv3 supporte le chiffrement des transactions (entre clients et serveurs ou entre serveurs) via l'utilisation de SSL (`ldaps`) ou de son successeur, TLS (`startTLS extended operation`).

SSL ou TLS servent également pour l'authentification par certificats :

- permet au client de prouver son identité au serveur et, en retour, à celui-ci d'en faire de même vis à vis du client.

Concepts : LDAP, modèle de duplication

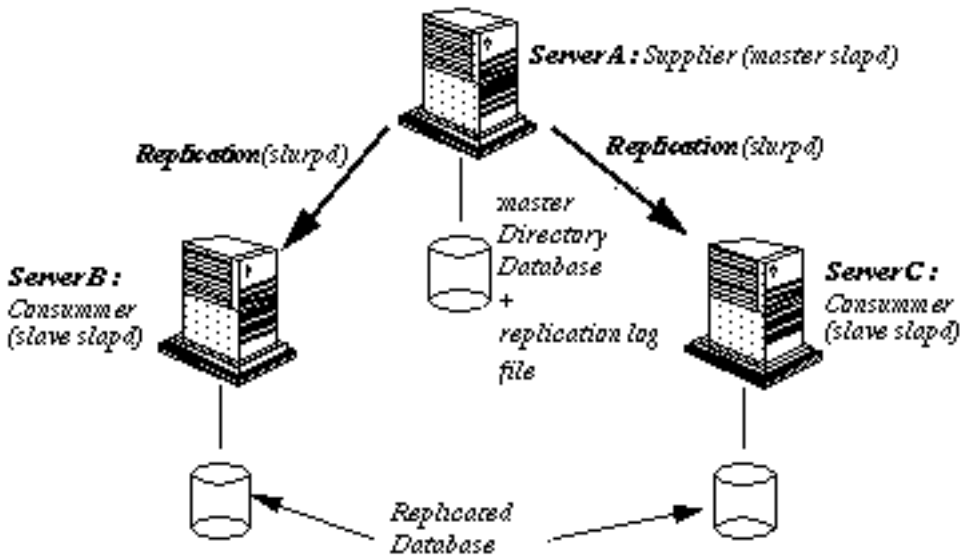
- Le modèle de duplication (replication service) définit comment dupliquer l'annuaire sur plusieurs serveurs.
- Dupliquer l'annuaire peut pallier à :
  - une panne de l'un des serveurs,
  - une coupure du réseau,
  - surcharge du service.

et garantir la qualité de service : temps de réponse et sûreté de fonctionnement.

- Permet également :
  - d'améliorer les performances en plaçant les serveurs près des clients
  - de répartir le travail entre plusieurs serveurs (load balancing)
  - de gérer les entrées localement et de les diffuser sur plusieurs sites.
- Pas encore standard, mais est proposé par la plupart des serveurs.
- L'IETF prépare le protocole LDUP.

Concepts : LDAP, modèle de duplication

- La duplication met en jeu plusieurs serveurs : les supplier servers fournissent les données, les consumer servers les reçoivent.
- Les informations de configuration décrivant les suppliers, les consumers et quelles données ils échangent, forment le replication agreement.



Concepts : LDAP, modèle de duplication

- On peut dupliquer
  - l'arbre entier ou seulement un sous arbre,
  - une partie des entrées et de leurs attributs qu'on aura spécifiés via un filtre du genre :

    « on ne duplique que les objets de type personne »

    « on ne duplique que les attributs non confidentiels » (annuaire interne vs. annuaire externe)
- Plusieurs manières de synchroniser les serveurs :
  - mise à jour totale
  - mise à jour incrémentale...
- Plusieurs stratégies de duplications :
  - single-master replication
  - multiple-master replication
  - cascading replication.

Concepts : LDAP, modèle de duplication

- La duplication se fait en temps-réel ou à heure fixe (scheduling replication).

- Deux précautions :
  - les serveurs doivent tous utiliser le même schéma de données,
  - les règles d'accès aux données dupliquées doivent être dupliquées.

La mise en œuvre de la duplication nécessite de la prévoir au moment du design du DIT.

LDAP : APIs

- Ces bibliothèques de programmation permettent de créer des applications annuaire-compatibles.
- Les APIs disponibles actuellement :
  - U-M LDAP SDK -- C (UMICH, OpenLDAP)
  - Innosoft LDAP Client SDK (ILC-SDK) -- C (InnoSoft)
  - Netscape Directory SDK -- Java, C (Netscape)
  - PerLDAP Modules -- Perl (Netscape)
  - Net- LDAPapi -- PERL (GNU)
  - Java Naming and Directory Interface (JUNI) -- Java (SUN)
  - Active Directory Service Interface (ADSI) -- COM (Microsoft)

LDAP : LDIF

- LDAP Data Interchange Format (LDIF) est le standard de représentation des entrées sous forme texte.
- Utilisé pour afficher ou modifier les données de la base suivant deux modes :
  - faire des imports/exports de base,
  - faire des modifications sur des entrées.
- Le format utilisé est l'ASCII. Toute valeur d'attribut ou tout DN qui n'est pas ASCII, est codé en base 64.

LDAP : LDIF

- Mode import

La forme générale est :

```
dn: <distinguished name>
objectClass: <object class>
objectClass: <object class>
[...]
attribute type:<attribute value>
attribute type:<attribute value>
[...]
```

Un entrée de type personne se présente de la manière suivante :

```
dn: cn=June Rossi, ou=accounting, o=Ace Industry, c=US
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: June Rossi
sn: Rossi
givenName: June
mail: rossi@aceindustry.com
userPassword: {sha}KDIE3AL9DK

dn: cn=Walter Scott, ou=accounting, o=Ace Industry, c=US
objectClass: top
objectClass: person
objectClass: organizationalPerson
```

LDAP : LDIF

❑ Mode modification

La forme générale est :

```
dn: distinguished name
changetype identifier
change operation identifier
list of attributes...
-
change operation identifier
list of attributes ...
```

Le caractère «-» spécifie le séparateur entre 2 instructions

Pour créer un nouvel enregistrement	**changetype: add**
Pour détruire un enregistrement	**changetype: delete**
Pour renommer une entrée	**changetype: modrdn**
Pour modifier un enregistrement	**changetype: modify**

-> Un opérateur de modification doit alors être spécifié.

- **add : ajouter des attributs et leurs valeurs.**
- **replace : remplacer des valeurs d'attributs par d'autres.**
- **delete : détruire l'attribut spécifié.**

LDAP : LDIF

❑ Exemple :

- Ajouter le numéro de téléphone et le nom du manager pour la personne «Lisa Jangles».

```
dn: cn=Lisa Jangles, ou=Sales, o=Ace Industry, c=US
changetype: modify
add: telephonenumber
telephonenumber: (408) 555-2468
-
add: manager
manager: cn=Harry Cruise, ou=Manufacturing, o=Ace Industry, c=US
```

LDAP : LDIF

- Le format utilisé dans LDIF est l'ASCII.

Toute donnée non ASCII doit être encodé en base 64. Dans ce cas le séparateur entre le type et la valeur de l'attribut est « :: ».

```
jpegPhoto:: /9j/4AAQSkZJRgABAQAAAQABAAD//gBHQ1JFQVRPUjogWFYgVmVyc2lvbiAzLjEwI
   CBSZXY6IDEyLzE2Lzk0ICBRdWFsaXR5ID0gNzUsIFNtb290aGluZyA9IDAK/9sAQwAIBgYHBgUIB
   wcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQy/
```

LDAP V3 utilise le jeu de caractères Unicode Transformation Format-8 (UTF-8) pour les attributs de type texte et les DNs.

UTF- 8 englobe tous les jeux de caractères (isoLatin, Shift- JLS...),

- annuaires multilingues : avec l'option language code de l'attribut (extension proposée par l'IETF) ().

```
description;lang-fr : texte en français
description;lang-ja : le même en japonais

(le code suit le standard ISO 639)
```

Les URLs LDAP

- Les URLs LDAP (RFC-1959) permettent aux clients Internet d'avoir un accès direct au protocole LDAP.

```
syntaxe :
ldap[s]://<hostname>:<port>/<base_dn>?<attributes>?<scope>?<filter>

<base_dn>     : DN de l'entrée qui est le point de départ de la recherche
<attributes>  : les attributs que l'on veut consulter
<scope>       : la profondeur de recherche dans le DIT à partir du <base_dn>
      - base : s'arrête au niveau courant (par défaut)
      - one  : descend d'un niveau
      - sub  : parcourt tous les sous-niveaux
<filter>      : filtre de recherche, par défaut (objectClass=*)

exemples :

ldap://ldap.netscape.com/ou=Sales,o=Netscape,c=US
ldap://ldap.worldcompany.com/cn=John%20Smith,ou=people,o=worldcompany
ldap://ldap.worldcompany.com/o=worldcompany?mail,uid?sub?(sn=Smith)
```

Déploiement

Déployer un service d'annuaire LDAP, c'est réfléchir à :

- la nature des données que l'on y met,
- la manière dont on les récupère,
- l'utilisation que l'on compte en faire,
- la façon de gérer le tout.

La mise en place d'un annuaire LDAP met donc en jeu plusieurs phases de conception que l'on va passer en revue.

Déploiement : besoins en service d'annuaire

- Un annuaire LDAP = entrepôt d'informations facilement accessibles aux utilisateurs ou aux applications.

- Déployer un système d'annuaire se fait généralement sous la contrainte de la mise en place ou du remplacement d'une application.

  - ⇨ Se poser la question d'élargir le service à d'autres types d'applications
  - ⇨ Envisager toutes les applications possibles, actuelles ou futures, d'un annuaire LDAP.

Déploiement : Données nécessaires

Il s'agit :

⇨ d'inventorier, suivant les applications, la liste des données à inclure dans le système d'information et leurs caractéristiques :

- format
- taille
- nombre d'occurrence
- droits d'accès
- dynamiques ou statiques
- partagées ou spécifiques à une application

⇨ de déterminer par quelle source les obtenir et les maintenir à jour.

Déploiement : Données nécessaires

- Les sources de données courantes :
  - autre service d'annuaire ou bases systèmes (Unix NIS, DNS, NT domain controler...)
  - bases de données de l'organisation (base du personnel, base du PABX...)
  - fichiers textes ou feuilles de calcul d'utilisateurs
  - des bases propres à des applications (fichier `htpasswd` d'Apache, carnet d'adresses...)

- Les mécanismes de mise à jour envisageables :
  - synchronisation avec un SGBD
  - batches
  - saisie manuelle

Déploiement : Données nécessaires

- Choisir, en fonction des données retenues, quelles classes d'objets et types d'attributs utiliser.
- Les schémas standards ou fournis avec les serveurs conviennent en général aux besoins.
- En règle générale, éviter de modifier le schéma existant car risque de rendre son annuaire inutilisable par les applications clientes ou les autres serveurs.
- Préférable de rajouter une classe d'objet et exploiter le mécanisme d'héritage d'attributs des classes objets.

Déploiement : Données nécessaires

Par exemple création de la classe d'objet `Personne` fille de `inetOrgPerson` dans laquelle on définira les attributs nécessaires à ses besoins :

```
attributetype ( 1.3.6.1.4.1.999.1.1 NAME 'fonction'
   EQUALITY caseIgnoreMatch
   SUBSTR caseIgnoreSubstringMatch
   SYNTAX 1.3.6.1.4.1.1466.115.121.1.26 )

objectclass ( 1.3.6.1.4.1.999.2.1 NAME 'Personne' SUP inetorgperson
   DESC 'membre du personnel'
   MUST ( sn $ cn $ fonction )
   MAY ( uidNumber $ gidNumber $ homeDirectory $ loginShell ))
```

- Dans tous les cas :
  - → documenter son schéma pour en faciliter la maintenance et l'évolution.
  - → éviter de désactiver l'option de schema checking.

Déploiement : concevoir son modèle de nommage

Consiste à définir comment les entrées de l'annuaire vont être organisées, nommées et accédées.

