Introduction aux annuaires LDAP

Claude Duvallet

Université du Havre
UFR Sciences et Techniques
25 rue Philippe Lebon - BP 540
76058 LE HAVRE CEDEX
Claude.Duvallet@gmail.com

Plan de la présentation

Qu'est-ce qu'un annuaire ?

Un annuaire est similaire à une base de données :

- on peut y mettre des informations et les consulter.

mais il est plus spécialisé :

- il est dédié plus à la lecture qu'à l'écriture,
- l'accès aux données se fait par recherche multi-critères.

Des exemples d'annuaires sont :

- l'annuaire téléphonique,
- les carnets d'adresse,
- les répertoires de rues,
- les post-it.

Un service d'annuaire électronique, c'est aussi en plus :

- un protocole qui permet l'accès au contenu,
- une syntaxe qui permet d'interroger la base de données,
- un modèle de duplication des données,
- un modèle de distribution des données.

Caractéristiques d'un annuaire

Un annuaire est caractérisée par :

- une mise à jour dynamique : les données consultées sont régulièrement mises à jour.
- un contenu évolutif : des informations supplémentaires peuvent être ajoutées.
- une organisation plus flexible des données : il est possible de créer des index et de faire des recherches avancées.

Caractéristiques comparées des annuaires et des bases de données :

- le rapport lecture/écriture est plus élevé,
- les bases sont plus facilement extensibles,
- la diffusion se fait à plus large échelle,
- la répartition des données entre des serveurs est plus éclatée,
- il y a duplication de l'information,
- l'importance des standards est plus grande,
- il existe des possibilités d'avoir de fortes quantités d'enregistrements mais de faibles capacités de stockage.

Un annuaire ≈ un entrepôt de données

Un annuaire électronique est une sorte d'entrepôt de données qui les rends disponibles pour des applications ou des utilisateurs.

- des mots de passe ou des certificats d'authentification.
- des adresses de courriels.
- des informations de contact : téléphone, adresse, bureau...
- des profils de configuration de logiciels.
- etc.

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 définissant comment on accède à l'information ,
- un modèle de sécurité définissant comment données et accès sont protégés,
- un modèle de duplication définissant comment la base est répartie entre serveurs,
- des APIs pour développer des applications clientes,
- LDIF, un format d'échange de données.

Historique du protocole LDAP

$\Longrightarrow$ Apparition de LDAP en 1993.

Lightweight Directory Access Protocol (LDAP) est né de l'adaptation de X.500 DAP au protocole TCP/IP.

Deux groupes de travail aboutissent à deux produits fonctionnant comme frontal X.500 :

- Directory Assistance Service (DAS) : RFC 1202.
- 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 garde beaucoup d'aspects de X.500 dans les grandes lignes, mais va dans le sens de la simplification.

Le protocole LDAP (1/3)

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 des données :
  - pas l'ASCII (comme pour HTTP, SMTP,...) mais le *Basic Encoding Rules (BER)*, sous une forme allégée (appelée LBER Lightweight).
- Les mécanismes de sécurité :
  - les méthodes de chiffrement,
  - les mécanismes des règles d'accès aux données.

Le protocole LDAP (2/3)

Le protocole définit (suite) :

- Les opérations de base :
  - interrogation : `search`, `compare`.
  - mise à jour : `add`, `delete`, `modify`, `rename`.
  - connexion au service : `bind`, `unbind`, `abandon`.
- Communication *client-serveur* :
  - normalisée par l'IETF : la version actuelle est LDAPv3 (RFC 2251).
- Communication *serveur-serveur* :
  - le *referral service* est définit par LDAPv3,
  - le *replication service* a été normalisé sous la dénomination *LDAP Duplication Protocol (LDUP)*.

Le protocole LDAP (3/3)

LDAPv3 est conçu pour être extensible sans avoir à modifier la norme grâce à trois concepts :

- *LDAP extended 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.

Le modèle d'information (1/14)

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.

Le modèle d'information (2/14)

Le schéma :

- Le *Directory schema* est la « charte » qui définit, pour le serveur, l'ensemble des définitions relatives aux objets qu'il sait gérer.
- Le schéma décrit les *classes d'objets*, leurs types d'*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.

