- Exposé Nouvelle Technologies Réseaux -

LDAP

Lightweight Directory Access Protocol

Sylvain Pernot

Sébastien Laruée

Florent de Saint-Lager

Ingénieur 2000
Informatique et Réseau - 3ième année

Sommaire

Introduction

Ce dossier a été réalisé par Sylvain Pernot, Sébastien Laruée et Florent de Saint-Lager dans le cadre d'un exposé de « Nouvelles Technologies Réseaux » du cours du même nom d'Etienne Duris, maître de Conférences à l'Université de Marne-La-Vallée et responsable de la filière informatique réseaux (troisième année) du dispositif Ingénieurs 2000.

Ce dossier présente la technologie LDAP (*Lightweight Directory Access Protocol).*

La première partie de ce dossier est consacrée à la présentation des annuaires, à quoi ils servent, dans quels cas ils sont utilisés et leurs points forts et leurs limites.

La deuxième partie de ce dossier est consacrée à la présentation du protocole LDAP.

La troisième et dernière partie de ce dossier est consacrée à l'étude des implémentations du protocole LDAP, en particulier celle de OpenLDAP.

En conclusion, une synthèse de la technologie sera établie, visant à dresser les avantages et les limites du LDAP, ainsi que l'avenir de ce protocole.

1 Présentation des annuaires

Avant d'entrer dans l'explication du protocole ***LDAP***, il convient de présenter le système de recueil de données associé à ce protocole que sont les annuaires électroniques.

1.1 Concept des annuaires électroniques

Un annuaire électronique est un catalogue de données dont le but premier est de proposer, grâce à des fonctions de recherche, un accès rapide à ses ressources aux différents clients qui les consulte.

Les annuaires électroniques permettent, aussi de **comparer**, de **créer**, de **modifier** ou **effacer** des données qu'ils contiennent.

Les annuaires électroniques ont la même vocation que les annuaires dits « papier » (comme les annuaires des pages jaunes ou blanches). Cette vocation est de faciliter la localisation de tous types d'objets comme, par exemple :

- des personnes,
- des sociétés,
- des ressources Informatiques,
- des applications

Les annuaires électroniques apportent un certain nombre d'avantages comparé aux annuaires papier. On dit qu'ils sont :

Dynamique : en effet, par opposition aux annuaires papiers qui sont mis à jour une seule fois par an, tous changements sur les annuaires électroniques s'effectuent en temps réels.

La responsabilité de la mise à jour de l'annuaire est délégué à des administrateurs et, si le droits de modification leurs est donné, aux propriétaires des informations.
Les coûts de mise à jour sont donc très faibles.

Flexibles : Un annuaire électronique n'est jamais figé. Sa peut être modifiée facilement, à la volée, sans nécessiter de reconstruire tout l'annuaire. Il est possible d'ajouter de nouveaux champs (de nouveaux attributs en terminologie annuaire) en fonction des besoins; il est également possible d'ajouter des nouvelles familles d'objets.

Sécurisé : Les annuaires électroniques permettent de contrôler les informations affichées en fonction de l'identité de l'utilisateur.

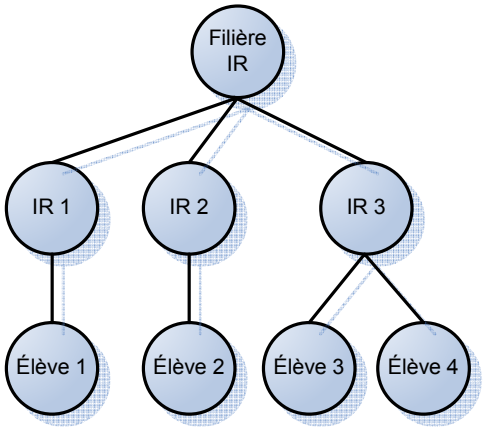
1.2 Caractéristiques des annuaires

Les annuaires électroniques sont *des bases de données spécialisées*. En effet, il existe un certains nombres de critères qui distingues les annuaires électroniques des bases de données conventionnelles :

