LDAP & OpenLDAP
Par la pratique…

Master Sciences & Technologie
Mention Mathématiques et Sciences Pour l'Ingénieur
Spécialité Informatique
Parcours Ingénierie du Logiciel Libre
2ème Année

Année 2009-2010

D'après la version de Jean-Christophe Soulié (2007-2008)
D. Duvivier
LIL – Université du Littoral Côte d'Opale
duvivier@lil.univ-littoral.fr

Plan

- LDAP
- Déployer LDAP
- OpenLDAP

LDAP

LDAP ? Qu’est-ce que c’est ?

- LDAP est le protocole d'annuaire sur TCP/IP
- Les annuaires permettent de partager des bases d'informations sur le réseau interne ou externe
- Ces bases peuvent contenir toute sorte d'information que ce soit des coordonnées de personnes ou des données système

Qu'est-ce qu'un annuaire ?

- On utilise fréquemment des annuaires, l'annuaire téléphonique par exemple
  - Cet annuaire regroupe différentes entrées contenant chacune des informations particulières
    - Nom,
    - Prénom,
    - Numéro de téléphone,
    - Adresse,
    - Etc.

Qu'est-ce qu'un annuaire ?

- Caractéristiques communes aux annuaires :
  - Un annuaire présente un ensemble défini de données (annuaire : nom, prénom, numéro de téléphone, adresse)
  - Il organise ces données (annuaire : classées par département, villes, nom)
  - Il offre un service de consultation (annuaire : diffusion au format papier)
  - Il peut protéger les données (annuaire : liste rouge)
  - Il est plus consulté que mis à jour
  - Il est disponible de manière permanente

Différences avec un Système de Gestion de Bases de Données ?

- Sur un annuaire, les écritures sont plus rares que les lectures, ce qui n'est pas forcément le cas pour un SGBD
  - Un annuaire n'est pas fait pour stocker des informations constamment en mouvement
- Un annuaire fournit une méthode de consultation standardisée, ce qui n'est pas le cas d'un SGBD
  - SQL est, certes, standardisé, …
  - Mais tous les SGDB en ont une implémentation différente (MySQL, PostgreSQL, Oracle avec SQL+, etc.)

Différences avec un Système de Gestion de Bases de Données ?

- Un annuaire LDAP organise les données de manière arborescente, tandis que les bases de données le font au sein de tableaux à deux dimensions
- Un annuaire fournit des modèles de données standardisés
  - Le modèle conceptuel de données (que stocker, où et comment ?) d'un SGBD peut varier d'une entreprise et d'une base à une autre

Différences avec un Système de Gestion de Bases de Données ?

- Un annuaire fournit des modèles de données officialisés (par le biais des schémas)
  - Une capacité d'interopérabilité sans commune mesure

Historique

- En 1988, l'Union Internationale des Communications (UIT) met au point les annuaires X.500
  - But : uniformiser l'accès aux services, de centraliser les ressources et de les protéger
- Le protocole utilisé pour y accéder est le protocole DAP (Directory Access Protocol)
- Malheureusement, le protocole DAP s'avère difficile à mettre en œuvre et ne fonctionne pas sur les réseaux TCP/IP

Historique

- En 1993, l'Université du Michigan réfléchit donc à un moyen de palier ces deux problèmes
  - Elle met en place le protocole LDAP (Lightweight Directory Access Protocol)
  - Au départ simple « connecteur » TCP/IP avec des annuaires X.500
- En 1995, LDAP devient un protocole natif et utilisable indépendamment de X.500

Historique

- LDAP est donc une évolution de la norme X.500
- Sa version actuelle est la version 3 (RFC 2251, http://www.ietf.org/rfc/rfc2251.txt ;
  voir également les RFC plus récentes sur http://www.ietf.org)
  - Le support des communications chiffrées via SSL/TLS
  - L'authentification via SASL
  - Le support des Referrals (une branche pointe vers un autre annuaire)
  - Le support d'Unicode (internationalisation)
  - La capacité d'étendre le protocole
  - Le support des schémas dans l'annuaire

D'autres types d'annuaires…

- D'autres types d'annuaires existent,
  - DNS : Domain Name Services
  - NIS : Network Information Services
  - Whois : base d'information concernant les noms de domaines
  - …

Quelques annuaires LDAP

- OpenLDAP :
  - http://www.openldap.org
- Apache Directory Server :
  - http://directory.apache.org
- Sun (One/Java) Directory Server :
  - http://www.sun.com

    (http://www.sun.com/software/products/directory_srvr_ee/)
- Active Directory :
  - http://www.microsoft.com

    (cherchez “active directory ldap” sur le site par exemple)
- [...]

Les concepts de LDAP

- LDAP est un protocole d'annuaire standard et extensible. Il fournit :
  - Le ***protocole*** permettant d'accéder à l'information contenue dans l'annuaire,
  - Un ***modèle d'information*** définissant le type de données contenues dans l'annuaire,
  - Un ***modèle de nommage*** définissant comment l'information est organisée et référencée,
  - Un ***modèle fonctionnel*** qui définit comment on accède à l'information ,
  - Un ***modèle de sécurité*** qui définit comment données et accès sont protégés,
  - Un ***modèle de duplication*** qui définit comment la base est répartie entre serveurs,
  - Des ***APIs*** pour développer des applications clientes,
  - ***LDIF***, un format d'échange de données.

Le protocole LDAP

- Le protocole définit comment s'établit la communication *client-serveur*
- Il fournit à l'utilisateur des commandes pour
  - Se connecter,
  - Se déconnecter,
  - Pour rechercher, comparer, créer, modifier ou effacer des entrées

Le protocole LDAP

- Les transactions et l'accès aux données grâce à
  - Des mécanismes de chiffrement (SSL ou TLS) et d'authentification (SASL),
  - Couplés à des mécanismes de règles d'accès (ACL)

Le protocole LDAP

- La plupart des logiciels serveurs LDAP proposent également un protocole de communication *serveur-serveur* permettant à plusieurs serveurs d'échanger leur contenu et de le synchroniser (*replication service*)
- Ou de créer entre eux des liens permettant ainsi de relier des annuaires les uns aux autres (*referral service*)

Le protocole LDAP

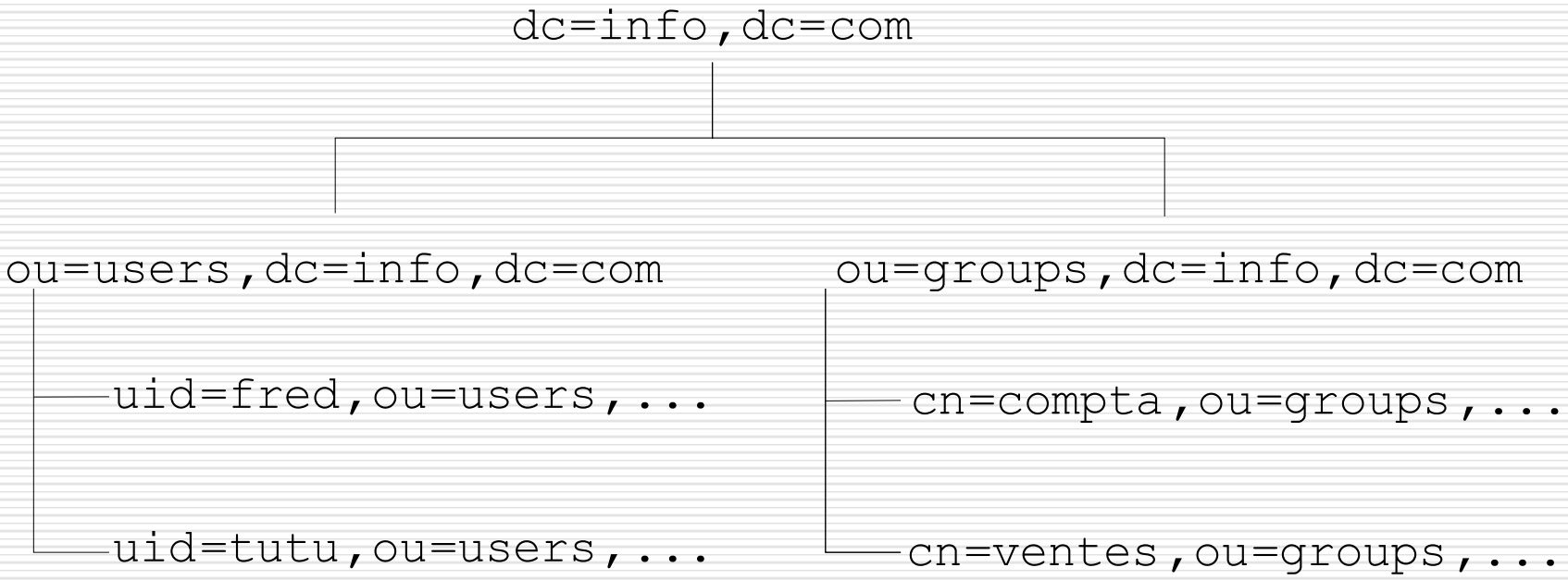
- La communication client-serveur est normalisée par l'Internet Engineering Task Force (IETF)
- Contrairement à d'autres protocoles d'Internet, comme HTTP, SMTP ou NNTP, le *dialogue* LDAP ne se fait pas en ASCII mais utilise le format de codage *Basic Encoding Rule* (BER)