- Dans cette phase, les paramètres qu'il faut prendre en compte sont :
  - Le nombre d'entrées prévu et son évolution ?
  - La nature (type d'objet) des entrées actuelles et futures ?
  - Vaudra-t-il mieux centraliser les données ou les distribuer ?
  - Seront-elles administrées de manière centrale ou faudra-t-il déléguer une partie de la gestion ?
  - La duplication est-elle prévue ?
  - Quelles applications utiliseront l'annuaire et imposent-elles des contraintes particulières ?
  - Quel attribut utiliser pour nommer les entrées et comment garantir son unicité ?

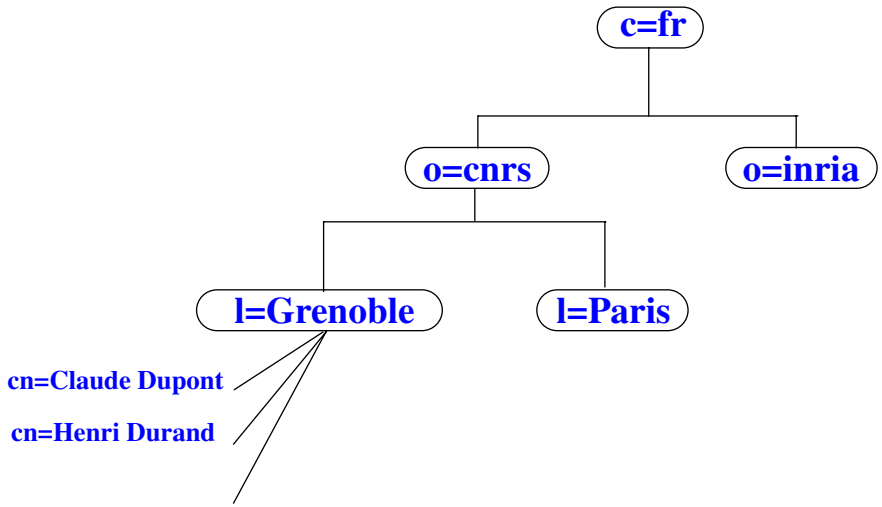
En fonction de ses priorités, on privilégiera tel ou tel espace de nommage.

Déploiement : concevoir son modèle de nommage

Design du Directory Information Tree

Le DIT X.500 est conçu dans l'optique d'un service global : il part du pays (top level) puis l'organisation, puis éventuellement la localisation...et il utilise l'attribut `cn` pour nommer les entrées.

Exemple de DIT à la X.500



Déploiement : concevoir son modèle de nommage

Design du DIT (suite)

Le modèle LDAP, n'impose pas une racine universelle du DIT car il renonce à être un service d'annuaire mondial.

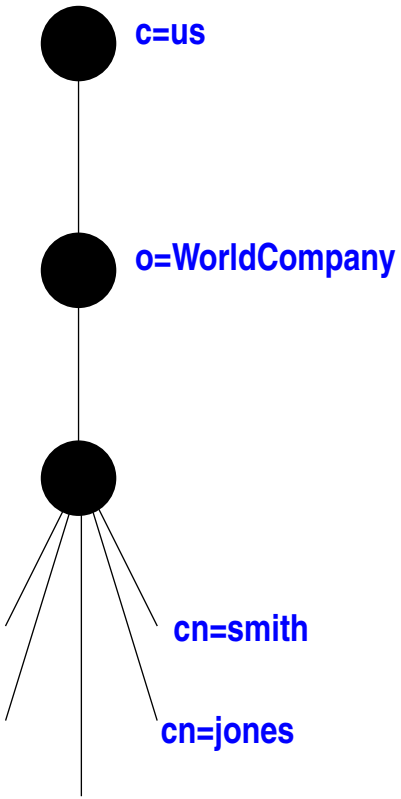
Dans ce cadre, le DIT peut être organisé de différentes façons :

- plat,
- découpé pour refléter l'organisation interne,
- branché par type d'objet,
- branché en vue de faciliter la duplication entre serveurs, la délégation de gestion, ou la définition de règles d'accès spécifiques à une branche.

Déploiement : concevoir son modèle de nommage

Design du DIT (suite)

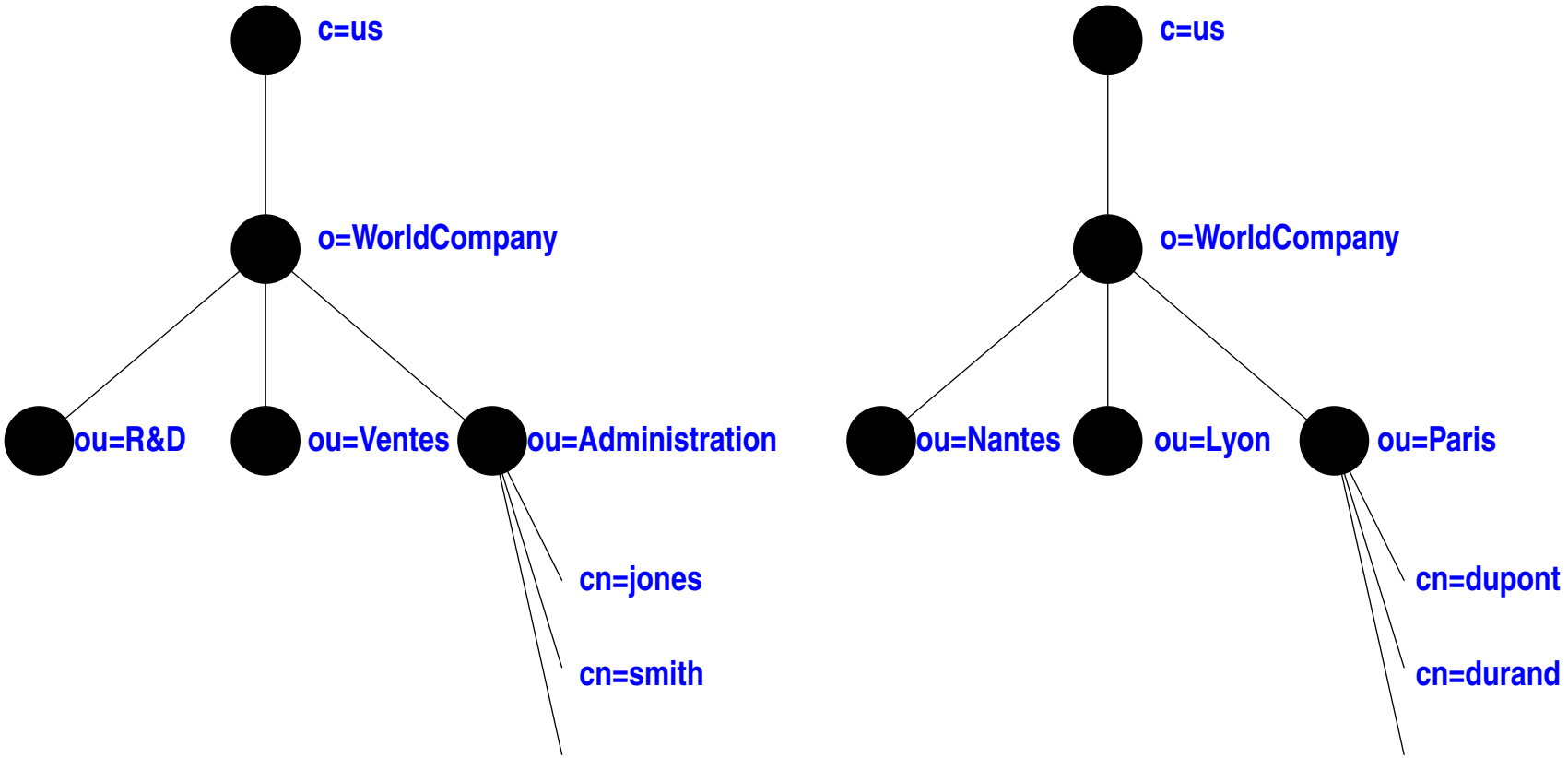
Exemple : arbre plat



Déploiement : concevoir son modèle de nommage

Design du DIT (suite)

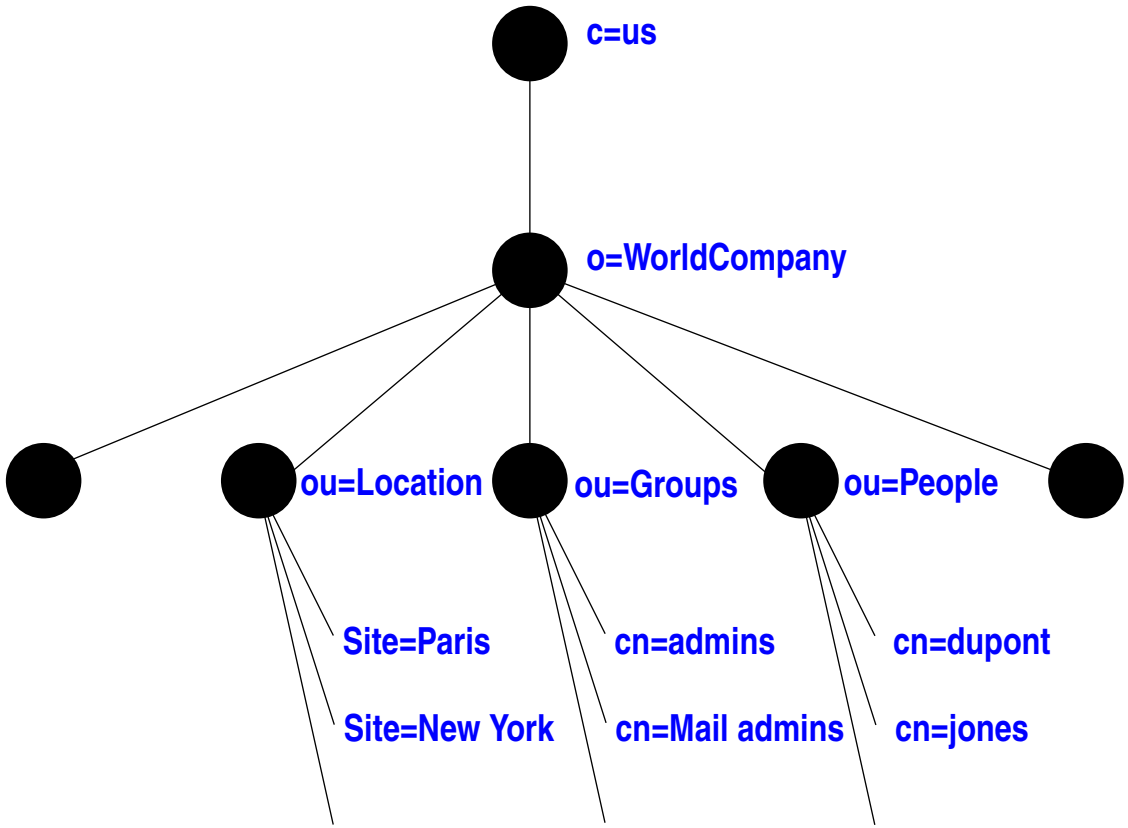
Exemple : branchage par service ou par localisation