Le modèle d'information (3/14)

Un attribut ou un type d'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
- un indicateur d'usage
- un format ou une limite de taille de valeur qui lui est associée

type d'attribut	valeur de l'attribut
cn :	Lætitia Casta
uid :	lcasta
telephonenumber :	+33 (0)1 4852 7738
mail :	Laetitia.Casta@univ-lehavre.fr
roomnumber :	C105

TAB.: Exemple de d'attributs pour une entrée

Le modèle d'information (4/14)

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.

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 téléphone
int	integerMatch	nombre entier, comparaison numérique
dn	distinguishedNameMatch	nom d'entrée, règles spécifiques
bin	octetStringMatch	données binaires, comparaison byte/byte

TAB.: Exemple de syntaxes d'attributs

Le modèle d'information (5/14)

Deux catégories d'attributs :

- *User attributes :* attributs « normaux » manipulés par les utilisateurs (`givenname`, `telephoneNumber`),
- *Operational attributes :* attributs « 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`

Le modèle d'information (6/14)

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).
- Exemples de classes d'objet :
  - → une organisation (`o`),
  - → ses départements (`ou`),
  - → son personnel (`organizationalPerson`),
  - → ses imprimantes (`device`),
  - → ses groupes de travail (`groupofnames`).

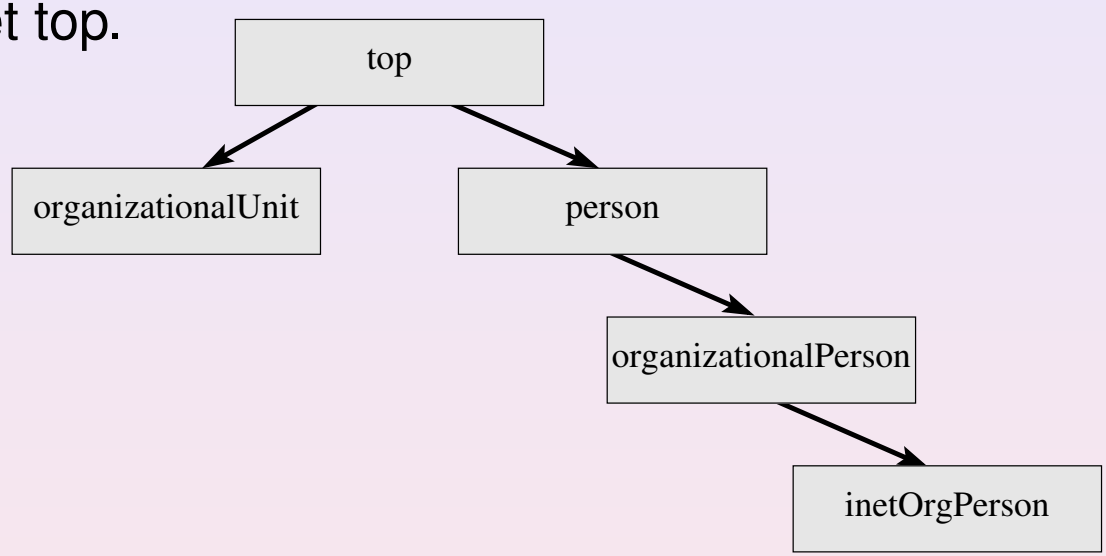
Le modèle d'information (7/14)

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.
- Une *classe abstraite* désigne des objets basiques de LDAP.

Le modèle d'information (8/14)

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.

Le modèle d'information (9/14)

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.

Le modèle d'information (10/14)

Les OIDs

- Les classes d'objets et les attributs sont normalisés par le RFC2256.
  - → 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             - the Internet OID
1.3.6.1.4.1         - IANA-assigned company OIDs, used for private MIBs
1.3.6.1.4.1.4203    - OpenLDAP
```

Le modèle d'information (11/14)

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`.

Le modèle d'information (12/14)

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
```

Le modèle d'information (13/14)

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}
```

Le modèle d'information (14/14)

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 ) )
```

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*.

Le modèle de nommage (1/5)

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).