Organisation des données (modèle de nommage)

- Le modèle de nommage est la manière dont sont organisées les données dans l'annuaire
- LDAP organise les données de manière hiérarchique dans l'annuaire
- Ceci signifie que toutes les informations découlent d'une seule et même « racine »

Organisation des données (modèle de nommage)

- Exemple :



Organisation des données (modèle de nommage)

- Cette arborescence est liée au nommage de chaque élément : un élément marque son appartenance à l'élément supérieur en en reprenant le nom, qu'il complète par le sien

- Ainsi, en étudiant simplement le nom de l'élément : **« cn=ventes,ou=groups,dc=info,dc=com »**

- Il est aussi possible de le situer dans la hiérarchie : il est situé sous l'élément « **ou=groups** » qui lui-même est situé sous l'élément « **dc=info,dc=com »**

Organisation des données (modèle de nommage)

- Termes à connaître :
  - Chaque élément est appelé une **entrée** (an entry). Une entrée peut être un branchement (un **noeud**, a node) ou un élément terminal (une **feuille**, a leaf)
  - Chaque élément possède un **DN** (Distinguished Name). Le DN est le nom complet de l'élément qui permet de le positionner dans l'arborescence. Il est unique dans l'annuaire
    - Exemple :

      « **cn=ventes,ou=groups,dc=info,dc=com** »

Organisation des données (modèle de nommage)

- Termes à connaître :
  - Chaque élément possède également un **RDN** (Relative Distinguished Name). Le RDN est la partie du **DN** de l'élément qui est relative au **DN** supérieur. Le RDN d'un élément ne
  - permet pas de l'identifier de manière absolue dans l'annuaire
    - Exemple : « **`cn=ventes`** »
  - La **racine** est l'élément supérieur de tous les autres, c'est la base de l'arborescence. On l'appelle **root** en anglais, parfois on parle de « **root DN** »
    - Exemple : « **`dc=info,dc=com`** »

Organisation des données (modèle de nommage)

- Termes à connaître :
  - Les DN de chaque entrées sont composés au moins d'un attribut de l'élément (par exemple « **cn** » ou « **uid** ») et de sa valeur. Un attribut est l'une des caractéristiques de cet élément
  - Dans ce cas la racine choisie ici est composée du nom du domaine où est hébergé le serveur LDAP, « **info.com** », décomposé en « **dc** » (Domain Components) pour obtenir « **dc=info,dc=com** »
  - L'arbre se découpe ensuite en deux « **ou** » (Organisational Units) qui constituent deux branchements : « **users** » et « **groups** », dans lesquels on trouve ensuite les entrées feuilles de l'arbre : les utilisateurs et les groupes
  - Chacune des entrées de l'arbre correspond à un type de donnée particulier, défini par une classe d'objet

Organisation des données (modèle de nommage)

- Règles de nommage :
  - La RFC 2253 normalise l'écriture des DN et conseille de ne pas ajouter d'espaces autour du signe « = », ni à la fin du DN
  - Les espaces sont autorisés par contre pour les valeurs des entrées
  - Ainsi, le DN suivant est correct :
    - **"cn=Toto Dupont,cn=ventes,ou=groups,dc=info,dc=com"**
  - Alors que celui-ci ne l'est pas :
    - **"cn = Toto Dupont, cn = ventes, ou = groups, dc = info, dc = com"**
  - Les majuscules seront ou non prises en compte en fonction du type d'attribut utilisé et de ses particularités

Accéder à l'annuaire (modèle fonctionnel)

- Il existe plusieurs types d'opérations que l'on peut effectuer sur l'annuaire, on peut citer les plus importantes :
  - Rechercher une entrée suivant certains critères
  - S'authentifier
  - Ajouter une entrée
  - Supprimer une entrée
  - (Modifier une entrée)
  - Renommer une entrée

Accéder à l'annuaire (modèle fonctionnel)

- La base :
  - La base est le DN à partir duquel nous allons agir
  - Pour une recherche, il s'agir du nœud à partir duquel est effectuée la recherche
  - Il peut s'agir de la racine de l'arbre pour une recherche sur la totalité de l'arbre, par exemple "**`dc=info,dc=com`**".

Accéder à l'annuaire (modèle fonctionnel)

- La portée
  - La portée (scope) est le nombre de niveaux sur lesquels l'action va être effectuée. Il existe 3 niveaux différents :
    - **SUB** : l'action est effectuée récursivement à partir de la base spécifiée sur la totalité de l'arborescence.
    - **ONE** : l'action est effectuée sur un seul niveau inférieur par rapport à la base spécifiée (les fils directs). Si on effecte une recherche avec la portée ONE à partir de "**dc=info,dc=com**", on pourrait trouver "**ou=users,dc=info,dc=com**" et "**ou=groups,dc=info,dc=com**".
    - **BASE** : l'action est effectuée uniquement sur la base spécifiée. Une recherche sur "**dc=info,dc=com**" avec la portée BASE renverrait cette entrée uniquement.

Accéder à l'annuaire (modèle fonctionnel)

- Les filtres :
  - Un filtre va permettre d'effectuer des tests de correspondance lors d'une recherche. Il s'agit, en quelque sorte, du critère de la recherche
  - Il existe 4 tests basiques, qui peuvent ensuite être combinés :
    - Le test d'égalité : **X=Y**
    - Le test d'infériorité : **X<=Y**
    - Le test de supériorité : **X>=Y**
    - Le test d'approximation : **X~=Y**

Accéder à l'annuaire (modèle fonctionnel)

- Les filtres :
  - Les autres opérateurs (**<**, **>**) ou des tests plus complexes peuvent être mis en place par combinaison, il faut alors utiliser les parenthèses **( )** et l'un des opérateurs suivants :
    - L'intersection (et) : **&**
    - L'union (ou) : **|**
    - La négation (non) : **!**
  - Un test d'infériorité stricte pourrait donner ceci :

(&(X<=Y)(!(X=Y)))

Accéder à l'annuaire (modèle fonctionnel)

- Les filtres :
  - On peut combiner plus de deux éléments :

```
(&(X=Y)(Y=Z)(A=B)(B=C)(!(C=D)))
```

  - Ces filtres seront appliqués sur des attributs choisis pour sélectionner finement les données que l'on veut extraire de notre annuaire

Accéder à l'annuaire (modèle fonctionnel)