Déploiement : concevoir son modèle de nommage

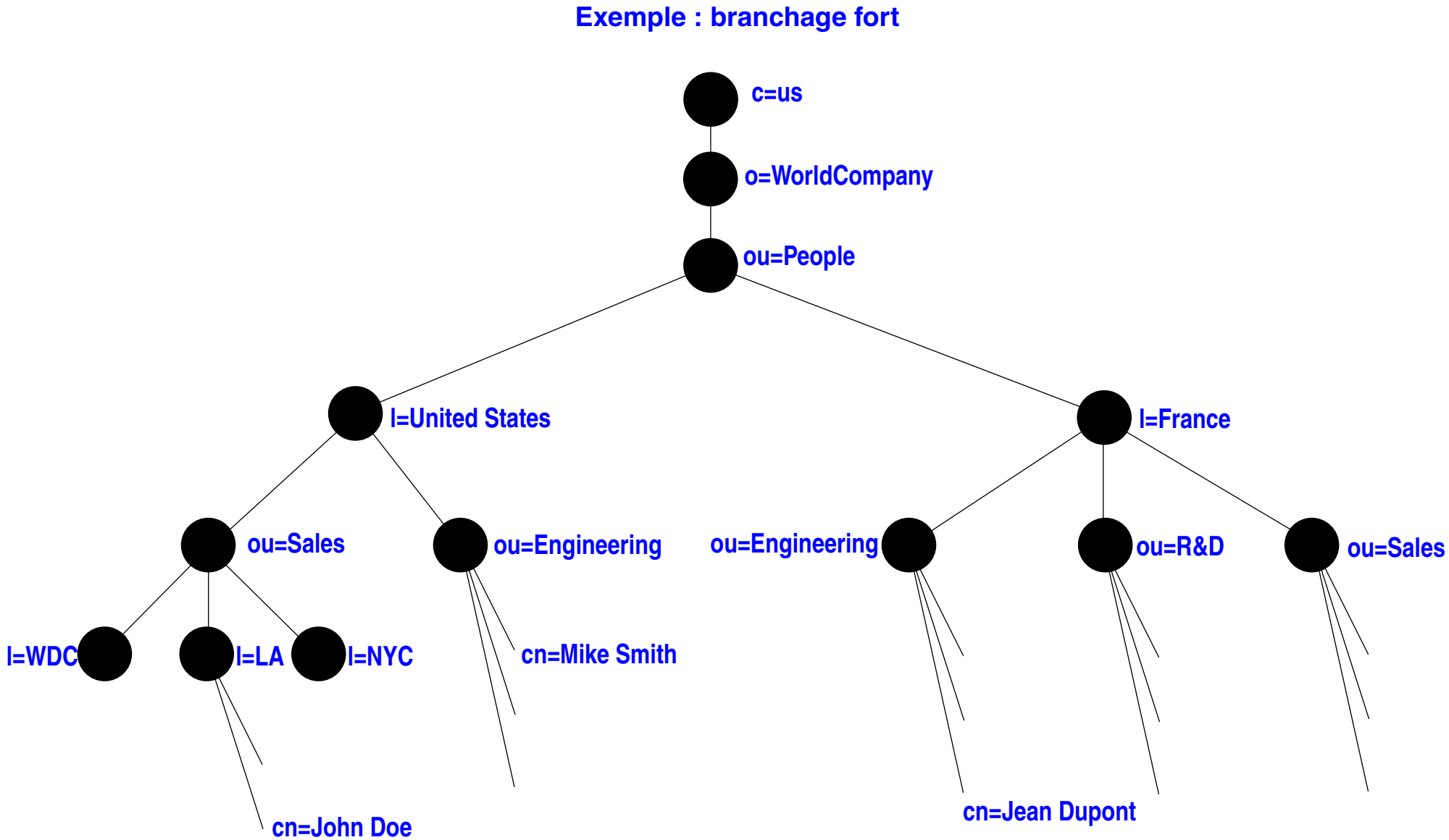
Design du DIT (suite)

Exemple : branchage par type d'objet



Déploiement : concevoir son modèle de nommage

Design du DIT (suite)



Déploiement : concevoir son modèle de nommage

Design du DIT : branchage fort ou faible ?

Fort : les plus	Faible : les plus
Reflète l'organisation interne. Minimise le problème de DNs identiques. Facilite le partitionnement des données entre plusieurs serveurs.	Pas de soucis de classification des entrées DN courts stabilité du DIT Meilleurs rapidité de recherche.
Fort : les moins	**Faible : les moins**
Longueur du DN. Problème si l'organisation change. Durée de recherche augmentée.	Risque de DNs identiques. Mal adapté au listage des entrées

Déploiement : concevoir son modèle de nommage

Choix du suffixe

Le suffixe = identifiant de l'annuaire.

Même si la base n'a qu'une vocation interne , elle peut à terme s'externaliser.

→ Choisir, si possible, un suffixe unique au monde.

Dans X.500 le top level est le pays, vient ensuite le nom de l'organisation, et éventuellement la localisation.
Ce qui donne par exemple comme suffixe : `o=World Company, c=us`

Aucun organisme de contrôle d'attribution des suffixes :

→ Pas de garantie de l'unicité de celui-ci.

Entre temps, l'Internet s'est développé :

→ NIC gère l'attribution des noms de domaines DNS.

Le choix du nom de domaine DNS comme suffixe de son annuaire est recommandé par l'IETF IDS group.

Déploiement : concevoir son modèle de nommage

Choix du suffixe (suite)

Il pourra s'exprimer sous deux formes :

- utilisation de l'attribut organization (o) :
  o=world-company.com
- utilisation de l'attribut Domain Component (dc) défini par le RFC 2377 :
  dc=world-company, dc=com

Cette dernière forme est préconisée par l'IETF.

Couplée avec le Service Record du DNS (SRV), permet de déterminer automatiquement le serveur LDAP à contacter, à partir du DN utilisé dans une requête.

le DN uid=jones,ou=people,dc=World-Company,dc=com **renvoie sur le domaine DNS** World-Company.com**.**

Requête sur l'entrée SRV **du DNS de** World-Company.com

_ldap._tcp.World-Company.com. IN SRV 0 0 389 ldap.World-Company.com

Déduction : serveur : ldap.world-company.com **- port :** 389

Déploiement : concevoir son modèle de nommage

Choix du suffixe (conclusion)

Pas de standard de design et pas de solution universelle :

→ faire des compromis visant à prendre la moins mauvaise solution, en essayant de définir les facteurs les plus contraignants.

- Prendre en compte son organisation : sa structure, sa taille, son évolution

- Prendre en compte l'usage de l'annuaire :

  - type de données
  - leur mode de gestion
  - type d'applications accèdant aux données

Déploiement : concevoir son modèle de nommage

Nommage des entrées : choix du RDN

Exemples :

```
dn = cn=robert jones,ou=people,dc=world-company,dc=com
dn = uid=rdupont,ou=people,dc=world-company,dc=com
```

- Problèmes :
  - garantir l'unicité
  - éviter les changements de DN
  - donner une information pertinente
  - prise en compte des clients

Recommendations de IETF : Identification des utilisateurs par leur email

```
dn = uid=rdupont@world-company.com,ou=people,dc=world-company,dc=com
```

Déploiement : Définir la topologie du service

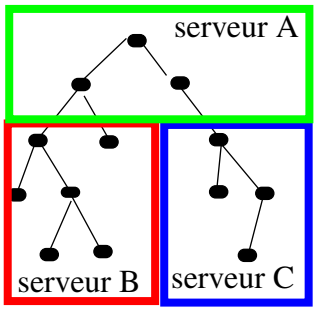
Analyser la manière dont le service d'annuaire LDAP va être rendu en termes de performance, de fiabilité et de facilité de gestion.

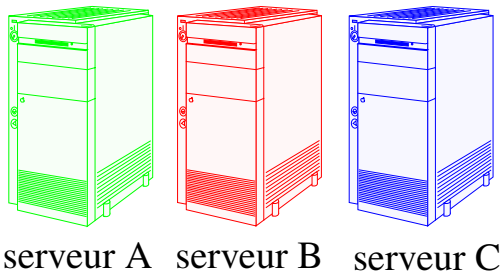
- Prendre en compte :
  - Les applications qui vont utiliser l'annuaire et leur nombre d'utilisateurs.
  - Les capacités du logiciel serveur qui va être choisi.
  - La topologie de son réseau.
  - Le design de son espace de nommage.
- Déterminer :
  - si la base sera centralisée ou répartie sur plusieurs serveurs.
  - le nombre de serveurs redondants à déployer et leur emplacement sur le réseau physique.

Déploiement : Définir la topologie du service

Le partitionnement

Consiste à éclater les données de l'annuaire sur plusieurs serveurs.





Il peut être imposé par :

- le volume d'entrées à gérer,
- leur gestion répartie sur plusieurs sites,
- les types d'accès au réseau physique,
- le mode d'organisation de la société.

Séparer les données ne veut pas dire forcément les dissocier : les standards LDAP et X.500 définissent des moyens de les relier (re-coller).

→ Ces moyens sont les services "referral service" et "replication service".

Déploiement : Définir la topologie du service

Le referral service

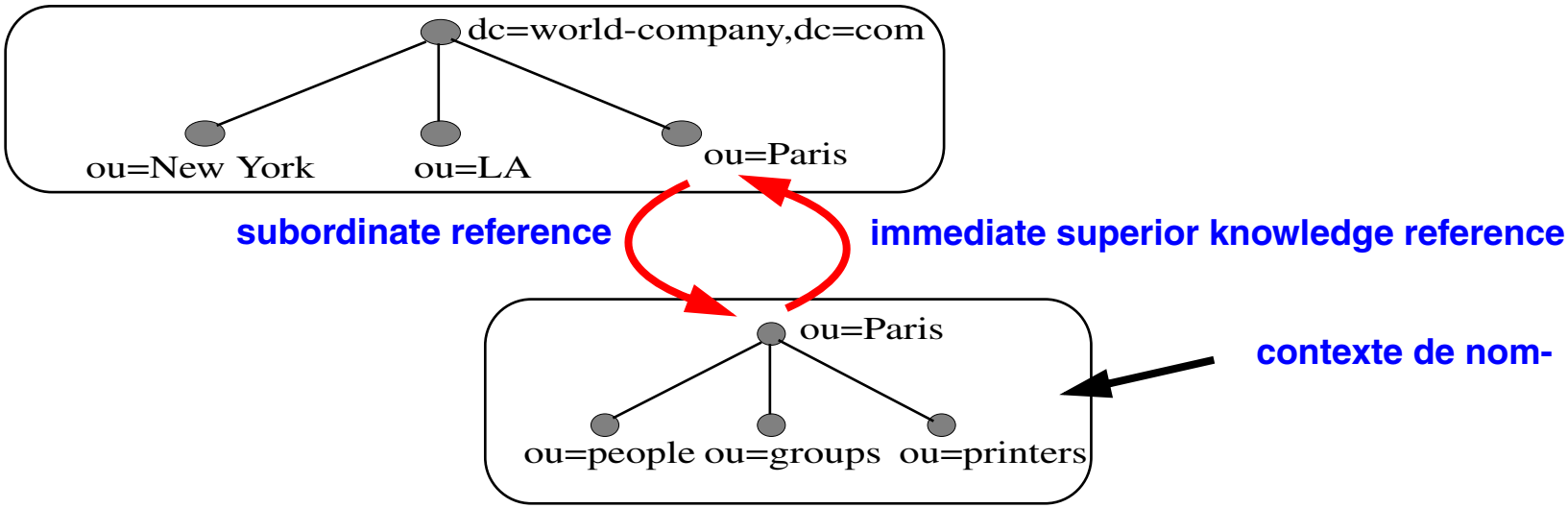
La résolution de nom est le mécanisme par lequel un serveur détermine quel objet de sa base désigne le DN qu'un client lui fournit.

- Si le DN est bien dans son contexte de nommage, il exécute la requête du client (`search, modify, bind`...),
- sinon il renvoie un signal object not found.

Déploiement : Définir la topologie du service

Le referral service (suite)

Les méthodes permettant de créer des liens virtuels entre des partitions d'annuaires sont appelées les knowledge references.



Les knowledge references permettent à un serveur de faire suivre les requêtes des utilisateurs lorsque l'objet recherché n'appartient pas à l'arbre qu'il gère.

Déploiement : Définir la topologie du service

Le referral service (suite)

Les serveurs LDAP utilisent deux méthodes pour faire suivre les requêtes le long de ces liens :

- Le Referral est une information que retourne au client le serveur LDAP, lorsque l'entrée recherchée n'appartient pas à son arborescence, lui indiquant vers quel serveur il doit re-formuler sa requête (via un URL LDAP). Le mécanisme de referral est standardisé dans le protocole LDAPv3.
- Le chaînage (`chaining`) est un mécanisme où c'est le serveur qui se charge de contacter un autre serveur pour le compte du client et lui retourne la réponse. Le chaînage n'est pas un standard du protocole LDAP, il est plutôt utilisé dans les logiciels X.500.