- les annuaires sont conçus pour recevoir beaucoup plus de requête en lecture qu'en écriture,
- les données stockés de manières hiérarchique et ne sont pas relationnelles, comme elles le sont dans des bases de données conventionnelles,

L'exemple suivant permet de présenter la différence entre l'organisation des données dans un annuaire (à gauche) et dans une base de données (droite).
Cet exemple représente l'organisation des élèves dans les promotions de la filière Ingénieur 2000 :

Organisation hiérarchique de données type annuaire



Organisation relationnelle de données type base de données.

1	Élève 1	1
2	Élève 2	2
3	Élève 3	3
4	Élève 4	3

IR 1	1
IR 2	2
IR3	3

- La recherche d'informations dans les annuaires électroniques ne comporte pas de requêtes compliquées comme elle peut l'être avec les bases de données conventionnelles (jointures SQL).
- Les annuaires peuvent communiquer entre eux.

1.3 Exemples d'utilisation des annuaires

On pourrait croire que les annuaires électroniques ne servent qu'à rechercher des personnes ou des ressources, mais ceux-ci permettent bien d'autres applications tel que :

- constituer des carnets d'adresse
- authentifier des utilisateurs
- définir des droits d'accès à des utilisateurs
- recenser des informations sur un parc matériel
- décrire des applications.
- stocker et diffuser des certificats dans une Infrastructure de clé publique (PKI)

1.4 Les annuaires que nous utilisons

- DNS : ***domain name server*** ou ***domain name system***. Service de l'Internet assurant la conversion des noms de domaine en adresse IP.

- WHOIS : Base de données, autrefois gérée par l'Internic et désormais maintenue par Network Solutions, aussi connue sous le nom de « NICname ». Elle stocke pas mal d'informations sur le réseau lui-même (adresses des sites, des entreprises, noms de domaines, classes attribuées, gestionnaires locaux...).

- Base de Registre Windows

Maintenant que nous avons vu ce qu'était un annuaire électronique, nous allons nous pencher sur le protocole qui permet de les exploiter : ***LDAP***.

2 Présentation du protocole LDAP

LDAP (Lightweight Directory Access Protocol, traduisez Protocole d'accès aux annuaires léger et prononcez "èl-dap") est un protocole standard permettant de gérer des annuaires.

Le protocole LDAP, développé en 1993 par l'université du Michigan, avait pour but de remplacer le protocole DAP (servant à accéder au service d'annuaire X.500 de l'OSI), en l'intégrant à la suite TCP/IP.

Le protocole LDAP est actuellement à la version 3 (LDAPv3) et a été normalisé par l'IETF. LDAPv3 est défini par neuf documents RFC: de 2251 à 2256, 2829, 2830, 3377 :

RFC 2251	: Lightweight Directory Access Protocol (v3)
RFC 2252	: Lightweight Directory Access Protocol (v3): Attribute Syntax
RFC 2253	: Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names
RFC 2254	: The String Representation of LDAP Search Filters
RFC 2255	: The LDAP URL Format
RFC 2256	: A Summary of the X.500(96) User Schema for use with LDAPv3
RFC 2829	Authentication Methods for LDAP
RFC 2830	: Lightweight Directory Access Protocol (v3): Extension for Transport Layer Security
RFC 3377	: Lightweight Directory Access Protocol (v3): Technical Specification

2.1 Rappel de X.500

Le standard X.500 a été établi pour normaliser les annuaires électronique, quel que soit leur domaine d'application.