- Opérateurs de recherche

Filtre	Syntaxe	Interprétation
Approximation	(sn~=Toto)	nom dont l'orthographe est voisine de Toto
Égalité	(sn=Toto)	vaut exactement Toto
Comparaison	(sn>Toto) , <= , >= , <	noms situés alphabétiquement après Toto
Présence	(sn=*)	toutes les entrées ayant un attribut sn
Sous-chaîne	(sn=Mir*), (sn=*irtai*), (sn=Mirt*i*)	expressions régulières sur les chaînes
ET	(&(sn=Toto) (ou=Semir))	toutes les entrées dont le nom est Toto et du service Semir
OU	(|(ou=Direction) (ou=Semir))	toutes les entrées dont le service est le Semir ou la Direction
Négation	(!(tel=*))	toutes les entrées sans attribut téléphone

Accéder à l'annuaire (modèle fonctionnel)

- Opérations de base :

Opération	**Description LDAP**
Search	Recherche dans l'annuaire d'objets à partir de critères
Compare	Comparaison du contenu de deux objets
Add	Ajout d'une entrée
Modify	Modification du contenu d'une entrée
Delete	Suppression d'un objet
Rename (Modify DN)	Modification du DN d'une entrée
Bind	Connexion au serveur
Unbind	Déconnexion
Abandon	Abandon d'une opération en cours
Extended	Opérations étendues (v3)

Accéder à l'annuaire (modèle fonctionnel)

- Paramètres d'une requête

Paramètre	**Description**
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 maximum 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

Accéder à l'annuaire (modèle fonctionnel)

- Les URLs LDAP
  - Récemment est apparue une méthode concise et simplifiée pour interroger un annuaire LDAP
  - Il s'agit d'un format d'URL combinant toutes les notions que vues précédemment
  - En une seule ligne, il est possible de spécifier tous les éléments d’une requête

Accéder à l'annuaire (modèle fonctionnel)

- Les URLs LDAP (http://www.ietf.org/rfc/rfc2255.txt ;
  voir également les RFC plus récentes sur http://www.ietf.org)
  - Format de cette URL (RFC 2255) :

```
ldap[s]://serveur[:port][/[base[?[attributs à
 afficher][?[portée][?[filtre][?extensions]]]]]
```

  - L'exemple ci-dessous recherche tous les **uid** de l'arbre, à partir de la branche **users** :

```
ldap://localhost:389/ou=users,dc=info,dc=com?
   uid?sub
```

Les données contenues dans l'annuaire (modèle d'information)

- Les attributs
  - Un attribut est une valeur contenue dans une entrée
  - Une entrée peut, bien entendu, contenir plusieurs attributs
  - Les attributs sont caractérisés 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
    - Un indicateur d'usage
    - Un format ou une limite de taille de valeur qui lui est associée

Les données contenues dans l'annuaire (modèle d'information)

- Les attributs
  - Certains serveurs LDAP respectent les standards X500 de hiérarchisation des attributs, qui permettent de décrire un attribut comme étant un *sous-type* d'un attribut *super-type* et d'hériter ainsi de ses caractéristiques
  - Par exemple, les attributs *cn*, *sn*, *givenname* sont des *sous-types* de l'attribut *super-type name*
  - Ces attributs *super-types* peuvent être utilisés comme critère de recherche générique qui porte sur tous ses *sous* attributs

Les données contenues dans l'annuaire (modèle d'information)

```
dn: uid=toto,ou=users,dc=info,dc=com
objectClass: account
objectClass: posixAccount
cn: toto
uid: toto
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/toto
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: info
description: info
```

Les données contenues dans l'annuaire (modèle d'information)

- Ceci correspond à une entrée complète, extraite par une interrogation de l'annuaire
- Le format affiché est le format **LDIF** (on en reparlera plus tard)
- Ce paragraphe présente tous les attributs, un par ligne, que comprend notre entrée
- Un attribut est séparé de sa valeur par « : »
- Suivant son type, un attribut peut avoir plusieurs valeurs : dans ce cas, il est dit « multi-valué » et apparaît sur plusieurs lignes avec des valeurs différentes

Les données contenues dans l'annuaire (modèle d'information)

- On peut observer ici des attributs nommés « **dn** », « **objectClass** », « **cn** », « **uid** », **…**
- L'attribut « **dn** » qui est indiqué en première ligne est le nom unique de l'entrée dans l'arbre dont on a parlé précédemment
- Il constitue un attribut à part entière dans l'entrée
- Il est composé du **dn** de l'entrée supérieure, ainsi que du **rdn**

Les données contenues dans l'annuaire (modèle d'information)

- Sur un annuaire LDAP la racine est toujours composée des attributs « **dc** » (Domain Component) associés à chacune des parties du nom de domaine où est hébergé le serveur ("**dc=info,dc=com**" pour le domaine info.com). Ceci est une convention
- X500 préconisait les attributs "o", "l" et "c", mais LDAP a simplifié le procédé (cf. RFC 2247)
- L'attribut « ou » constitue une « Organisational Unit », c'est à dire une unité organisationnelle : en quelque sorte un regroupement
- On a choisi d'en créer deux dans notre exemple : « **users** », qui accueillera nos utilisateurs et « **groups** », nos groupes

Les données contenues dans l'annuaire (modèle d'information)

- Les 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)

Les données contenues dans l'annuaire (modèle d'information)

- Les classes d'objets
  - Le type d'une classe est lié à la nature des attributs qu'elle utilise
    - Une *classe structurelle* correspond à la description d'objets basiques de l'annuaire : les *personnes*, les *groupes*, les *unités organisationnelles*... Une entrée appartient toujours au moins à une classe d'objet structurelle
    - Une *classe auxiliaire* désigne des objets qui permettent de rajouter des informations complémentaires à des objets structurels. Par exemple l'objet *mailRecipient* rajoute les attributs concernant la messagerie électronique d'une personne. L'objet *labeledURIObject* fait de même concernant les infos Web
    - Une *classe abstraite* désigne des objets basiques de LDAP comme les objets *top* ou *alias*

Les données contenues dans l'annuaire (modèle d'information)

- Les classes d’objets
  - 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 peut donc enrichir un objet en créant un objet fils qui lui rajoute des attributs supplémentaires

Les données contenues dans l'annuaire (modèle d'information)

- Les classes d’objets
  - 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

Les données contenues dans l'annuaire (modèle d'information)

- Les classes d'objets
  - Par exemple, l'objet *inetOrgPerson* a la filiation suivante :

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

(voir http://www.faqs.org/rfcs/rfc2798.html)

Les données contenues dans l'annuaire (modèle d'information)

- Les classes d'objets
  - 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`**, …

Type d'entrée	Attributs requis	Attributs optionnels
inetOrgPerson (définit les entrées pour une personne)	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

Les données contenues dans l'annuaire (modèle d'information)

Type d'entrée	Attributs requis	Attributs optionnels
organizationalUnit	ou objectClass	businessCategory description facsimileTelephoneNumber location (l) postalAddress seeAlso telephoneNumber
organization	o objectClass	businessCategory description facsimileTelephoneNumber location (l) postalAddress seeAlso telephoneNumber

Les données contenues dans l'annuaire (modèle d'information)

- Type des format des attributs

Type	Description
bin	binary information
ces	case exact string (case of text is significant during comparison)
cis	case ignore string (case of text is ignored during comparison)
tel	telephone number (numbers are treated as text, but all blanks and dashes (-) are ignored)
dn	distinguished name.