Le choix entre l'une ou l'autre méthode dépend essentiellement des fonctionnalités du serveur choisi.

Déploiement : Définir la topologie du service

Le referral service (suite)

les serveurs ne les positionnent pas tous au même endroit.

Netscape Directory utilise deux types de referral :

- le default referral
- le smart referral.

Le default referral est indiqué au niveau de la racine du serveur et agit comme une redirection par défaut pour toute requête hors espace de nommage.

Le smart referral est placé dans une entrée quelconque et agit comme un lien symbolique vers une autre entrée d'un autre serveur.

Les deux utilisent les URLs LDAP pour re-diriger la requête.

Déploiement : Définir la topologie du service

Le referral service (suite)

Le default referral est positionné dans le fichier `slapd.conf` de Netscape Directory ou OpenLDAP sous la forme d'une ligne :

```
referral ldap://ldap.airius.com:389/o=airius.com
```

Les smart referrals sont stockés dans l'attribut `ref` de l'objet auquel on a rajouté la classe d'objet referral.

Exemple en LDIF :

```
dn: ou=Paris, dc=world-company, dc=com
objectclass: top
objectclass: organization
objectclass: referral
ou: Sophia
description: Filiale Paris
l: Paris
ref: ldap://ldap.paris.world-company.fr:389/ou=paris
```

Déploiement : Définir la topologie du service

Le replication service

La duplication consiste à recopier le contenu de tout ou partie de son arbre sur un autre serveur (voir § LDAP-Concepts)

Son but :

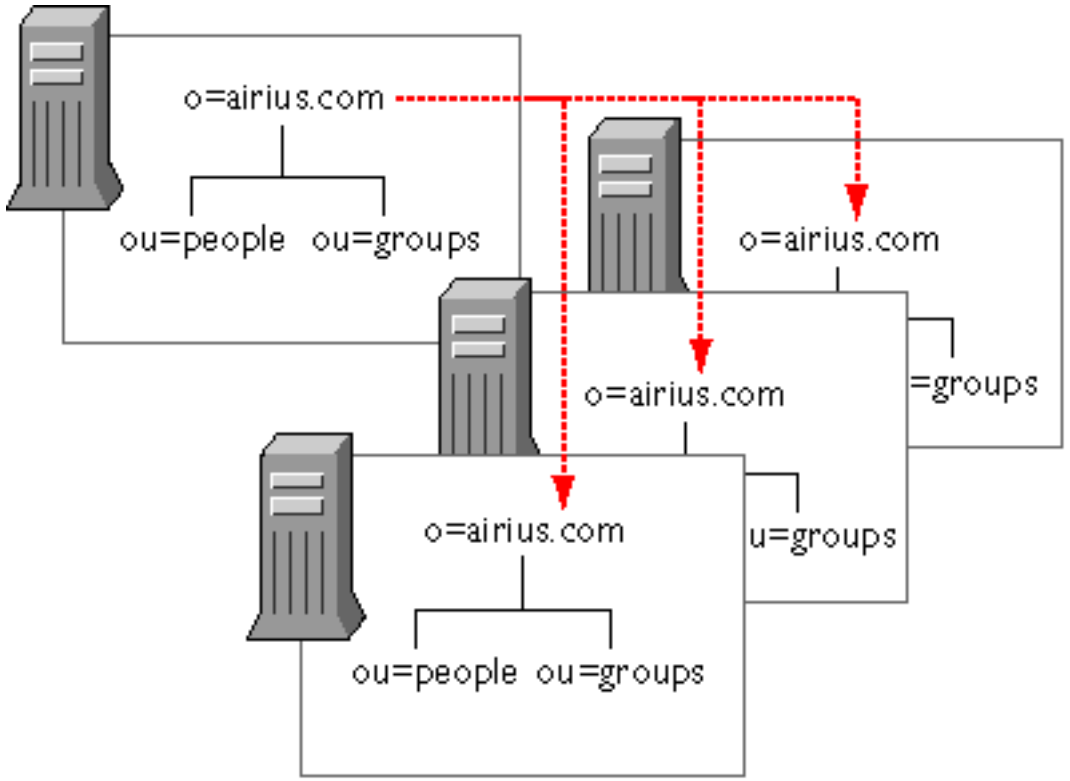
- rapprocher le service du réseau physique des clients (performances),
- répartir la charge sur plusieurs serveurs (load balancing),
- assurer une redondance en cas de panne (disponibilité),
- gérer localement des entrées et les diffuser dans l'organisation (partitionnement).

Le replication service est LE moyen d'assurer un service d'annuaire fiable, hautement disponible, et performant.

Déploiement : mettre en service la duplication

Les différents modes de duplication.

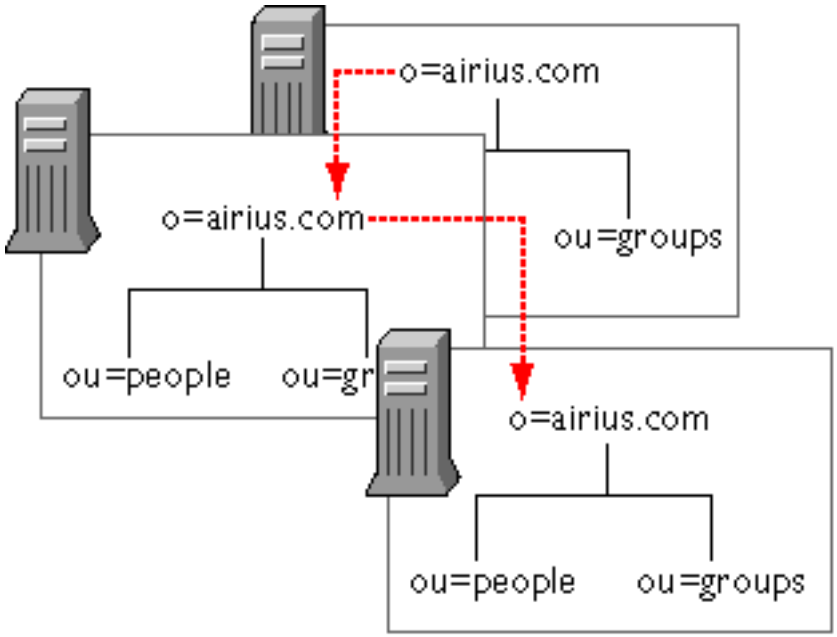
Duplication de l'arbre entier sur 1 ou plusieurs consumers (source Netscape)



Le supplier (read-write) duplique sur un ou plusieurs consumers (read-only).

Déploiement : mettre en service la duplication

Duplication de l'arbre entier en cascade (source Netscape)

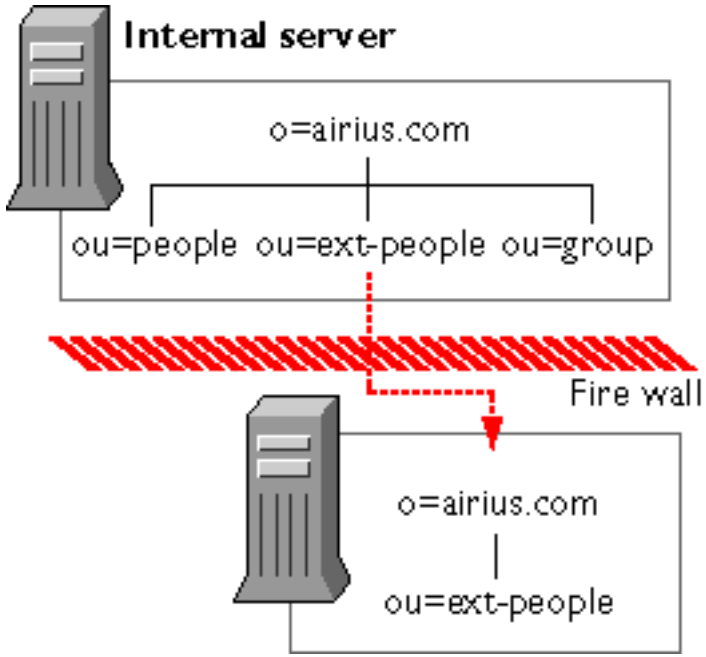


Le supplier duplique sur un consumer qui lui-même duplique sur un autre.

Cas où les liaisons réseau entre sites sont de qualité variable.

Déploiement : mettre en service la duplication

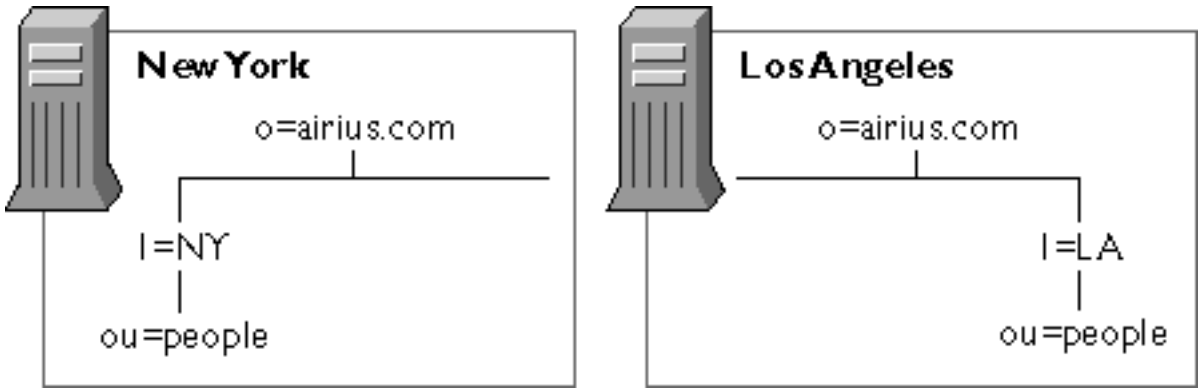
Duplication d'une partie de l'arbre (source Netscape)



Le supplier coupé de l'extérieur ne duplique qu'une branche publique de l'arbre sur un consumer accessible depuis l'internet.