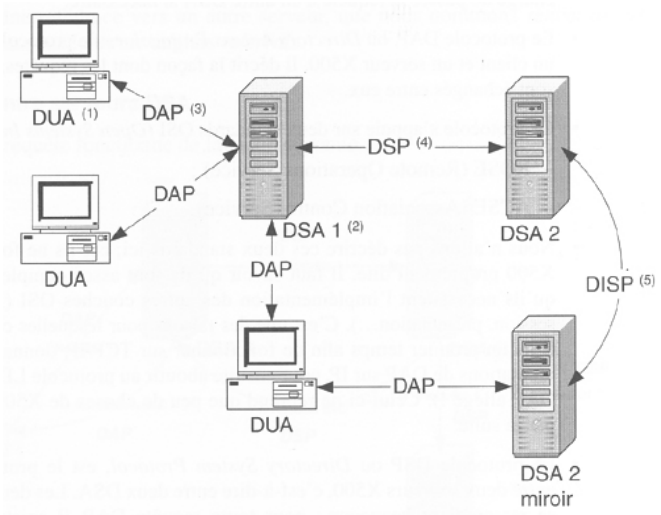
L'objectif de cette normalisation est de mettre à disposition de l'industrie des télécommunications un standard, indépendant de tous constructeur, capable de faire fonctionner ensemble une multitude d'annuaires à l'échelle mondiale, afin de

constituer un annuaire Pages Blanches et Pages Jaunes virtuel unique. Elle doit permettre à chaque pays de mettre à jour son propre annuaire et d'interroger les autres de la même manière.

Ses caractéristiques sont les suivantes :

- l'ouverture : qui assure une interconnexion des annuaires,
- l'extensibilité : qui permet de modifier simplement la structure des données tout en conservant la compatibilité avec la structure d'origine et d'être ainsi adaptable à toutes sortes de besoins.
- La Distribution : qui rend capable de répartir ou de répliquer les données sur plusieurs serveurs.

2.1.1 Composant d'un annuaire X.500



Composants d'un Système d'annuaire X.500

Source : Marcel Rizcallah, Annuaire LDAP, Eyrolles

DUA : *Directory User Agent* = client qui interroge l'annuaire

DAP : Protocol DAP, *Directory Access Protocol* = protocole de communication entre client et serveur X.500. Ce protocole s'appuie sur deux standards OSI qui sont ROSE (*Remote Operations Service*) et ACSE (*Association Control Service*).

DSA : *Directory System Agent* = serveur d'annuaire qui comprend la base de données appelée DIB (*Directory Information Base*). Ce composant peut soit dialoguer avec des clients, soit avec d'autre DSA

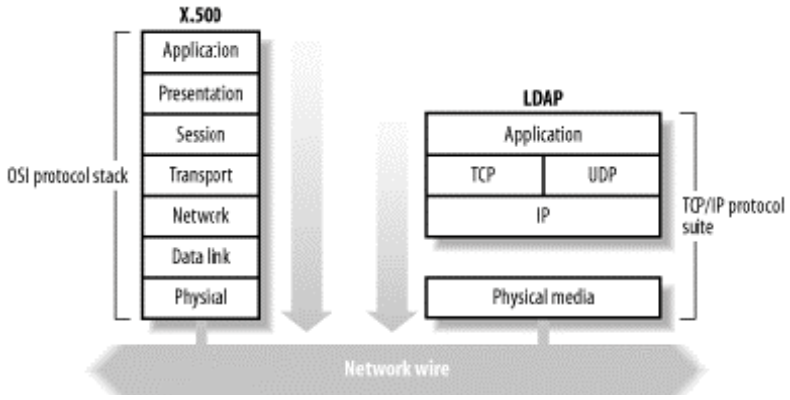
DSP : Protocol *Directory System Protocol* = protocole de communication entre deux serveur X.500. Semblable à DAP.

DISP : Protocol *Directory Information Shadowing Protocol* = protocole permettant la réplication d'un serveur DSA maître vers un autre serveur miroir.