Les données contenues dans l'annuaire (modèle d'information)

- Les schémas
  - Comment savoir quels sont les **`objectClass`** disponibles et quels attributs ils contiennent ?
  - C'est très simple, la syntaxe et la liste des attributs connus de l'annuaire sont écrits dans ce que l'on appelle les « schémas »
  - Un annuaire LDAP a la capacité de charger en mémoire plusieurs schémas. A travers ces schémas, il est possible de définir de nouveaux attributs et de nouveaux **`objectClass`**
  - Cette souplesse permet de définir très finement ce qui sera stocké dans notre annuaire
  - Concrètement, un schéma est un fichier qui décrit un à un les attributs disponibles (leur nom, leur type, etc...), ainsi que les **`objectClass`** qui y font appel

Les données contenues dans l'annuaire (modèle d'information)

- Avant d'aller plus loin, installons OpenLdap :

- ***# aptitude install slapd ldap-utils***

  - Indiquez « toto » comme mot de passe pour les tests « habituels » en TP

Les données contenues dans l'annuaire (modèle d'information)

- Les schémas
  - Au démarrage du serveur LDAP, le ou les fichiers de schéma spécifiés dans sa configuration seront chargés
  - Dans notre exemple, l'**`objectClass posixAccount`** est défini dans le fichier **nis.schema**
  - Voyons une partie de ce fichier, livré avec OpenLDAP et situé dans le répertoire **/etc/ldap/schema** :

Les données contenues dans l'annuaire (modèle d'information)

```
# [...]
attributetype ( 1.3.6.1.1.1.1.0 NAME 'uidNumber'
DESC 'An integer uniquely identifying a user in a domain'
EQUALITY integerMatch
SYNTAX 1.3.6.1.4.1.1466.115.121.1.27 SINGLE-VALUE )
# [...]
objectclass ( 1.3.6.1.1.1.2.0 NAME 'posixAccount' SUP top AUXILIARY
DESC 'Abstraction of an account with POSIX attributes'
MUST ( cn $ uid $ uidNumber $ gidNumber $ homeDirectory )
MAY ( userPassword $ loginShell $ gecos $ description ) )
# [...]
```

Les données contenues dans l'annuaire (modèle d'information)

- Les schémas :
  - Le fichier est assez volumineux et a été tronqué
  - Le premier paragraphe définit l'un des attributs utilisés par le **`posixAccount`** : **`uidNumber`**
  - Le second, l'**`objectClass posixAccount`**
  - Nous n'allons pas étudier en détail ces deux définitions, simplement, sachez que :
    - A chaque définition correspond un **`OID`** (**`Object IDentifier`**), qui permet de rendre unique l'attribut spécifié. Ces **`OIDs`** sont déposés auprès de l'IANA (**http://www.iana.org**) et sont donc officiels.
    - Un attribut définit un type d'égalité à mettre en œuvre lors d'une recherche (ici, **`intergerMatch`**) ainsi que le type de données qu'il contient (l'**`OID`** spécifié après **`SYNTAX`**).
    - Un **`objectClass`** définit les attributs que l'objet **doit** présenter (**`MUST`**) et ceux qu'il **peut** posséder (**`MAY`**).

Les données contenues dans l'annuaire (modèle d'information)

- Les schémas :
  - Les schémas constituent donc une source d'information très importante. En cas de doute concernant le type ou le nom des attributs à spécifier dans une entrée, n'hésitez pas à vous y reporter !
  - Enfin, sachez qu'il est tout à fait possible de créer ses propres schémas, cependant, penser à réutiliser les schémas existants : ils offrent déjà de nombreuses possibilités et il y a fort à parier qu'un schéma existe déjà pour gérer les informations que vous souhaitez !

Les données contenues dans l'annuaire (modèle d'information)

- Le format LDIF
  - Les données contenues dans l'annuaire sont présentées dans un certain format : il s'agit du format LDIF (LDAP Data Interchange Format - RFC 2849, voir http://www.ietf.org/rfc/rfc2849.txt)
  - Toute interaction avec un annuaire se fait par le biais de ce format : l'ajout, la modification, la suppression d'entrées, l'interrogation de l'annuaire y compris
  - Dans ce format, chaque entrée constitue un paragraphe, et, au sein de chaque paragraphe, chaque ligne constitue un attribut
  - Voyons un exemple un peu plus complet, incluant le groupe de notre utilisateur :

Les données contenues dans l'annuaire (modèle d'information)

```
# [...]
dn: cn=utilisateurs,ou=groups,dc=info,dc=com
objectClass: posixGroup
cn: utilisateurs
gidNumber: 10001
dn: uid=toto,ou=users,dc=info,dc=com
objectClass: account
objectClass: posixAccount
cn: toto
uid: toto
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/toto
userPassword:: e0NSWVBUfWJjT29IUk5SbG1HbC4=
loginShell: /bin/sh
gecos: info
description: info
# [...]
```

La sécurité (modèle de sécurité)

- L'annuaire met en place un mécanisme d'authentification : pour avoir accès aux données qu'il contient, il faut être autorisé !
- L'authentification simple, le binding
  - L'une des opérations préalables à l'interrogation de l'annuaire est cette opération dite de « binding » (dans le cas d'une authentification simple)
  - Le client envoie alors le DN d'un compte contenu dans l'annuaire lui-même, ainsi que le mot de passe associé
  - On pourra par la suite appliquer des droits particuliers sur ce compte en utilisant les ACLs
  - Ceci correspond, si l'on fait le parallèle avec l'annuaire téléphonique, à la fonctionnalité de liste rouge, où certaines données ne sont pas accessibles à tout le monde
  - Notez enfin qu'il est possible de se connecter de manière anonyme : le client envoie alors un DN vide au serveur LDAP.

La sécurité (modèle de sécurité)

- Les ACLs
  - Les ACLs (Access Control Lists) interviennent après la notion de binding
  - Il sera possible de donner des droits de lecture, d'écriture (ou d'autres droits divers) sur des branches particulières de l'annuaire au compte connecté

La sécurité (modèle de sécurité)

- Le chiffrement des communications (SSL/TLS)
  - Le chiffrement des communications, via SSL (Secure Socket Layer, ou TLS - Transport Layer
  - Security) est également une méthode de protection de l'information
  - Il est possible, avec la plupart des annuaires existants, de chiffrer le canal de communication entre l'application cliente et l'annuaire
  - Ceci permet de garantir (un minimum) la confidentialité des données et d'éviter qu'un tiers n'écoute les communications sur le réseau

La sécurité (modèle de sécurité)

- SASL
  - SASL (Simple Authentication and Security Layer) est un mécanisme qui permet d'ajouter des méthodes d'authentification à des protocoles orientés connexion tels que LDAP ou IMAP
  - Il est défini dans la RFC 2222 ; l'implémentation la plus couramment utilisée est Cyrus-Sasl (http://asg.web.cmu.edu/sasl)

La sécurité (modèle de sécurité)

- SASL
  - SASL donne la possibilité au client et au serveur de sélectionner quelle sera la méthode d'authentification utilisée
  - Ces méthodes sont extensibles via des plugins
  - Il permet également de mettre en place une couche de connexion sécurisée telle que SSL/TLS (sans rapport direct avec le chiffrement indépendant des connexions que nous avons cité ci-dessus)

Concepts avancés

- La réplication
  - Certains serveurs LDAP, dont OpenLDAP, permettent de manière native, de mettre en place un annuaire répliqué
  - Un annuaire dit « maître » envoie alors, par le biais du format LDIF, toutes les modifications effectuées sur un annuaire « esclave »

Concepts avancés

- La réplication
  - L'avantage d'une telle opération est double :
    - Permettre une meilleure montée en charge pour de gros annuaires : il est possible de rediriger le client vers l'un ou l'autre des annuaires répliqués
    - Disposer d'une copie conforme du premier annuaire, utile en cas de crash (attention, toute opération est reportée de l'annuaire maître vers l'esclave, donc ceci est non valable en cas de mauvaise manipulation)