Déploiement : mettre en service la duplication

Duplications croisées 1 (source Netscape)

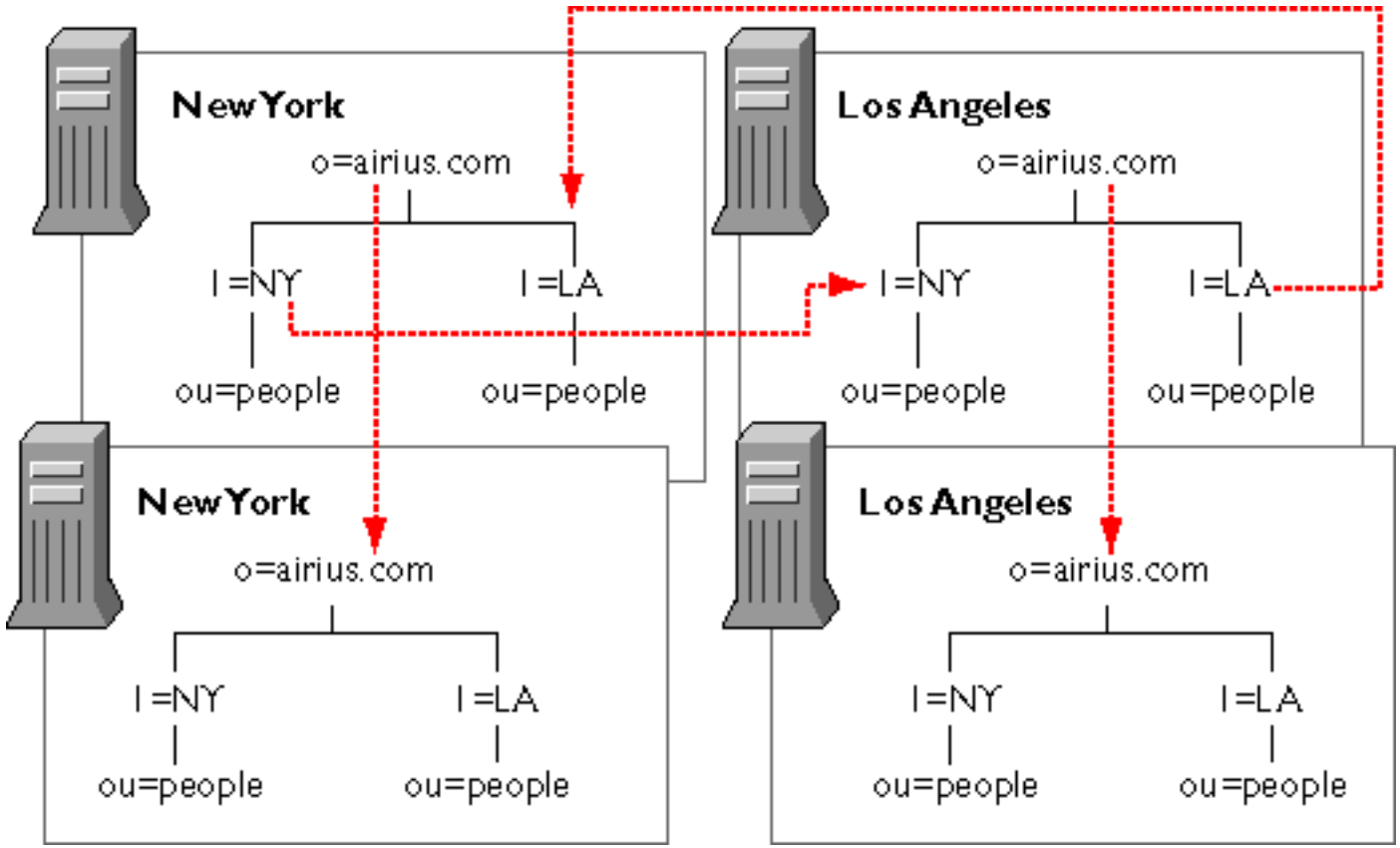


La société Airius a deux agences à NY et LA qui gèrent chacune leur branche du serveur d'annuaire.

La duplication est mise en œuvre pour ramener les branches distantes localement (performance) et assurer une redondance de tout l'arbre en local (disponibilité).

Déploiement : mettre en service la duplication

Duplications croisées 2 (source Netscape)



Les branches sont dupliquées réciproquement sur chaque site.
De plus, l'arbre entier est dupliqué en local.

Déploiement : mettre en service la duplication

Répartir la charge en utilisant le DNS round robin

Ce mécanisme du DNS permet de configurer plusieurs adresses IP pour un même hostname.
Le service DNS fait une rotation de l'ordre des numéros IP, lorsqu'il retourne le résultat d'une requête sur le nom du serveur LDAP.

- Choisir la stratégie de duplication

Consiste à définir le flux de mise à jour des données entre les serveurs de duplication.

Plusieurs méthodes existent :

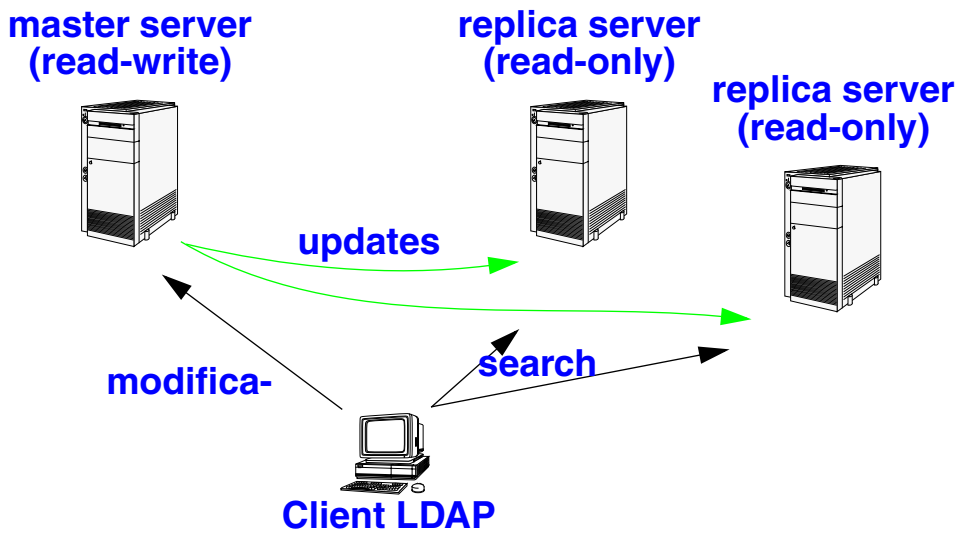
- Single-master replication
- Floating-master replication
- Multi-master replication

Déploiement : mettre en service la duplication

Single-master replication

Un serveur en lecture-écriture (master) et tous les serveurs replicas sont read-only. Les modifications des clients sont re-dirigés par des knowledge references sur le master.

Cette solution présente une faiblesse si le master est en panne.



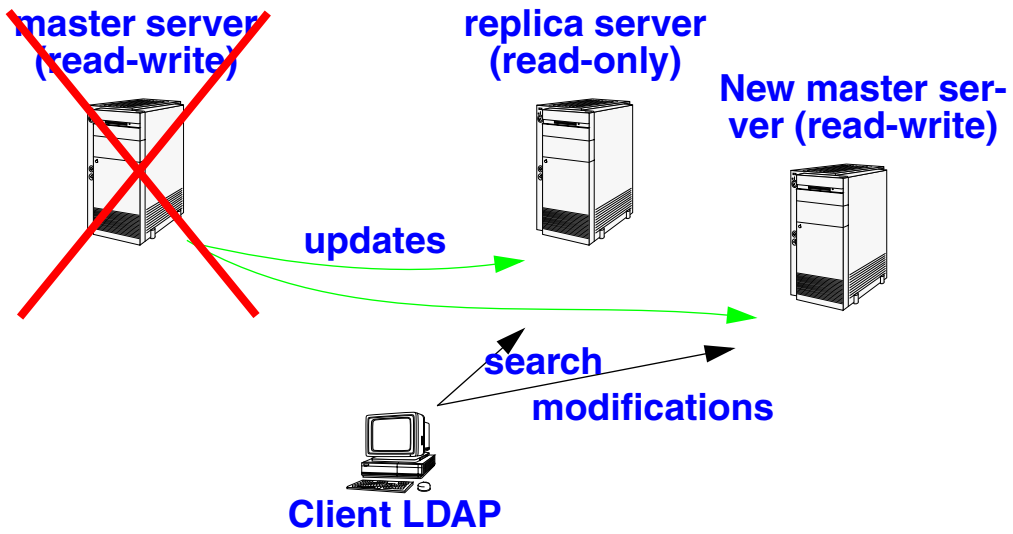
Déploiement : mettre en service la duplication

Floating-master replication

Nouveau master en secours en cas de panne du master. Mécanisme de synchronisation lorsque le serveur repart.

Utilisé par Windows NT 4.0 pour ses contrôleurs de domaines (PDC, SDC).

Il n'est pas encore adopté par les logiciels serveurs LDAP.



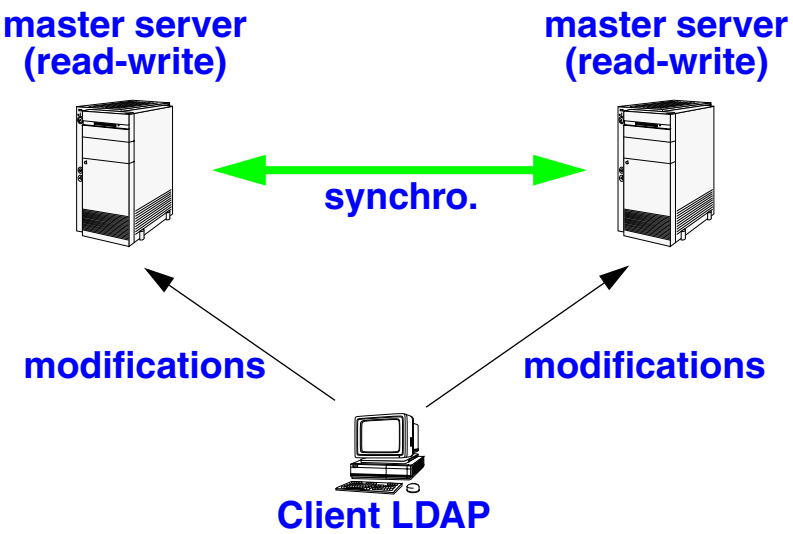
Déploiement : mettre en service la duplication

Multi-master replication

Plusieurs serveurs read-write sur lesquels les clients peuvent faire les modifications.

Des mécanismes de synchronisation se chargent de gérer les conflits (règle du dernier arrivé l'emporte en utilisant l'attribut `timestamp` des entrées).

Ces 2 derniers modes de replication sont en cours d'étude à l'IETF pour intégrer au standard LDAP.



Déploiement : mettre en service la duplication

Duplication totale/incrémentale

La synchronisation peut être totale ou incrémentale. Dans ce cas, le processus de synchronisation utilise un historique des mises à jours.

- Duplication à heures fixes

  Certains logiciels permettent de différer les mises à jours à certains horaires.

  → Utile dans le cas de liaisons réseau non permanentes ou chargées par périodes.

- Duplication basée sur les attributs

  X.500 prévoit la possibilité de filtrer les données dupliquées par une sélection d'attributs.

  → sélection d'objets via filtre sur l'attribut objectclass,

  → sélection de certains attributs (`uid, password`...) pour filtrer les données confidentielles, par ex.

Déploiement : mettre en service la duplication

- Schéma et duplication

  A partir du moment où ils partagent les mêmes données, il est impératif que supplier servers et consumer servers utilisent le même schéma.

- Contrôle d'accès et duplication

  Le contrôle d'accès se fait via des ACLs. Il est nécessaire de dupliquer ces ACLs pour que les mêmes protections s'appliquent sur les données dupliquées et originales...

  → ...Consumers et suppliers doivent interpréter de la même manière ces ACLs (pas normalisées...) : donc utiliser le même logiciel...

  Pratiquement tous les logiciels stockent les ACLs en tant qu'attribut d'entrées de l'annuaire.
  Parfois ces ACLs s'appliquent aux entrées inférieures (scope)...

  → ...Vérifier que ces ACLs sont bien dans la partie dupliquées du DIT ou comment c'est pris en compte par le logiciel.

Déploiement : mettre en service la duplication

Méthodologie de mise en œuvre

- connaître la topologie du réseau physique.
- connaître l'emplacement des applications clientes et la charge générée.
- choisir la méthode de duplication en fonction des capacités du logiciel
- synchronisation totale ou incrémentale
- single/multi-master replication
- scheduling replication ou pas
- outils d'audit de la duplication
- positionner au mieux sur le réseau les serveurs LDAP replicas en fonction des caractéristiques du logiciel, du réseau et de l'emplacement des clients.

Attention : le suffix doit toujours être le même entre les serveurs replicas.

Déploiement : mettre en œuvre le partitionnement

Rappel : le partitionnement est une solution pour les trop gros volumes d'entrées (> 10000), ou des organisations éclatées en unités autonomes.