2.2 La naissance de LDAP

DAP est un protocole « ***heavyweight*** » (lourd) car il nécessite que le client et le serveur communique en utilisant le modèle OSI. Ce modèle de sept couches est beaucoup plus lourd que le modèle TCP/IP qui n'en comporte que quatre.

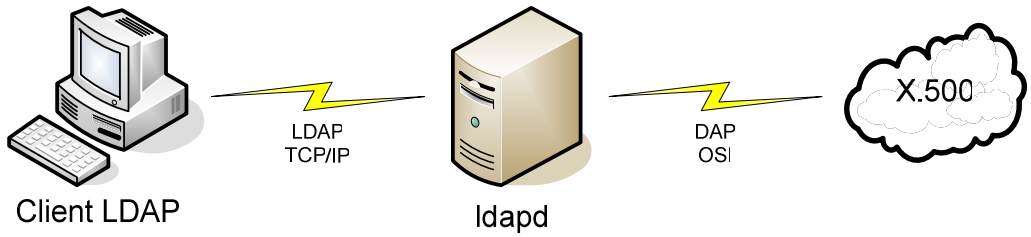
En 1993, l'université du Michigan a adapté le protocole DAP de la norme X.500 au protocole TCP/IP et mis au point LDAP.



X.500 sur OSI versus LDAP sur TCP/IP

Source : Gerald Carter, LDAP System Administration, O'Reilly

Il est initialement une passerelle d'accès à des bases d'annuaires X.500 (translateur LDAP/DAP).La première implémentation de LDAP contient le démon de LDAP (ldapd) qui est une passerelle entre LDAP et DAP



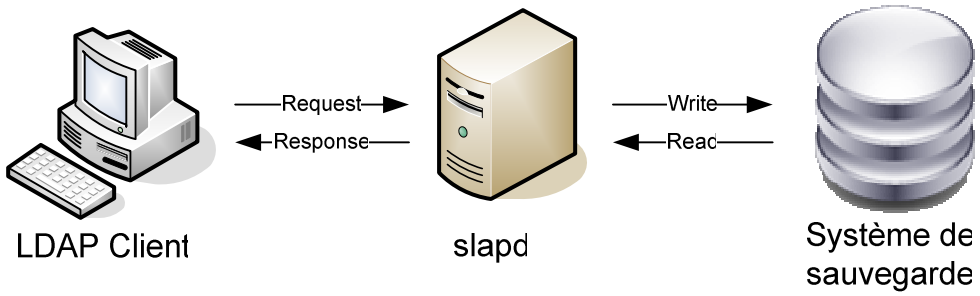
Passerelle LDAP – DAP

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

2.3 LDAP natif

A partir de 1995, LDAP est devenu un annuaire natif (*standalone LDAP*), afin de ne plus servir uniquement de passerelle d'accès à des annuaires X.500.

Standalone LDAP va gérer son propre mécanisme de sauvegarde de données qui va être incorporé au démon de LDAP : il s'agit de *slapd* (standalone ldap daemon).



Architecture d'annuaire avec des démons slapd

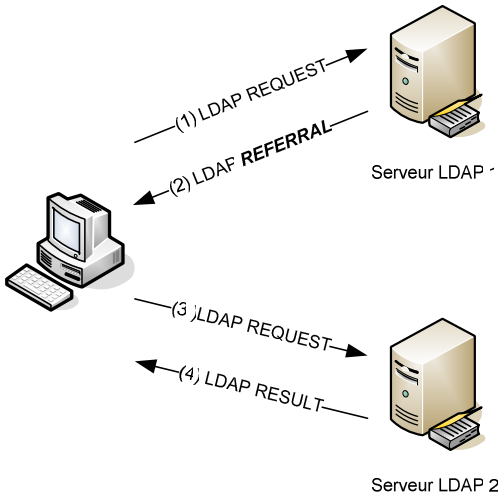
2.4 LDAPv3

L'étape suivante dans le développement de LDAP était LDAPv3, cette version, entièrement compatible avec LDAPv2, vise à combler les limitations de LDAPv2. Ces principaux ajouts sont :

- La prise en compte des caractères internationaux via le standard UTF-8 (Unicode Transformation Format-8).