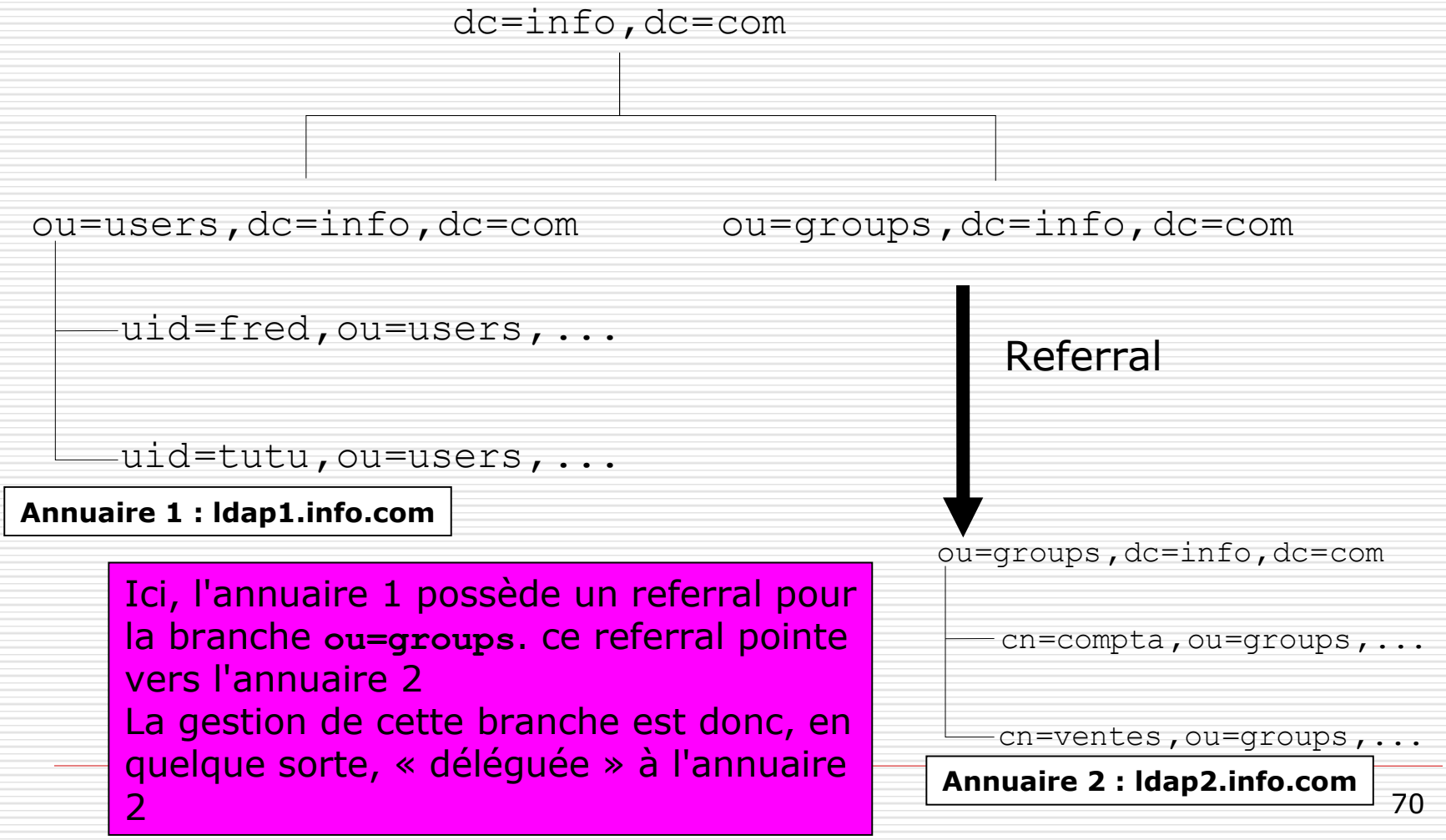
Concepts avancés

- La réplication
  - Deux types de réplication existent :
    - le mode « maître-esclave », le plus courant : la réplication est unidirectionnelle, un annuaire maître envoie toutes les modifications à un annuaire esclave. Ceci n'autorise bien évidemment l'écriture que sur l'annuaire maître ; l'esclave est alors disponible uniquement en lecture.
    - le mode « maître-maître » : la réplication est bidirectionnelle, chaque annuaire peut être maître de l'autre. Ceci permet d'écrire indifféremment sur l'un ou l'autre des annuaires.
  - Enfin, il est possible de chaîner les réplications pour obtenir plusieurs réplicats.

Concepts avancés

- La distribution (les referrals)
  - La distribution est un mécanisme qui va permettre de faire pointer un lien vers un autre annuaire pour une branche particulière
  - Ceci va permettre de déléguer la gestion de cette branche, un peu au sens DNS lorsqu'on délègue la gestion d'un domaine

Concepts avancés



Concepts avancés

- La distribution (les referrals)
  - Au niveau de l'annuaire 1, ceci se traduit par une entrée de la classe « **referral** », qui contient alors un attribut « **ref** » contenant l'adresse de la suite de l'arborescence :

```
dn: ou=groups,dc=info,dc=com
objectClass: referral
ref: ldap://ldap2.info.com/ou=groups,dc=info,dc=com
```

  - Il existe également les « **alias** » qui sont des liens symboliques au sein du même annuaire
  - Cf. l'objectClass « **alias** »

Déployer LDAP

Déployer LDAP

- Déployer un service d'annuaire LDAP nécessite
  - Une réflexion sur la nature des données que l'on y met,
  - Sur la manière dont on les récupère,
  - Sur l'utilisation que l'on compte en faire et sur 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

Déterminer les besoins en service d'annuaire et ses applications

- Déployer un système d'annuaire se fait généralement sous la contrainte de la mise en place ou du remplacement d'une application
- C'est alors que se pose la question d'élargir le service à d'autres types d'applications, la première venant à l'esprit étant un annuaire des personnes
- Cette phase consiste donc à prévoir toutes les applications possibles, actuelles ou futures, d'un annuaire LDAP

Déterminer quelles données sont nécessaires

- Il s'agit d'inventorier la liste exhaustive des données que l'on souhaite inclure dans le système d'information et de déterminer ensuite par quelle source les obtenir et les maintenir à jour
- Les aspects à considérer sont :
  - le format,
  - la taille des données,
  - leur confidentialité,
  - leur pertinence,
  - leur source (statique, dynamique...),
  - leur pérennité,
  - les personnes susceptibles de les fournir, de les maintenir et d'y accéder

Déterminer quelles données sont nécessaires

- De ce point de vue, c'est la plus délicate à franchir car elle implique d'autres intervenants, comme (par exemple) :
  - Le service du personnel,
  - Et la plupart du temps, tout un tas de sources d'informations diverses et variées qu'il faudra répertorier et trier (des bases de données, des tableaux Excel, des fichiers texte...)
- Il faut également se faire une idée précise sur la manière dont les données vont être maintenues à jour : synchronisation avec un SGBD, intervention manuelle, scripts automatiques…

Choisir son schéma

- Dans cette phase de design du schéma, il s'agit de choisir, en fonction des données que l'on a retenues, quelles sont les *classes d'objets* et les *types d'attributs* qui s'en rapprochent le plus pour construire son annuaire LDAP

Choisir son schéma

- La plupart du temps, les schémas standards, issus de X500 et de LDAP conviennent aux besoins de modélisation
- De plus, en fonction du logiciel que l'on choisira, des objets supplémentaires seront fournis
- Il reste, au final, la possibilité de créer ses propres objets, spécifiques à ses besoins
- En règle générale, il faut éviter de modifier le schéma existant car l'on risque de rendre son annuaire inutilisable par les applications clients ou les autres serveurs

Choisir son schéma

- Il est préférable de créer une *sous classe* d'une classe d'objet existante et exploiter le mécanisme d'héritage d'*attributs* des *classes objets*
- Par exemple on peut vouloir créer la classe d'objet *ulcoPerson* fille de *inetOrgPerson* dans laquelle on définira les attributs nécessaires à ses besoins :

Choisir son schéma

```
objectclass ulcoPerson
superior inetOrgPerson
requires sn,
      cn
allows   uidNumber,
      gidNumber,
      homeDirectory,
      loginShell,
      dateArrive,
      dateDepart
```

Choisir son schéma

- Dans tous les cas, il faut prévoir de documenter son schéma pour en faciliter la maintenance et l'évolution
- Il faut proscrire également la désactivation de l'option de *schema checking* implantée dans la plupart des serveurs, qui permet de vérifier que les attributs saisis sont bien conformes au schéma que l'on a choisi