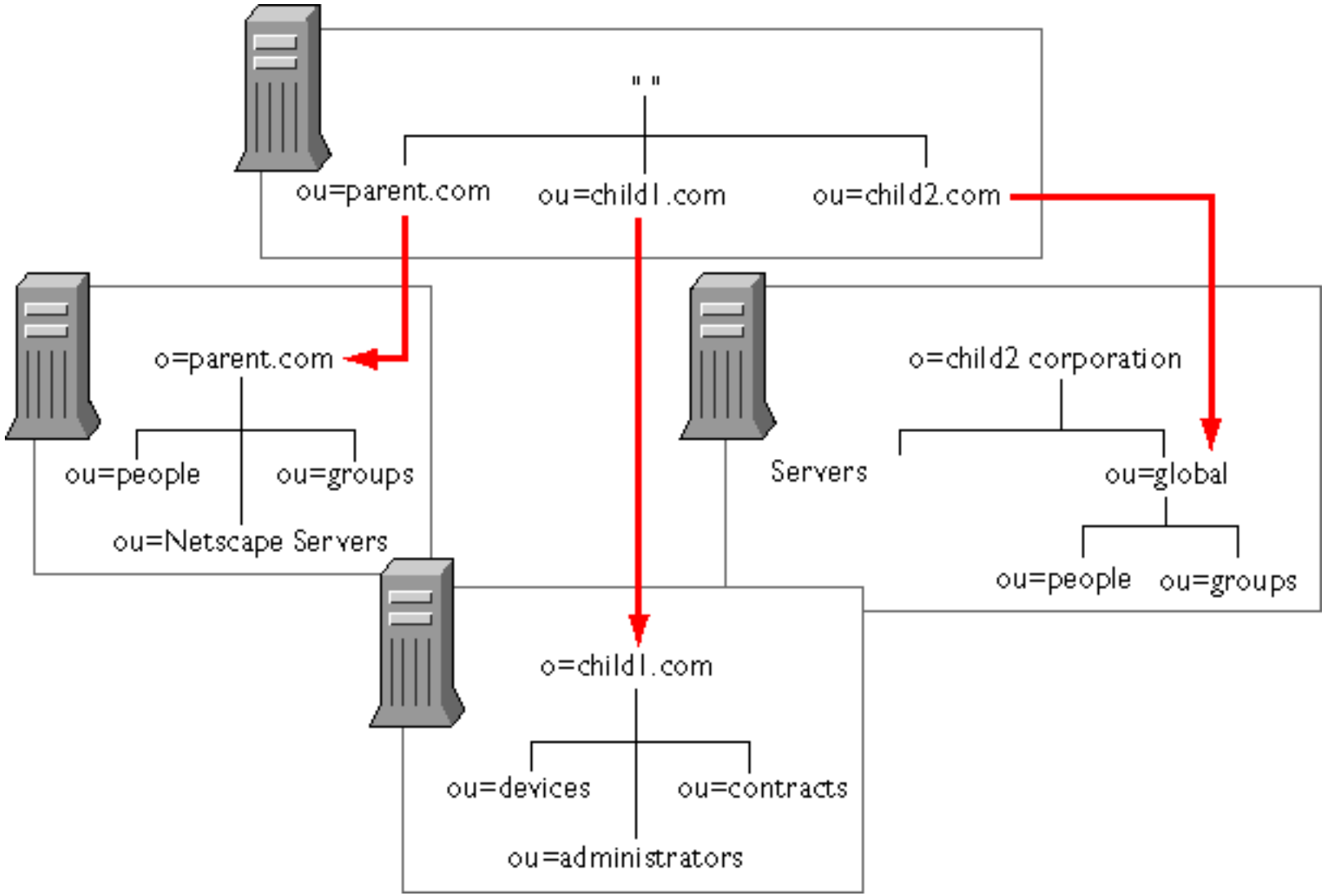
Les mécanismes de referral peuvent être une alternative à la duplication.

Quelques précautions :

- limiter les referrals à des suffixes ou des branches principales de l'arbre (ne pas s'en servir comme alias pour des entrées),
- maintenir la cohérence des liens... et vérifier la disponibilité du serveur distant,
- attention au contrôle d'accès et à l'authentification : les authentifications et les règles d'accès du serveur initial ne s'appliquent plus aux données du serveur pointé,
- attention au temps de réponse : traversée de réseaux WAN,
- problème de sécurité : les données transitent sur les réseaux WAN...

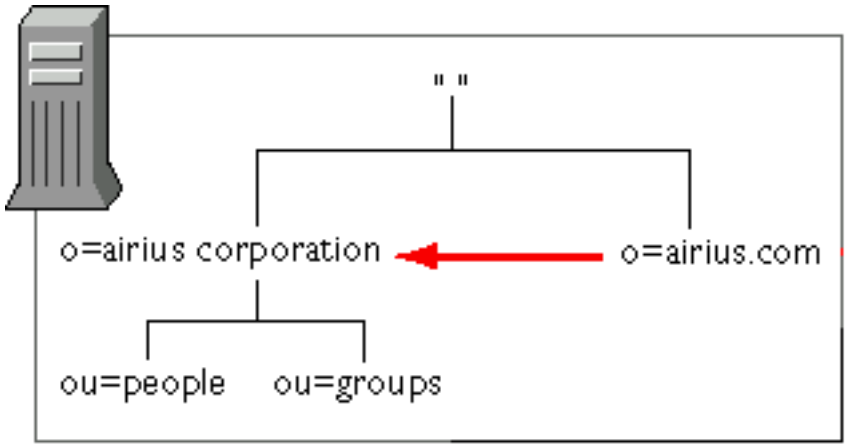
Déploiement : mettre en œuvre le partitionnement

Cas d'une organisation large : multi-site, multi-suffixe, multi-serveur (source Netscape)



Déploiement : mettre en œuvre le partitionnement

Cas d'un changement de suffix : un serveur, plusieurs suffixes (source Netscape)



Déploiement : sécuriser le service

Les aspects sécurité et confidentialité doivent être pris en compte dès la phase de conception. Quels sont les aspects à étudier ?

- Les accès non autorisés.
- Les attaques de type denial-of-service.
- Les droits d'accès aux données.

Le gros du travail est de déterminer les règles d'accès aux données.

Le serveur peut être de type read-only ou read-write. Dans les deux cas il faut déterminer pour chaque attribut :

→ Quel est son niveau de confidentialité (un numéro de sécurité sociale est une donnée plus sensible qu'une adresse mail) ?

→ Quel utilisateur ou quelle application pourra y accéder en lecture (tout le monde, certains utilisateurs, uniquement les administrateurs...) ou en écriture (utilisateur, manager, administrateur) ?

Déploiement : sécuriser le service

Les mécanismes qui peuvent être mis en œuvre sont ceux que l'on retrouve dans nombre de services/serveur de l'Internet :

- L'authentification
- Les signatures électroniques
- Le chiffrement
- Le filtrage réseau
- Les règles d'accès (ACLs LDAP) aux données
- L'audit des journaux

Déploiement : sécuriser le service

Mettre en place des règles de contrôle d'accès

→ Etape 1 : analyser pour chaque attribut son mode d'accès :

Tableau 1 :

Attribut	Personne	Droit d'accès
`cn,sn,givenname`	**tous** **administrateur**	**lecture** **lecture/modification**
uid	**utilisateurs authentifiés** **administrateur**	**lecture** **lecture/modification**
`telephoneNumber`	**tous** **propriétaire** **administrateur**	**lecture** **lecture/modification** **lecture/modification**
`employeeNumber`	**tous** **manager** **administrateur**	**lecture** **lecture/modification** **lecture/modification**

Déploiement : sécuriser le service

Mettre en place des règles de contrôle d'accès

→ Etape 2 : traduire ces règles en aci (LDIF)

Exemple pour attribut `telephoneNumber`

```
règle pour tous
aci: (target="ldap:///ou=people,dc=world-company,dc=com)
  (targetattr="telephonenumber")
  (version 3.0;acl "anonymous read-search access";
   allow (read,search,compare) (userdnattr="manager");)

règle pour administrateur
aci: (target="ldap:///dc=world-company,dc=com)
  (targetattr="*")
  (version 3.0;acl "Admin write access";
   allow (write) (userdn="ldap:///cn=Directory Manager");)

règle pour propriétaire
aci: (target="ldap:///ou=people,dc=world-company,dc=com)
  (targetattr="telephonenumber||roomnumber||userpassword")
  (version 3.0;acl "self write access";
   allow (write) (userdn="ldap:///self");)
```

Déploiement : sécuriser le service

Mettre en place des règles de contrôle d'accès

→ Etape 2 : traduire ces règles en aci (suite)

Exemple pour attribut `employeeNumber`

règle pour manager

```
aci: (target="ldap:///ou=people,dc=world-company,dc=com)
  (targetattr="employeenumber")
  (version 3.0;acl "manager write access";
  allow (read,write) (userdnattr="manager");)
```

avec l'attribut `manager` indiquant le DN du manager de l'entrée

```
dn: cn=John Smith,ou=people,dc=world-company,dc=com
objectclass: top
objectclass: person
cn: John Smith
manager: cn=Bill Baxter, ou=people,dc=world-company,dc=com
...
```

Déploiement : sécuriser le service

Mettre en place des règles de contrôle d'accès

→ Etape 2 : traduire ces règles en aci (suite)

Exemple d'utilisation d'un filtre (targetfilter) permettant aux membres du groupe *Servadm Manager* de modifier une partie des attributs des personnes membres du service administratif.

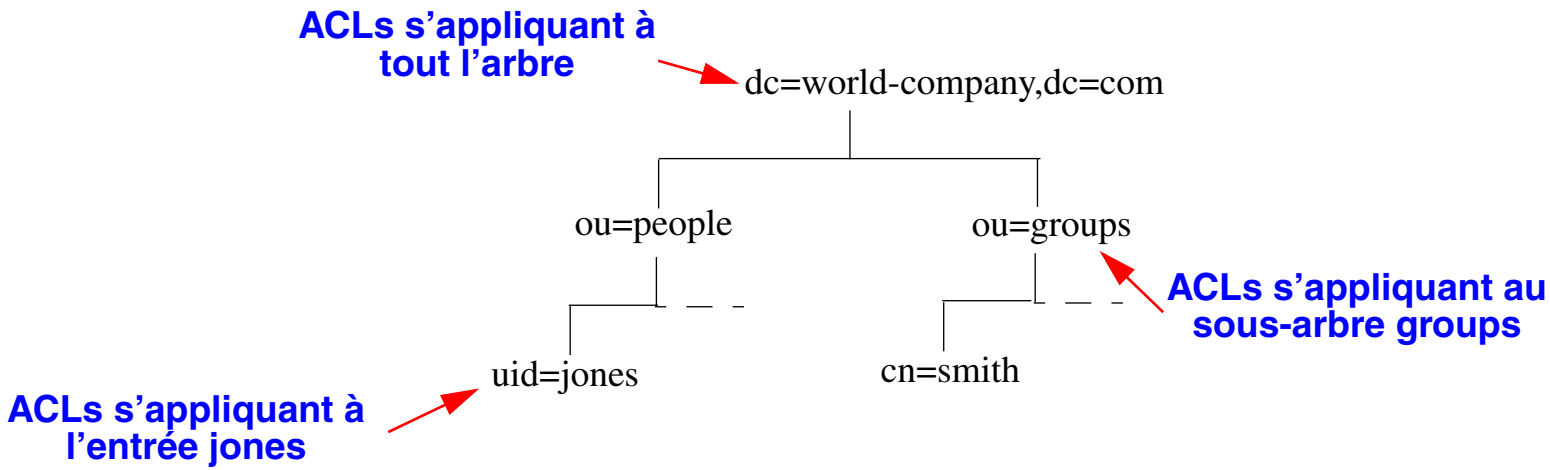
règle pour membre du groupe Servadm Manager

```
aci: (target="ldap:///dc=world-company,dc=com)
  (targetattr != "uid||mail")
  (targetfilter="(&(ou=servadm)(objectclass=person))")
  (version 3.0;acl "Servadm group permissions";
   allow (write)
   (groupdn="ldap:///cn=servadm manager,ou=groups,dc=world-company,dc=com");)
```

Déploiement : sécuriser le service

Mettre en place des règles de contrôle d'accès

Le placement des ACLs influe sur leur portée.



Déploiement : gestion des données

Etablir une méthode de gestion des données, selon la nature des attributs, pour déterminer qui administre quels attributs et dans quelle partie du DIT.