- La standardisation du mécanisme de chaînage des requêtes par renvoie de référence (***referrals***), lorsque l'informations est répartie sur plusieurs serveurs LDAP.

  Le schéma suivant présente une architecture LDAP répartie, où l'information est distribuée sur plusieurs serveurs. Quand un client effectue une requête vers le serveur LDAP sur des données présente sur le serveur 2, celui-ci envoie une réponse (un renvoie par référence : ***referral***) au client lui indiquant que les informations qu'il recherche sont sur le serveur 2 :



Architecture répartie LDAP et
mécanisme de chaînage des requêtes (referral)

- La gestion de la sécurité pour l'authentification SASL (*Authentication and Security Layer*) et le transport des données confidentielles TLS (*Transport Layer Security*).
- La norme inclut maintenant des mécanismes d'extension. Il est possible de réaliser des opérations supplémentaires à celles décrites dans la norme, tout en s'appuyant sur le protocole existant. Il est aussi possible, par le biais de contrôle, de modifier le comportement des opérations de base.
- Un annuaire peut être interrogé pour accéder à son schéma, et pour connaître les extensions et les contrôles qu'il implémente.
- Intégration dans la norme LDAP du schéma X500. Certaines classes d'objets et attributs définis dans la norme X500 doivent être reconnus par les serveurs LDAP.

2.5 Les modèles

LDAP est défini par 5 modèles qui permettent de décrire différents aspects de l'annuaire: nommage, structure de stockage et structure hiérarchique, sécurisation et échange des données. Certains de ces modèles sont définis dans la norme comme le modèle d'échange de données ou de nommage. D'autres doivent être définis dans chaque annuaire comme le modèle d'authentification et le modèle d'information.

2.5.1 Le modèle d'information

Le modèle d'information du protocole LDAP définit le type de données pouvant être stocké dans l'annuaire LDAP.

On appelle **entrée** (en anglais **entry**) l'élément de base de l'annuaire. Chaque entrée de l'annuaire LDAP correspond à un objet abstrait ou réel (par exemple une personne, un objet matériel, des paramètres, ...). Une entrée est constituée de plusieurs objets.

L'ensemble des paramètres qui définissent le type des données ainsi que leurs syntaxes forment ce qu'on appelle le schéma de l'annuaire.

2.5.1.1 Le schéma de l'annuaire

Ainsi, on appelle **schéma** (plus exactement en anglais Directory Schema) l'ensemble des définitions d'objets et d'attributs qu'un serveur LDAP peut gérer ainsi que leur syntaxe. On peut donc définir des contraintes sur les entrées pour s'assurer de la validité des données insérées.

De cette façon, un annuaire peut uniquement comporter des entrées correspondant à une classe d'objet définie dans le schéma. Le schéma est en effet lui-même stocké dans l'annuaire à un emplacement spécifique (il s'agit pour être exact d'une instance de la classe **subschema**).

Grâce au schéma, l'annuaire peut garantir de façon autonome la validité des enregistrements et de leur syntaxe. Lorsqu'une entrée est créée dans l'annuaire, celui-ci vérifie sa conformité à la classe d'objet, on parle alors de schema checking.

2.5.1.2 Les attributs des entrées

Chaque entrée est constituée d'un ensemble d'attributs (paires clé/valeur) permettant de caractériser l'objet que l'entrée définit. On distingue habituellement deux types d'attributs:

- Les **attributs utilisateurs** (***user attributes***) sont les attributs caractérisant l'objet manipulé par les utilisateurs de l'annuaire (nom, prénom, ...)
- Les **attributs opérationnels** (***system attributes***) sont des attributs auxquels seul le serveur peut accéder afin de manipuler les données de l'annuaire (dates de modification, ...)