Concevoir son espace (modèle) de nommage

- Cette étape consiste à définir comment les entrées de l'annuaire vont être organisées, nommées et accédées
- L'objectif est de faciliter leur consultation et leur mise à jour mais aussi de prévoir leur duplication, leur répartition entre plusieurs serveurs ou leur gestion par plusieurs personnes
- En fonction de ces priorités, on privilégiera tel ou tel espace de nommage

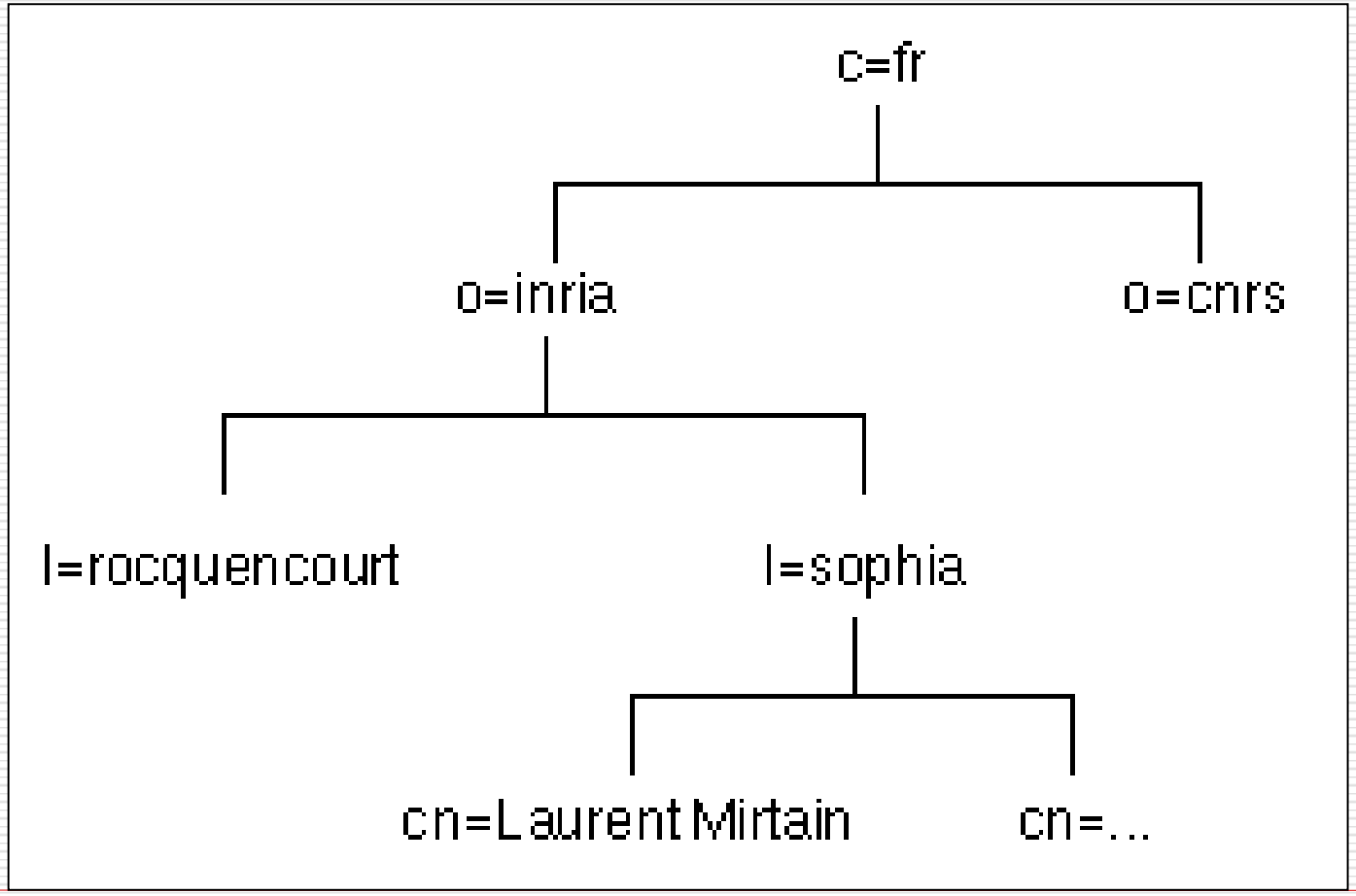
Concevoir son espace (modèle) de nommage

- Les paramètres qu'il faut prendre en compte lors de cette étude sont les suivants :
  - Le nombre d'entrées prévu et son évolution ?
  - La nature (type d'objet) des entrées actuelles et (surtout) futures ?
  - Vaut-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 ?
- Durant cette phase, il faut choisir le modèle d'organisation des données, leur mode de désignation et le suffixe de notre organisation

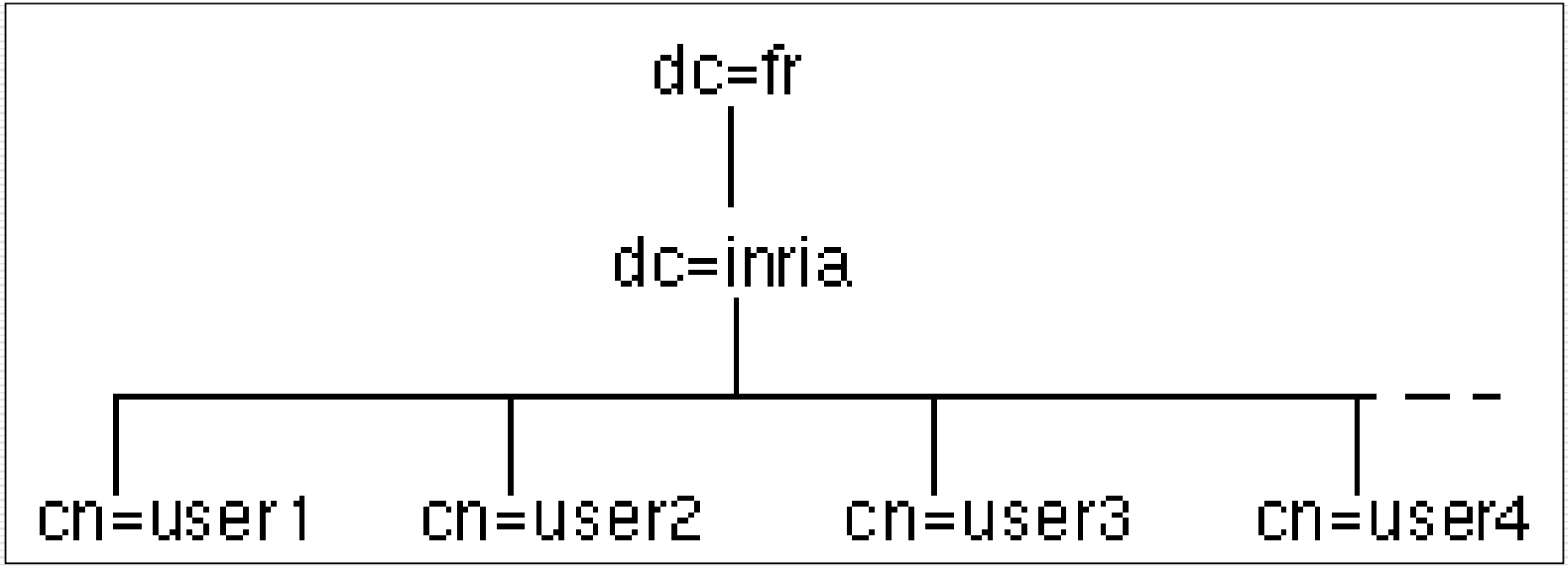
Le Directory Tree

- Le modèle LDAP, n'impose pas une racine universelle du Directory Tree car il renonce à être un service d'annuaire mondial et se limite à une petite communauté
- Dans ce cadre, le modèle LDAP peut être :
  - *Plat*,
  - *Découpé* pour refléter l'organisation interne,
  - *Branché* par type d'objet ou en vue :
    - De faciliter la duplication entre serveurs, la délégation de gestion,
    - La définition de règles d'accès spécifiques à une branche