Le modèle de nommage (2/5)

Le *Directory Information Tree* (DIT) :

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

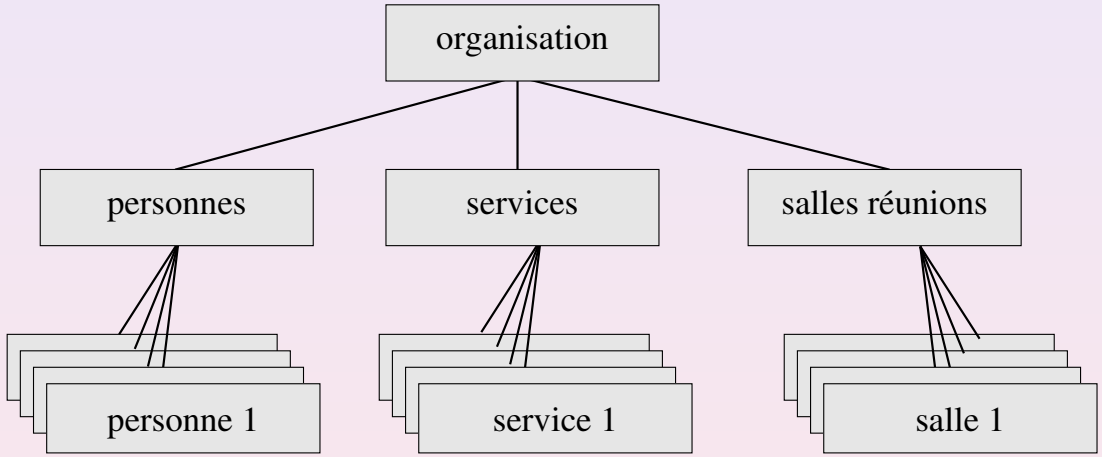


FIG.: Exemple de modélisation d'une organisation

- Chaque nœud de l'arbre correspond à une entrée de l'annuaire ou *directory service entry* (DSE).
- Au sommet de l'arbre se trouve l'entrée *Suffix* ou *Root Entry* ou *BaseDN*, qui caractérise une base LDAP.

Le modèle de nommage (3/5)

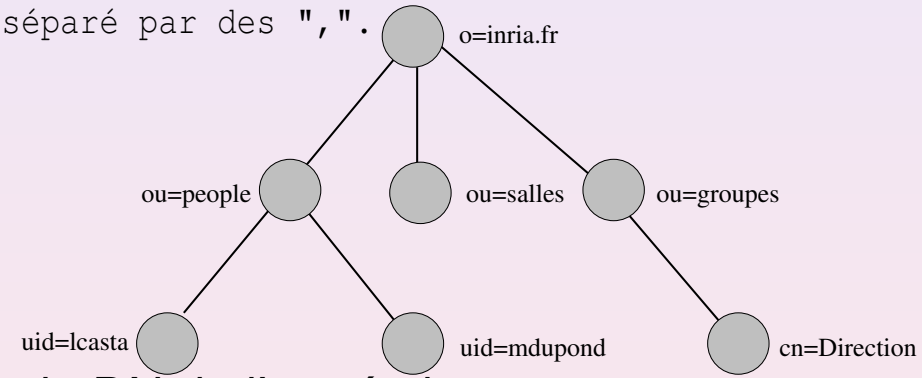
- Le suffix définit l'espace de nommage dont le serveur a la gestion.
- Un serveur peut gérer plusieurs arbres (donc plusieurs suffixes).
- Il possède une entrée spéciale, appelée *root directory specific entry* (rootDSE) qui contient la description du DIT.
  - $\Rightarrow$ Avec LDAP, vous êtes libres d'organiser vos données comme bon vous semble (*design du DIT*).
  - $\Rightarrow$ Des contraintes (performance, gestion...) impliqueront de choisir tel ou tel type de modèle (cf. déploiement).

Le modèle de nommage (4/5)

Le *Distinguish Name* (DN)

- Référence de manière unique une entrée du DIT (⇔chemin d'un fichier UNIX).

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



  → Exemple : le DN de l'entrée lcasta vaut :

  `uid=lcasta, ou=people, dc=inria, dc=fr`

- Chaque composant du DN est appelé *Relative Distinguish 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.

Le modèle de nommage (5/5)

Alias et referral

- Deux objets abstraits particuliers : les *aliases* et les *referrals*
  - $\rightarrow$ permettent à une entrée de l'annuaire de pointer vers une autre entrée du même ou d'un autre annuaire.

  - $\Rightarrow$ L'attribut `aliasObjectName` de l'objet `alias` a pour valeur le DN de l'entrée pointée.
  - $\Rightarrow$ L'attribut `ref` de l'objet `referral` a pour valeur l'URL LDAP de l'entrée désignée.

Le modèle fonctionnel

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

1. Les opérations d'interrogation.
2. Les opérations de comparaison.
3. Les opérations de mise à jour.
4. Les opérations d'authentification et de contrôle.