LDAP permet de définir des types d'attributs, c'est-à-dire des caractéristiques permettant de le définir de façon précise.
Chaque attribut possède de cette façon une syntaxe qui lui est propre (la façon selon laquelle l'attribut doit être renseigné, c'est-à-dire le format des données) mais aussi la manière selon laquelle la comparaison doit s'effectuer lors d'une recherche de l'annuaire (par exemple définir si la recherche sera sensible à la casse, c'est-à-dire si la recherche devra différencier minuscules et majuscules).

Voici les principales syntaxes d'attributs définies dans le protocole LDAPv3 :

syntaxe d'attribut	**description**
binary	Attribut constitué d'une suite d'octets, c'est-à-dire d'un fichier binaire (image, vidéo, fichier, ...)
boolean	Attribut constitué d'un booléen (vrai ou faux)
dn	Pointeur vers un objet de l'annuaire repéré par son *distinguished name*
Directory string	Attribut constitué d'une chaîne de caractères au format UTF-8
integer	Attribut constitué d'un entier
telephoneNumber	Numéro de téléphone

Voici les principales règles de comparaison d'attributs définies par le standard LDAPv3 :

règle de comparaison LDAP	règle de comparaison X500	description
cis	caseIgnoreMatch	Attribut texte non sensible à la casse
ces	caseExactMatch	Attribut texte sensible à la casse
tel	telephoneNumberMatch	Attribut texte représentant un numéro de téléphone (les virgules et les espaces sont ignorés dans la recherche)
int	integerMatch	Attribut entier (pour une comparaison numérique)
dn	distinguishedName	Nom d'entrée. Permet de comparer deux entrées
bin	octetStreamMatch	Attribut binaire. Permet de comparer octet par octet
bin	booleanMatch	Attribut bolléen. Permet de comparer deux attributs booléens

2.5.1.3 Les attributs prédéfinis

LDAP définit un ensemble de classes et d'attributs par défaut convenant pour la grande majorité des applications. Ces attributs doivent impérativement être implémentés par les serveurs d'annuaires LDAPv3. Cela permet de garantir une certaine homogénéité entre les différents annuaires.

Voici une petite liste non exhaustive des principaux attributs utilisateurs définis par le standard LDAPv3 :

Attribut	Description
aliasedObjectName	DN de l'objet dont celui en cours est un alias
authorityRevocationList	Liste de certificats révoqués par l'autorité chargée de les réguler
businessCategory	Activité professionnelle d'une entreprise ou d'une personne
c	Code du pays en deux lettres (respectant le standard ISO 3166)
caCertificate	Certificat de l'autorité de régulation
certificateRevocationList	Liste des certificats révoqués par l'autorité de régulation
cn	Nom de l'objet (*common name*)
description	Description de l'objet
distinguishedName	Nom distingué (utilisé par d'autres attributs par héritage)
facsimileTelephoneNumber	Numéro de fax
givenName	Prénom de la personne
houseIdentifier	Identifiant d'un batiment
initials	Initiales d'une personne
internationalSDNNumber	Numéro ISDN
l	localité de l'objet (géographique)
member	Distinguished Name des membres
name	Nom (utilisé par d'autres attributs par héritage)
o	Nom de l'organisation
objectClass	Classe d'objets

ou	Unité organisationnelle (branche de l'organisation)
owner	Nom du propriétaire de l'objet
postalAddress	Adresse postale (sans le code postal)
postalCode	Code postal
postalOfficeBox	Boîte aux lettres (postale)
presentationAddress	Adresse réseau de la présentation de l'objet (généralement une URL vers la présentation en ligne)
protocolInformation	Attribut complémentaire à *presentationAddress* pour définir le protocole à utiliser
registeredAddress	Adresse postale pour des envois de courriers recommandés et de colis
seeAlso	DN d'objets complémentaires
serialNumber	Numéro de série de l'objet
sn	Nom de famille de la personne (*surname*)
st	Etat ou région (*state*)
street	Nom de la rue et assimiilé (boulevard, ...)
telephoneNumber	Numéro de téléphone
telexNumber	Numéro de télex
title	Titre de la personne (différent de fonction)
uid	Identifiant unique de l'objet
userCertificate	Certificat de l'utilisateur
userPassword	Mot de passe de l'utilisateur