On distingues plusieurs catégories de gestion :

- attributs maintenus par l'administrateur de l'annuaire (contrôle d'accès...)
- attributs maintenus par les fournisseurs de données (service du personnel...)
- attributs maintenus par l'utilisateur final (photo, téléphone...)
- attributs maintenus par les applications (préférences...)
- attributs maintenus par le service d'annuaire lui-même

Pour chacune, il faut définir la méthode et la fréquence de mise à jour, évaluer la qualité des données et évaluer l'incidence sur les performances du serveur.

Déploiement : gestion des données

A terme, les attributs maintenus par les applications deviendront majoritaires.

Ces applications doivent respecter certaines règles pour optimiser les performances du serveur :

- minimiser les connexions en groupant les opérations
- optimiser le nombre d'opération : rechercher plusieurs attributs d'un coup, ne récupérer que les attributs nécessaires, faire des recherches efficaces
- minimiser les mises à jours

Déploiement : gestion des données

Les attributs maintenus de manière centralisées font l'objet des choix suivants :

- → mise à jour par commande ou par import de fichier
- → protéger les transactions de mises à jours de données sensibles
- → qui fait les mises à jour (personnes, scripts...)
- → quelle fréquence
- → vérifier les données en amont

Déploiement : gestion des données

Les attributs maintenus par l'utilisateur présentent les caractéristiques suivantes :

- source d'information, d'où des données plus à jour (bureau, téléphone...)
- implication des utilisateurs
- risque de saisies erronées ou invalides
- disposer d'une interface spécifique de mise à jour
- attention à la fréquence globale de mise à jour et son impact sur les performances

Logiciels serveurs

- Concepts
- Déployer un service LDAP

❑ **Les logiciels serveurs**

- Les clients LDAP
- Les outils de développement
- Les applications de LDAP aujourd'hui et demain
- Bibliographie

Logiciels serveurs

A cette date, les logiciels les plus connus sont :

- **OpenLDAP server,**
- **Innosoft's Distributed Directory Server,**
- **Netscape Directory Server,**
- **Sun Microsystems's Directory Services,**
- **IBM's DSSeries LDAP Directory,**
- **University of Michigan's SLAPD.**

D'autres annuaires supportent les requêtes au format LDAP :

- **Novell's NetWare Directory Services (NDS) 3.0,**
- **Microsoft's Active Directory (AD),**
- **Lotus Domino.**

Logiciels serveurs

Choisir un logiciel serveur : quelques critères de choix

- ⇨ le prix d'achat
- ⇨ les coût de maintenance et de support
- ⇨ l'adéquation du logiciel avec le type d'applications envisagées : détermine l'importance à accorder aux critères d'évaluations (performances, nombre d'entrées supportés, niveau de sécurisation...)
- ⇨ la facilité de prise en main
- ⇨ l'adéquation entre son choix de design et les fonctionnalités du logiciel (schéma, replication, referral...)
- ⇨ la compatibilité avec le logiciel antérieur (réutilisabilité)

Logiciels serveurs

Choisir un logiciel serveur : quelques critères d'évaluation

- → les fonctionnalités de base
  - **les plates-formes hardware/software supportées**
  - **le schéma et ses extensions**
  - **les opérations LDAP standards et étendues**
  - **les possibilités de duplication**
  - **le support de la distribution (referral, chaining)**
  - **outils d'import-export, de backup**
- → les outils de gestion
  - **procédure d'installation**
  - **outils de configuration et d'administration (interface web, commandes en ligne pour automatisation...)**
  - **interfaces de gestion de la base (clients natifs, web, commandes en ligne...)**
  - **possibilité d'administrer à distance**

Logiciels serveurs

- Choisir un logiciel serveur : quelques critères d'évaluation (suite)
  - → Les outils de développement
    - **API**
    - **SDK**
    - **logiciels clients**
  - → la fiabilité
    - **sauvegardes et modifications de configuration à chaud**
    - **mécanismes de *replication* multi-master**
    - **outils de monitoring**
    - **qualité de la base de données utilisée en cas d'arrêt intempestif**

Logiciels serveurs

- Choisir un logiciel serveur : quelques critères d'évaluation (suite)
  - → performance et évolutivité
    - **temps de latence**
    - **nombre d'opérations par seconde**
    - **nombre de connexions simultanées**
    - **nombre d'entrées, d'attributs et taille supportés**
    - **nombre de replicas et de partitions supportés**
    - **benchmark DirectoryMark** (`http://www.mindcraft.com/benchmarks/dirmark`)
  - → sécurité
    - **méthodes de contrôle d'accès**
    - **gestion des droits d'accès**
    - **méthodes d'authentification**
    - **chiffrement des transactions, de la duplication**

Logiciels serveurs

- Choisir un logiciel serveur : quelques critères d'évaluation (suite)
  - → conformité aux standards
    - **LDAPv2 core : RFC1777-1779**
    - **LDAPv3 core : RFC2251-2256**
    - **LDAPv3 extension**
    - **LDIF**
    - **API**
    - **SSL/TLS, certificats X509**
    - **schémas standards**
    - **standards X.500**
  - → interopérabilité

Le respect des standards est une première garantie d'interopérabilité

Choisir un logiciel serveur : évaluation

- → comparer les fonctionnalités
- → tester les softs sur une base pilote
- → faire quelques benchmarks

Logiciels clients

- Concepts
- Déployer un service LDAP
- Les logiciels serveurs

❑ **Les clients LDAP**

- Les outils de développement
- Les applications de LDAP aujourd'hui et demain
- Bibliographie

Clients LDAP

- **Accès natif :**
  - **Netscape Communicator**
  - **Microsoft Outlook, NetMeeting**
  - **Netscape SuiteSpot (les serveurs mail, news, web...)**
  - **Oblix (gestionnaire d'annuaire)**
  - **Navigateur Web : URLs LDAP**
  - **U-Mich xaX.500**
  - **GQ (GTK-based LDAP client)**
  - **LDAP Browser/Editor (Java-based LDAP client)**
  - **Applications développées avec un SDK LDAP**
- **Accès via passerelle :**
  - **LDAP vers X.500 et X.500 vers LDAP**
  - **HTTP vers LDAP (web500gw)**
  - **WHOIS++ vers LDAP**
  - **FINGER vers LDAP**
  - **PH/CSO vers LDAP**

Appels systèmes LDAP

- **Microsoft Windows NT**
  NT 5 utilise une base LDAP à la place des bases SAM
- **PADL software :**
  ypldapd : a gateway between NIS/YP and LDAP
  NSS LDAP : Nameservice switch library module
  PAM LDAP : Pluggable authentication module
- **Sun Solaris**
  NSS : Nameservice switch library module
- **Linux**
  Linux Directory Services : projet de remplacement de NIS par LDAP

Les outils de développement

- Concepts
- Déployer un service LDAP
- Les logiciels serveurs
- Les clients LDAP

❑ **Les outils de développement**

- → **Netscape C SDK**
- → **Netscape PerLDAP SDK**
- → **Netscape JAVA SDK**
- → **SUN JNDI**
- → **ADSI SDK**
- → **Netscape Directory Server Plug-Ins**
- → **Les autres...**

- Les applications de LDAP aujourd'hui et demain
- Bibliographie

Les outils de développement : Netscape C SDK

Connexion/Déconnexion

```
#include <stdio.h>
#include "ldap.h"
#define HOST "ldap.worldcompany.com"
#define PORT "389"

LDAP *ld;             /* LDAP Data Structure */
int rc;

if ( (ld = ldap_init("ldap.worldcompany.com",389)) == NULL) {
   rc = ldap_get_lderrno(ld,NULL,NULL);
   fprintf(stderr, "erreur %s\n",ldap_err2string(rc);
   return(rc);
}

[...]
if ( ldap_unbind(ld) != LDAP_SUCCESS) {
   rc = ldap_get_lderrno(ld,NULL,NULL);
   fprintf(stderr, "erreur %s\n",ldap_err2string(rc);
   return(rc);
}
```

Les outils de développement : Netscape C SDK

Bind/Unbind

```
[...]
#define DN NULL/* anonymous dn */
#define PW NULL/* anonymous dn */

[...connexion...]

if ( (rc = ldap_simple_bind_s(ld,DN,PW)) != LDAP_SUCCESS ) {
   fprintf(stderr, "erreur %s\n",ldap_err2string(rc);
}
else {
   printf("authentification réussie\n");
}

[...]

ldap_unbind(ld) ;
```

Les outils de développement : Netscape C SDK

Search

```
[...]
#define SEARCHBASE "ou=people,o=WorldCompany"
#define SCOPE LDAP_SCOPE_SUBTREE
#define FILTER "(uid=jsmith)"
LDAPMessage *result, *e;
BerElement *ber;
char *attribute, **vals;
[...connexion...]
/* recherche */
rc = ldap_search_ext_s(ld, SEARCHBASE, SCOPE, FILTER, NULL, 0, NULL,
                       NULL, LDAP_NO_LIMIT, 0, &result);
/* affichage */
for (e = ldap_first_entry(ld, result); e != NULL; e = ldap_next_entry(ld,e)) {
   printf("dn: %s\n", ldap_get_dn(ld, e));
   for (attribute = ldap_first_attribute(ld, e, &ber);
        attribute != NULL ; attribute = ldap_next_attribute(ld, e, ber)) {
      if ( (vals = ldap_get_values(ld, e, attribute) != NULL ) {
         for (i = 0; vals[i] != NULL; i++) {
            printf("%s: %s\n",attribute, vals[i]);
         }
      }
   }
}
```

Les outils de développement : Netscape C SDK

Add entry

```
[...]
#define DN "cn=john smith,ou=people,o=WorldCompany"
LDAPMod attribut1, attribut2, attribut3, mods[];
char objectclass_values[] = {"top", "person", "organizationalperson", NULL};
char cn_values[] = {"John Smith", NULL};
char ou_values[] = {"people", NULL};

[...connexion...]
/* création de l'entrée */
attribut1.mod_op = LDAP_MOD_ADD;
attribut1.mode_type = "cn";
attribut1.mode_value = cn_values;
attribut2.mod_op = LDAP_MOD_ADD;
attribut2.mode_type = "ou";
attribut2.mode_value = ou_values;
...
mods[0] = &attribute1 ;
mods[1] = &attribute2 ;

/* ajout de l'entrée */
rc = ldap_add_ext_s(ld, DN, mods, NULL, NULL);
[...deconnexion...]
```

Les outils de développement : Netscape C SDK

Delete entry

```
[...]
#define DN "cn=john smith,ou=people,o=WorldCompany"

[...connexion...]
[...authentification...]

/* destruction de l'entrée */
rc = ldap_delete_ext_s(ld, DN, NULL, NULL);

[...deconnexion...]
```

Les outils de développement : Netscape PerLDAP SDK

❑ Connexion/Déconnexion/Bind

```
use Mozilla::LDAP::Conn;
use Mozilla::LDAP::Utils;
my $ldap = "ldap.worldcompany.com";
my $port = "389";
my $base = "o=worldcompany";
# cas 1 : authentification anonyme
my $bind = "NULL";
my $passwd = "NULL";
# cas 2 : authentification utilisateur
my $bind = "uid=jsmith,ou=people,o=WorldCompany";
my $passwd = "toto";
# cas 3 : authentification administrateur
my $bind = "cn=Directroy Manager";
my $passwd = "le_chef";

# connexion et authentification
my $conn = new Mozilla::LDAP::Conn("$ldap","$port","$bind","$passwd","$cert");
die "Could't connect to LDAP server $ld{host}" unless $conn;
[...]
# déconnexion
$conn->close if $conn;
```