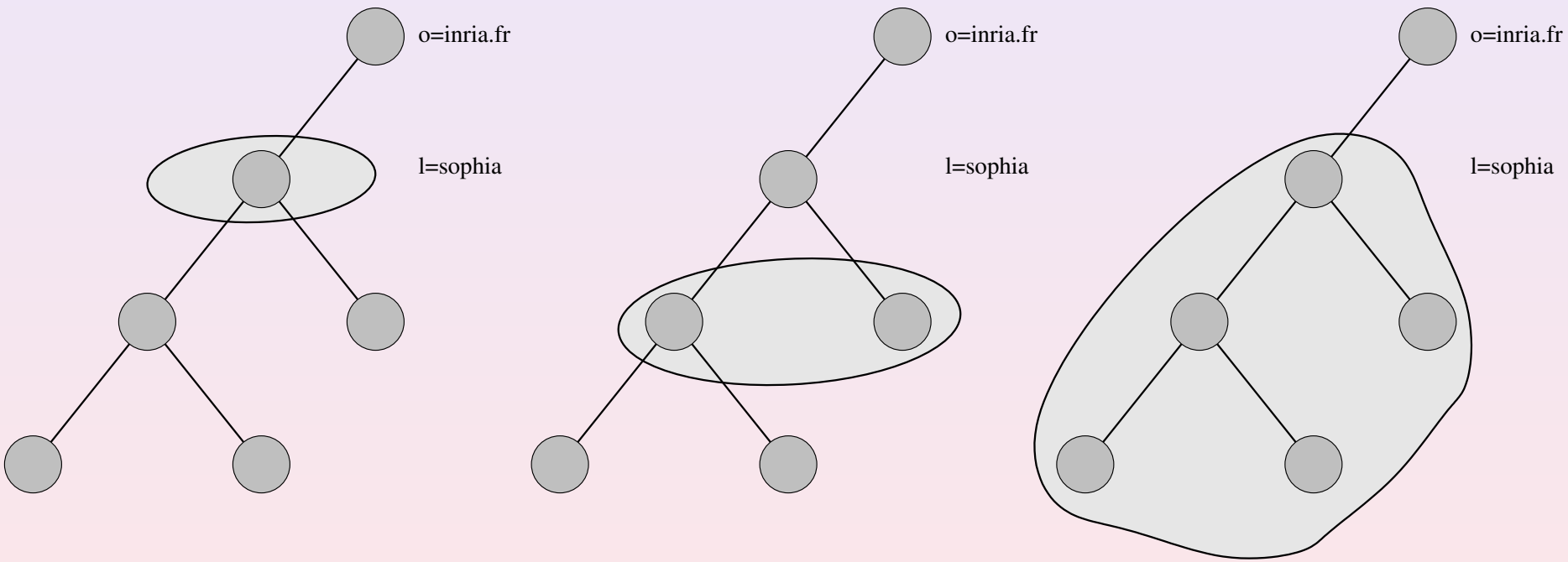
Les opérations d'interrogation (1/3)

- 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 :

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

Les opérations d'interrogation (2/3)

Le scope



search base = "l=sophia, dc=inria, dc=fr"

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

Les opérations d'interrogation (3/3)

Les filtres de recherche (RFC 1558)

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

`(cn=Laurent Mirtain)`	égalité	Nom vaut "Laurent Mirtain"
`(cn=*Mart*)`	sous-chaîne	Nom contient "Mart"
`(cn~=martin)`	approximation	Nom sonne comme "martin"
`(employeenumber>=100)`	comparaison	Numéro supérieur à 100
`(sn=*)`	existence	Tous les noms propres
`(&(sn=Mirtain)(l=sophia))`	ET	Nom vaut "Mirtain" ET localisation vaut Sophia
`(|(ou=sophia)(ou=rocquencourt))`	OU	ou vaut sophia ou rocquencourt
`(!(tel=*))`	NON	Toutes les entrées sans attribut téléphone

TAB.: Exemples de filtres de recherche

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

Les opérations de comparaison

- Comparaison Héritage de X.500 : vérifier si l'attribut d'une entrée contient bien une valeur spécifiée. Le serveur répond vrai ou faux.
- Équivalent à une recherche, sauf que le serveur 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.

Les opérations de mise à jour

$\Rightarrow$ 4 opérations : add, delete, rename, modify
Ces quatre opérations nécessitent les droits de contrôle appropriés et des prérequis :

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

Les opérations d'authentification et de contrôle

$\Rightarrow$ 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.