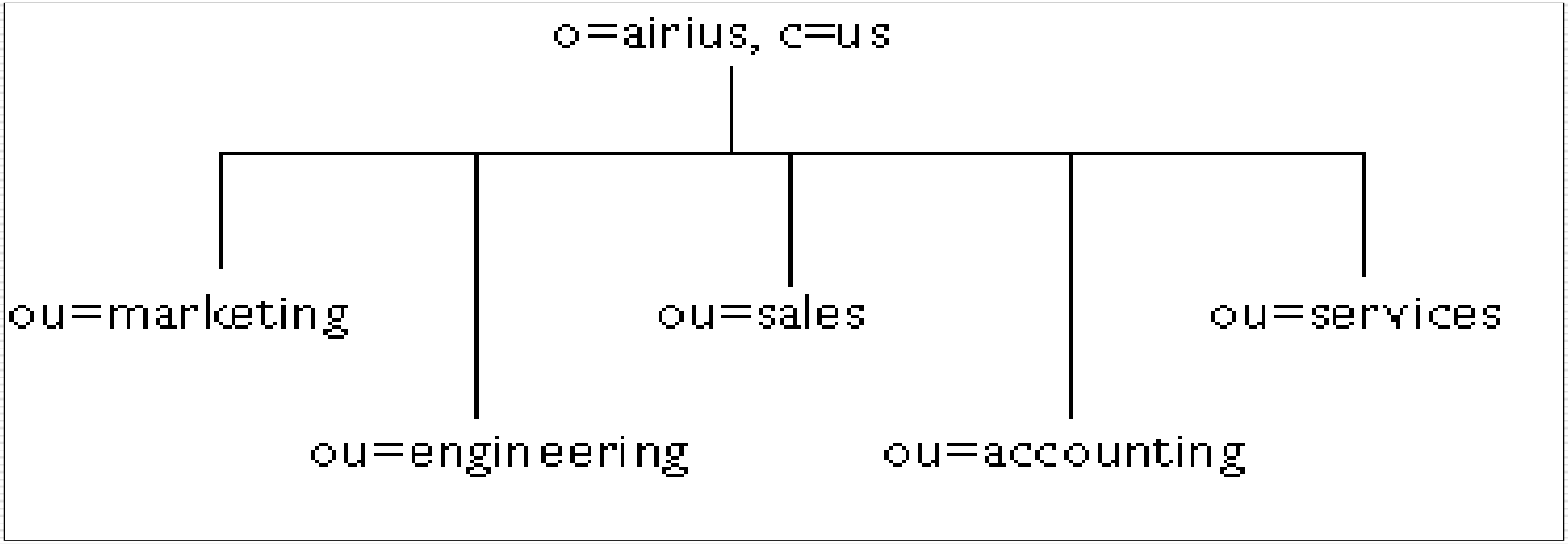
Espace de nommage X500



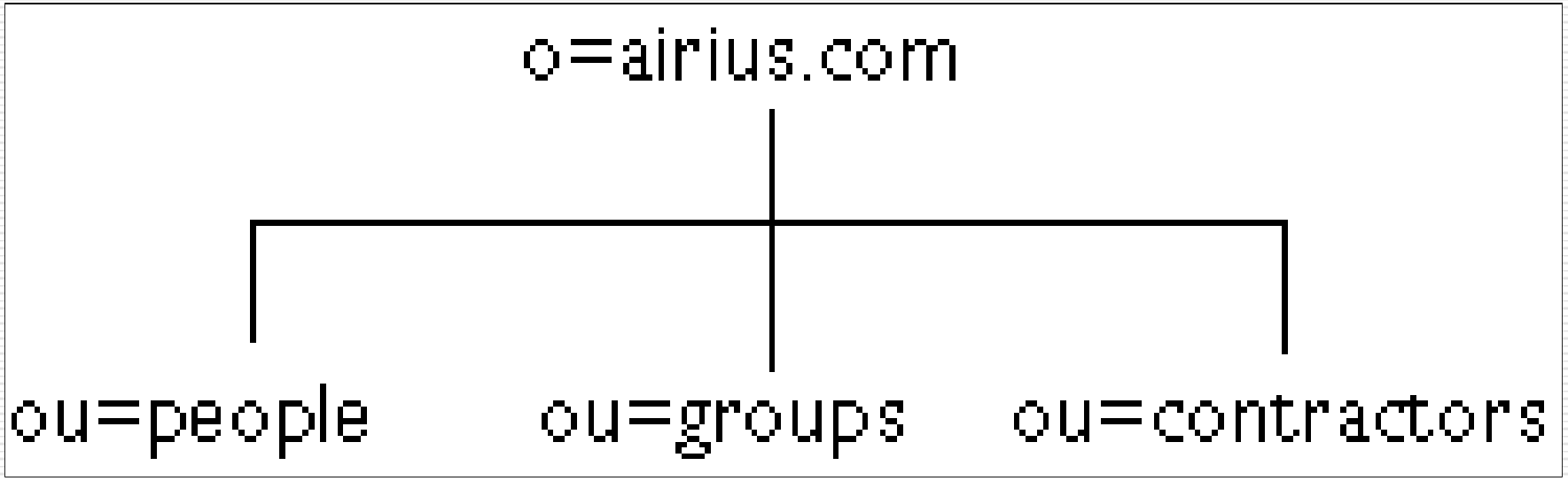
Espace de nommage plat



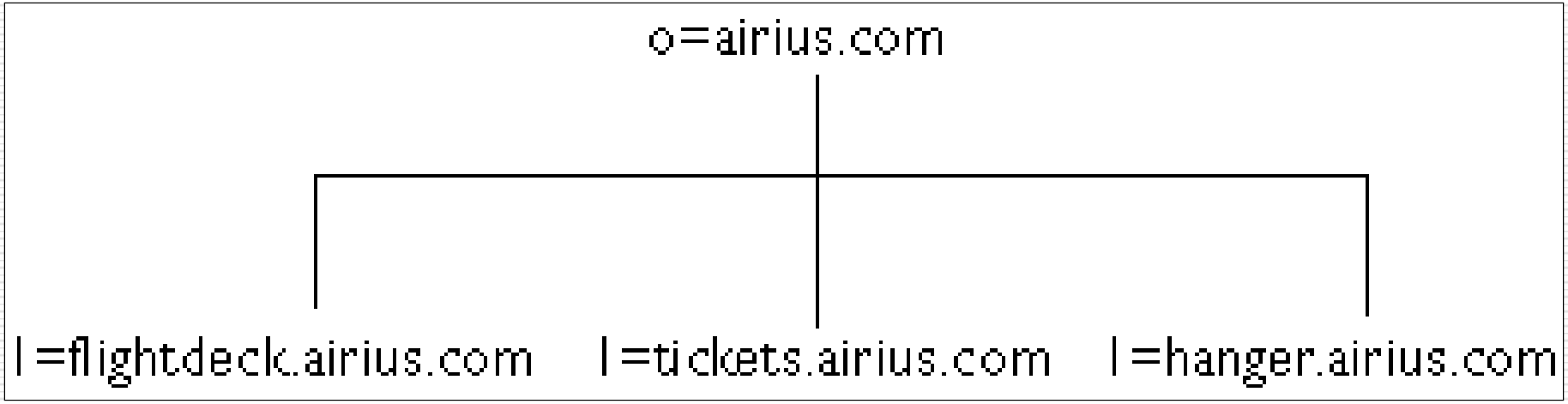
Espace de nommage basé sur l'organisation



Espace de nommage basé sur les objets



Espace de nommage en vue de dupliquer



Le Directory Tree

- En règle générale, en terme de performance, il vaut mieux avoir un arbre le plus plat possible
- Par contre, en terme de facilité d'administration, il vaut mieux introduire du branchage par type d'objet ou par organisation
- Cela facilite les mises à jour de données ou la mise en place de règles d'accès spécifiques à une partie de l'annuaire ou la répartition de la gestion de l'arbre entre plusieurs serveurs
- Si l'organisation change souvent ou bien que le personnel est très mobile, le branchage par organisation est alors à proscrire