Les outils de développement : Netscape PerLDAP SDK

Search

```
use Mozilla::LDAP::Conn;
use Mozilla::LDAP::Entry;

my $base = "ou=people,o=WorldCompany";
my $scope = "subtree";
my $filter = "(&(objectclass=person)(ou=semir))" ;

[...connexion...]

# search
my $entry = $conn->search($base, $scope, $filter);
$conn->printError() if $conn->getErrorCode();
# affichage du résultat
if (! $entry ) {
   print "Recherche infructueuse.\n";
}
else {
   while ($entry) {
      $entry->printLDIF();
      $entry = $conn->nextEntry;
   }
}
```

Les outils de développement : Netscape PerLDAP SDK

Add entry

```
[...]

# DN de l'entrée
my $dn = "cn=john smith,ou=people,o=WorldCompany";
[...connexion...]
# construction de l'entrée
my $newentry = new Mozilla::LDAP::Entry();
$newentry->setDN($dn);
$newentry->{objectclass} = [ "top", "person", "organizationalPerson" ];
$newentry->{manager} = [ "uid=jsmith,ou=people,o=WorldCompany" ];
$newentry->{cn} = [ "John Smith" ];
$newentry->{description} = [ "indescriptible !" ];
$newentry->{ou} = [ "people" ];
$newentry->addValue("o", "french part of company");
$newentry->addValue("seeAlso", "http://www.worldcompany.com/~jsmith");

# ajout de l'entrée
$conn->add($newentry);
if ($conn->getErrorCode()) {
   print $conn->printError();
}
```

Les outils de développement : Netscape PerLDAP SDK

Delete entry

```
use Mozilla::LDAP::Conn;
use Mozilla::LDAP::Utils;
use Mozilla::LDAP::Entry;

# DN de l'entrée
my $dn = "cn=john smith,ou=people,o=WorldCompany";

[...connexion...]
# destruction de l'entrée
$conn->delete($dn);
if ($conn->getErrorCode()) {
   print $conn->printError();
}
else {
   print "Utilisateur supprimé.\n";
}
```

Les outils de développement : Netscape Java SDK

Connexion/Déconnexion/Bind

```
import netscape.ldap.*;
import java.io.*;
import java.util.*;

[...]
LDAPConnection ldap = new LDAPConnection();
/* connexion */
ldap.connect("ldap.worldcompany.com",389);

/* authentification anonyme */
ldap.authenticate("","");

/* authentification utilisateur */
ldap.authenticate("uid=jsmith,ou=people,o=WorldCompany","toto");

/* connexion et authentification en un coup */
ldap.connect("ldap.worldcompany.com",389,"uid=jsmith,ou=people,o=WorldCom-
pany","toto");

/* déconnexion */
ldap.disconnect();
```

Les outils de développement : Netscape Java SDK

Search

```
[...connexion...]
String base = "o=worldcompany";
int scope = LDAPConnection.SCOPE_SUB;
String filter = "(objectclass=person)";
[...]
/* search */
LDAPSearchResults res = ldap.seach(base,scope,filter,null,false);
/* affichage */
while (res.hasMoreElements()) {
   LDAPEntry findEntry = (LDAPEntry) res.next();
   System.out.println("dn: " + findEntry.getDN());
   LDAPAttributeSet attributeSet = findEntry.getAttributeSet();
   for (int i=0;i<attributeSet.size();i++) {
      LDAPAttribute attribute = (LDAPAttribute)attributSet.elementAt(i);
      String attrName = attribute.getName();
      System.out,println(attrName + " :");
      Enumeration enumVals = attribute.getStringValues();
      while (enumVals.hasMoreElements()) {
         String nextValue = (String)enumVals.nextElement();
         System.out,println(nextValue);
      }
   }
}
```

Les outils de développement : Netscape Java SDK

Add entry

```
[...connexion...]
String dn = "cn=john smith,ou=people,o=WorldCompany";
String objectclass_values[] = {"top", "person", "organizationalperson"};
String cn_values[] = {"John Smith"};
String ou_values[] = {"people"};
[...]
LDAPAttributeSet attrib_set = new LDAPAttributeSet();
LDAPAttribute attribute = null;
attribute = new LDAPAttribute("objectclass", objectclass_values);
attrib_set.add(attribute);attribute = new LDAPAttribute("cn", cn_values);
attrib_set.add(attribute);attribute = new LDAPAttribute("ou", ou_values);
attrib_set.add(attribute);

/* création de l'objet */
LDAPEntry entry = new LDAPEntry(dn,attrib_set);

/* ajout de l'entrée */
ld.add(entry);
[...]
```

Les outils de développement : Netscape Java SDK

Delete entry

```
[...connexion...]

String dn = "cn=john smith,ou=people,o=WorldCompany";
[...]

/* destruction de l'entrée */
ldap.delete(dn);

[...]
```

Les outils de développement : les autres

U-M LDAP SDK -- C (UMICH, OpenLDAP)

→ **le premier SDK**

Innosoft LDAP Client SDK (ILC-SDK) -- C (InnoSoft)

→ **proche du précédent**

Les outils de développement : les autres

LDAP Command Line Tools -- packages LDAP (U-M, OpenLDAP, Netscape)

→ ldapsearch

```
ldapsearch -h ldap.worldcompany.com -b "o=worldcompany" -s sub "&((object-
class=person)(ou=semir))" cn,uid,mail,telephonenumber
```

→ ldapmodify

```
ldapmodify -h ldap.worldcompany.com -b "o=worldcompany" -D "cn=Directory Manager"
-w "toto"
dn: uid=jsmith, ou=people, o=worldcompany
changetype: modify
replace: roomnumber
roomnumber: C105
-
add: description
description: newsmaster
-
delete: title
\n\n
```

Les outils de développement : les autres

Java naming and Directory Interface (JNDI) -- Java (SUN)

- → **conçu comme interface à différents protocoles de type annuaire (LDAP, Sun NIS/NIS+, Novell NDS...).**

Active Directory Service Interfaces - COM (Microsoft)

- → **concept similaire à JNDI.**

Net- LDAPapi -- PERL (GNU)

- → **comme PerLDAP mais entièrement en Perl.**

LDAP API to Python -- Python (University of Queensland)

- → **langage orienté développement d'interface graphique.**

LDAP API to PHP (`http://www.php.net`)

- → **langage de script orienté Web - server-side dynamic HTML.**

DSML -- Directory Service Markup Language (`http://www.dsml.org/`)

- → **standard pour représenter des informations issues de service d'annuaire en XML.**

Les outils de développement : les autres

PS Enlist - ODBC interface to LDAP (`http://www.pspl.co.in/PSEnList`)

→ **accès à LDAP via ODBC (i.e. accèder à LDAP depuis MS Office !).**

Server-Side Javascript LDAP SDK -- JavaScript (Netscape)

→ **module orienté Web - dynamic HTML pour les serveurs Web Netscape SuiteSpot.**

ColdFusion (Allaire)

→ **Langage/outil de développement orienté Web - database, s'interfaçant avec LDAP.**

Les applications de LDAP

- Concepts
- Déployer un service LDAP
- Les logiciels serveurs
- Les clients LDAP
- Les outils de développement

❑ **Les applications de LDAP aujourd'hui et demain**

- Bibliographie

Les applications de LDAP

- **Les différents domaines d'application possibles des annuaires LDAP :**
  - → Les applications système
  - → Les applications Intranet/Extranet
  - → Les applications Internet
  - → Les bases de données

Les applications de LDAP : applications systèmes

- **Les applications systèmes**

L'annuaire utilisé pour servir aux besoins des services réseaux tels que l'authentification, le contrôle d'accès, la localisation des imprimantes ou des serveurs de fichier.

Dans ce cas, il est étroitement lié au système d'exploitation.

De plus en plus de fabricants se tournent vers le standard LDAP pour l'implanter dans leur système.

Exemple : Windows 2000, Novell, Solaris, Linux...

Les applications de LDAP : applications intranet

❑ Les applications Intranet

Le service d'annuaire sert typiquement aux applications utiles à l'utilisateur final :

- **accès à des pages Web,**
- **annuaire téléphonique ou pour la messagerie électronique,**
- **profils de configuration... (Netscape suitespot, Lotus Domino...)**

Les applications de LDAP : applications extranet

- **Les applications Extranet**

L'annuaire peut servir de base d'information entre un fournisseur et ses sous-traitant, une banque et ses clients...

Ce sont celles mises en œuvre par les ISPs ou les grandes entités industrielles ou universitaires.

L'annuaire sert à gérer les abonnées, les hébergements de services comme le Web et la messagerie.

Les applications de LDAP : bases de données

❑ Les bases de données

L'annuaire peut remplacer un SGBD traditionnel dans le cas de données simples, intensivement interrogées, distribuées à large échelle et utilisées par des multiples applications (fichier clientèle, catalogues de fournitures...).

Il peut épauler un SGBD, en étant synchronisé avec lui, pour faciliter la consultation des données ou la mise à jour de certains champs.

Parfois, l'organisation possède plusieurs bases de données déconnectées et gérant des informations redondantes :

- **la paye**
- **le bureau du personnel**
- **les comptes informatiques**
- **les badges d'accès**
- **les cartes de restaurants...**

Un annuaire LDAP peut fédérer les données communes (informations sur les employés), les données sensibles étant gérées dans les SGBD => Meta-Directory.

Les applications de LDAP : exemples

- **Gestion centralisée de l'authentification et des droits d'accès**
  - → **Remplacer les multiples mots de passe applicatifs/systèmes par une authentification LDAP centralisée.**

Netscape Directory Server - synchronisation des bases utilisateurs Windows NT4 avec base LDAP

Netscape SuiteSpot - serveur de Mail, de News, Web utilisant LDAP pour l'authentification

Cyrus IMAP/POP3' pwcheck_ldap.c - programme externe d'authentification LDAP pour les serveurs IMAP/POP3 de Cyrus.

Apache::AuthLDAP - module d'authentification et de gestion des autorisations d'accès au serveur Web Apache via LDAP.

PADL Software's PAM (Pluggable Authentification Module) & NSS (Name Service Switch) Modules - authentification/lookup redirigés sur LDAP sous Solaris et Linux

Les applications de LDAP : exemples

- **Gestion des mailing-lists et des aliases mail par LDAP**

Netscape Messenger Server - Serveur de Mail « full LDAP ».
Sendmail 8.9.x : peut utiliser LDAP pour les résolutions d'adresses.
Sympa : gestionnaire de listes de diffusions « LDAP capable »

Les applications de LDAP : exemples

- **Mobilité utilisateur : accès distant des applications aux options, configurations et préférences**
  - → **permettre à l'utilisateur de retrouver son environnement applicatif indépendamment de sa localisation**

Netscape Communicator Roaming Access.

Netscape Calendar nscalUser object class.

Les applications de LDAP : exemples

- **Annuaires...**

Annuaire du personnel

Inventaire du matériel

Stockage des certificats (X509) et des listes de révocation (CLRs) - pour des infrastructures à base de clefs publiques -

Les applications de LDAP : exemples

- **Directory Enabled Networks Initiative (DEN)**
  - → **Consortium pour définir un modèle d'information standard facilitant le développement d'applications reseaux « Directory-Enabled » interopérables.**
  - → **Faciliter l'accès des utilisateurs aux services réseaux : authentification, droits d'accès...**

Futur

- apparition des méta-annuaires
- intégration des annuaires dans les OS
- tendance à utiliser LDAP comme un protocol léger d'accès à des bases de données
- prédominance de LDAP
- des annuaires partout !

Bibliographie

- Linuxworld LDAP in action:

  http://linuxworld.com/linuxworld/lw-1999-07/lw-07-ldap_1.html
- Linux LDAP services:

  http://www.rage.net/ldap/
- OPenLDAP.org:

  http://www.openldap.org
- Netscape Deployment Guide:

  http://developer.netscape.com/docs/manuals/directory/deploy30/index.htm
- LDAP FAQ:

  http://www.critical-angle.com/ldapworld/ldapfaq.html
- LDAP roadmap and FAQ:

  http://www.kingsmountain.com/ldapRoadmap.shtml
- LDAP Central

  http://www.ldapcentral.com/
- Understanding and deploying LDAP directory services, T. Howes, M. C. Smith, G. Good; Macmillan