Le 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 *de contrôle d'accès* aux données.
  - par le *chiffrement* des transactions entre les clients et les serveurs ou entre les serveurs.

L'authentification

- L'authentification LDAP est un protocole avec connexion : il faut s'authentifier pour ouvrir la connexion (`bind`) en fournissant une identité.
- LDAPv3 propose plusieurs choix d'authentification :
  - *Anonymous authentification* - accès sans authentification permettant de consulter les données accessibles en lecture pour tous.
  - *Root DN authentification* - accès administrateur (tous les droits).
  - *Mot de passe en clair* - un DN plus un password qui transite en clair sur le réseau.
  - *Mot de passe + SSL 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.

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 ou d'autres types.

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

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.

Le modèle de duplication (1/4)

- 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.

Le modèle de duplication (2/4)

- 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*.

Le modèle de duplication (3/4)

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 versus annuaire externe).

Plusieurs manières de synchroniser les serveurs :

- mise à jour totale ou incrémentale...

Plusieurs stratégies de duplications :

- *single-master replication, multiple-master replication, cascading replication* .

Le modèle de duplication (4/4)

- 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 du *replication service* nécessite de le prévoir au moment du design du DIT.

Les 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)

LDIF

- LDAP Data Interchange Format (LDIF) est le standard de représentation des entrées sous forme de texte.
- Il est utilisé pour afficher ou modifier les données de la base suivant deux modes :
  - faire des importations/exportations de la 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.

LDIF : le mode import

La forme générale est :

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

Une 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
```

LDIF : le mode commande (1/2)

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.
```

LDIF : le mode commande (2/2)

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=AceIndustry, c=US
```

LDIF : le format ASCII

- 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/9sAQwAIBgYHBgU
   wcHCQkICgwUDQwLCwwZEhMPFB0aHx4dGhwcICQuJyAiLCMcHCg3KSwwMTQ0NB8nOT04MjwuMzQ
```

- LDAP 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 de 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.loria.fr/cn=Laurent%20Mirtain,ou=Moyens%20Informatiques,o=loria.fr
ldap://ldap.loria.fr/o=loria.fr?mail,uid?sub?(sn=Mirtain)
```

Les avantages d'OpenLDAP

- OpenLDAP est open source disponible sous la licence publique (OpenLDAP Public License) à l'adresse `http ://www.openldap.org/.`
- OpenLDAP 2 est conforme à la norme LDAPv3.
- OpenLDAP existe pour de nombreuses plate-formes dont Linux, Solaris, Mac OS 10.2, et de nombreuse versions de Windows.
- Le projet OpenLDAP s'inscrit dans la continuité du serveur LDAP de l'université du Michigan.

Le fichier de configuration `slapd.conf`

- Ce fichier sert à stocker les informations de configuration du serveur autonome OpenLDAP (*slapd*), du démon auxiliaire de réplication (*slurpd*), ainsi que des outils associés tel que *slapcat* et *slapadd*.

Les butineurs OpenLDAP

- LDAP Browser/Editor est un butineur graphique LDAP en Java.
- JXplorer est un autre butineur graphique LDAP en Java.
- GQ est un butineur graphique s'appuyant sur GTK.
- Luma est un gestionnaire graphique, basé sur python-ldap, extensible via des plugins.
- Frood est un butineur Gtk-Perl/PerLDAP.
- phpLDAPadmin est un client web d'administration/exploration LDAP.
- KLDAP est un client ldap pour KDE.

$\Rightarrow$ Retrouvez tous ces butineurs sur la page du CRU : `http ://www.cru.fr/ldap/.`

Bibliographie

- *LDAP - Administration système.* **Gérald Carter**. Editions O'Reilly. 2004.
- *Les annuaires LDAP.* **Pierre-Yves Cloux et Rafael Corvalan**. Editions Dunod. 2004.
- *Annuaires LDAP.* **Rizcallah**. Editions Eyrolles. Novembre 2004.
  `http ://www.editions-eyrolles.com/Livre/9782212115048/annuaires-ldap`
- *Tutorial LDAP.* **Laurent Mirtain**.
  `http ://www-sop.inria.fr/semir/personnel/Laurent.Mirtain/ldap-livre.html`.
- *La page du CRU.* `http ://www.cru.fr/ldap/`. 2006.