Nommage des entrées

- La deuxième étape consiste à choisir l'attribut utilisé pour le DN de l'entrée, dans la partie RDN
- Ce choix peut dépendre des applications clientes, mais il doit se faire surtout de manière à garantir l'unicité de la valeur utilisée
- Pour les personnes, le *canonicalName* (cn) n'apporte pas de garantie d'unicité et il faut lui préférer un attribut comme l'*uid* ou le *mail* ou encore l'*employeenumber*, la règle consistant à choisir un attribut qui a peu de chance de changer dans le temps et qui est unique pour chaque entrée
- Pour les autres types d'objets, on utilisera l'attribut *cn*

Choix du suffixe

- La dernière étape consiste à choisir le suffixe
- C'est en quelque sorte l'identifiant de l'annuaire
- Son choix est important car, même si la base n'a qu'une vocation interne, elle peut à terme devenir en partie un maillon d'un annuaire global ou d'un système d'information
- Le suffixe peut, à l'extrême, être une chaîne vide, mais dans l'optique d'une diffusion de son annuaire, on choisira, en général, un suffixe unique au monde
- Pour cela, il est recommandé d'utiliser comme suffixe d'annuaire, le nom de domaine DNS de l'organisation

Choix du suffixe

- Dans la norme X500 le top level est le pays et vient ensuite le nom de l'organisation, ce qui donne par exemple comme suffixe :

```
o=UNIV-LITTORAL, c=FR
```

Choix du suffixe

- Le danger d'un tel nommage est qu'aucun organisme ne contrôle l'attribution des suffixes (la communauté X500 avait commencé à le mettre en place) et que rien ne garantit donc l'unicité de celui-ci
- Entre temps, l'Internet s'est développé et la problématique d'attribution des noms de domaines DNS a été prise en compte de manière globale
- Le choix du nom de domaine DNS comme suffixe de son annuaire s'impose donc naturellement
- Il pourra s'exprimer sous deux formes :
  - Utilisation de l'attribut *organization* (o) :

    `o=univ-littoral.fr`
  - Utilisation de l'attribut *Domain Component* (dc) défini par le RFC 2377 :

    `dc=univ-littoral, dc=fr`

Choix du suffixe

- La dernière forme est préconisée par l'IETF car elle permettra, en utilisant le *Service Record* du DNS (SRV - mappe un nom d'hôte à un type de service donné) de déterminer automatiquement le serveur LDAP à contacter, à partir du DN utilisé dans une requête
- Par exemple le DN **`uid=toto,ou=people,dc=univ-littoral,dc=fr`** renvoie intuitivement sur le domaine DNS **`univ-littoral.fr`**
- A partir de cette déduction, une requête sur l'entrée SRV du DNS fournira les coordonnées du serveur LDAP à contacter
- Ci-dessous un exemple de *record* DNS de type SRV pour un service de type LDAP :

```
_ldap._tcp.univ-littoral.fr.   IN   SRV   0 0 389 ldap.univ-littoral.fr
```

Choix du suffixe

- Cette phase est certainement la plus casse-tête dans le cas d'une organisation un peu complexe
- Il faut donc parfois faire des compromis visant à prendre la moins mauvaise solution, en essayant de définir les facteurs les plus contraignants

Définir la topologie de son service

- Dans cette phase, il faut réfléchir sur la manière dont le service d'annuaire LDAP va être rendu en termes de performance, de fiabilité et de facilité de gestion
- Il faut prendre en compte :
  - Les applications qui vont utiliser l'annuaire et le nombre d'utilisateurs
  - Les capacités du logiciel serveur qui va être choisi
  - La topologie de son réseau
  - Le design de son espace de nommage

Définir la topologie de son service

- La conception de la topologie du service d'annuaire LDAP est étroitement liée à celle de l'espace de nommage
- Il est donc possible de devoir revenir sur l'une ou l'autre durant la phase de conception ou même celle d'exploitation, dans le cas d'un changement d'organisation interne ou de marque de logiciel serveur

Définir la topologie de son service

- La question principale de cette phase est de déterminer si la base et sa gestion seront centralisées sur un seul serveur ou si elles devront être éclatées sur plusieurs serveurs
- La deuxième étude porte sur le nombre de serveurs redondants à déployer et leur emplacement sur le réseau physique

OpenLDAP

Introduction et bref historique

- OpenLDAP est, à ce jour, et, à ma connaissance, l'implémentation libre la plus utilisée
- OpenLDAP est un projet libre diffusé sous licence « OpenLDAP Public License » (http://www.openldap.org/license.html)
- Il est supporté par la fondation OpenLDAP, créée en 1998 par une société du nom de « Net Boolean », fournisseur de services professionnels liés à la messagerie.

Introduction et bref historique

- La première version d'OpenLDAP (1.0) sort en août 1998
- Il faudra attendre août 2000 pour que la version 2.0 ne voie le jour, offrant le support de LDAP v3
- La version stable actuelle est la version 2.4, sortie en octobre 2007
  - Elle apporte des améliorations telles qu'une meilleure montée en charge et une meilleure internationalisation, ainsi que de nouvelles fonctionnalités comme le support des transactions.

Les outils fournis par OpenLDAP

- Le projet OpenLDAP implémente un serveur LDAP, mais également les commandes clientes permettant de manipuler des informations contenues dans l'annuaire

Les outils fournis par OpenLDAP

- Les commandes liées au serveur

```
# dpkg -L slapd | grep bin | sort
/usr/sbin
/usr/sbin/slapacl
/usr/sbin/slapadd
/usr/sbin/slapauth
/usr/sbin/slapcat
/usr/sbin/slapd
/usr/sbin/slapdn
/usr/sbin/slapindex
/usr/sbin/slappasswd
/usr/sbin/slaptest
```

Les outils fournis par OpenLDAP

- Les commandes liées au serveur
  - Démons :
    - **slapd** : le démon OpenLDAP !
    - [**slurpd** : le démon de réplication]
  - Commandes de manipulation de la base (backend) gérée par OpenLDAP
    - **slapindex** : crée les index au sein de la base
    - **slapcat** : effectue un dump (une copie intégrale) de la base
    - **slapadd** : ajoute des entrées LDIF dans la base
    - **slappasswd** : utilitaire de conversion de mots de passe
  - Commandes de test/validation :
    - **slaptest** : teste la validité du fichier de configuration slapd.conf
    - **slapdn** : teste la conformité d'un DN donné en ligne de commande
    - **slapacl** : vérifie l'accessibilité d'une liste d'attributs
    - **slapauth** : vérifie les authentifications

Les outils fournis par OpenLDAP

- Les commandes clientes

```
# dpkg -L ldap-utils | grep bin | sort
/usr/bin
/usr/bin/ldapadd
/usr/bin/ldapcompare
/usr/bin/ldapdelete
/usr/bin/ldapexop
/usr/bin/ldapmodify
/usr/bin/ldapmodrdn
/usr/bin/ldappasswd
/usr/bin/ldapsearch
/usr/bin/ldapurl
/usr/bin/ldapwhoami
```

Les outils fournis par OpenLDAP

- Les commandes clientes
  - **ldapadd** : ajoute une entrée
  - **ldapcompare** : permet de comparer l'attribut d'une entrée à une valeur spécifiée
  - **ldapdelete** : supprime une entrée
  - **ldapmodify** : modifie une entrée (ajoute/supprime un attribut, ajoute/supprime une entrée...)
  - **ldapmodrdn** : modifie le rdn d'une entrée (renomme une entrée)
  - **ldappasswd** : modifie le mot de passe d'une entrée LDAP
  - **ldapsearch** : effectue une recherche au sein de l'annuaire
  - **Ldapurl** : outil de formattage d'URL LDAP
  - **ldapwhoami** : affiche avec quel utilisateur le binding a eu lieu