Voici une petite liste non exhaustive des principaux attributs opérationnels définis par le standard LDAPv3 :

Attribut	Description
attributeTypes	Liste des attributs de l'annuaire. Cet opérateur opérationnel fait partie du schéma de l'annuaire décrit par l'objet *subschema*
altServers	Liste de serveurs LDAP alternatifs en cas de défaillance de celui-ci
createTimestamp	Contient la date de création d'un objet, et est ainsi présent dans tout objet. Son occurrence est unique et il ne peut être modifiée
creatorsName	Contient le DN de l'objet ayant servi à la création de l'objet, et est ainsi présent dans tout objet. Son occurrence est unique et il ne peut être modifié par un utilisateur
matchingRules	Contient l'ensemble des règles de comparaison. Cet opérateur opérationnel fait partie du schéma de l'annuaire décrit par l'objet *subschema*
matchingRuleUse	Contient l'ensemble des attributs utilisant chaque règle de comparaison. Cet opérateur opérationnel fait partie du schéma de l'annuaire décrit par l'objet *subschema*
modifiersName	Contient le DN de l'objet utilisé pour s'identifier lors de la modification. Il est présent dans tous les objets modifiés par la commande *modify*. Son occurrence est unique et il ne peut être modifié par un utilisateur
modifyTimestamp	Contient la date de la dernière modification de l'objet. Il est présent dans tous les objets modifiés par la commande *modify*. Son occurrence est unique et il ne peut être modifié par un utilisateur
namingContexts	Contient l'ensemble des contextes supportés par le serveur. Son occurrence est unique et il ne peut être modifié par un utilisateur
objectClasses	Contient l'ensemble des classes d'objets. Cet opérateur opérationnel fait partie du schéma de l'annuaire décrit par l'objet *subschema*
subschemaSubentry	Contient le DN de l'objet contenant le schéma de l'annuaire (*subschema*)
supportedControl	Contient l'ensemble des OID des contrôles supplémentaires ajoutés à l'annuaire

supportedExtensions	Contient l'ensemble des OID des extensions supplémentaires (fonctions utilisateurs) ajoutés à l'annuaire
supportedLDAPVersion	Contient les versions du protocole LDAP gérées par le serveur
supportedSASLMechanisms	Contient la liste des mécanismes SASL supportés par l'annuaire LDAP

2.5.1.4 Les classes d'objets

Par analogie avec la terminologie objet on parle de classe d'objet pour désigner la structure d'un objet, c'est-à-dire l'ensemble des attributs qu'il doit comporter. De cette façon on dira qu'un objet est une "instanciation" de la classe d'objet, c'est-à-dire un ensemble d'attributs avec des valeurs particulières.
Une classe d'objet est ainsi composée d'un ensemble d'attributs obligatoires (devant obligatoirement être renseignés dans les objets qui en découlent) et éventuellement des attributs facultatifs.
On distingue plusieurs types de classes d'objets:

- Les **classes abstraites** sont des classes non instanciables. Il s'agit de classes pouvant être dérivées, c'est-à-dire dont d'autres classes peuvent hériter. La classe d'objet de plus haut niveau étant la classe top dont toute classe d'objet dérive
- Les **classes structurelles** sont des classes instanciables. Il est donc possible d'avoir des objets
- Les **classes auxiliaires** sont des classes permettant d'ajouter des attributs facultatifs à des classes structurelles.

Une des caractéristiques intéressantes des classes d'objets LDAP est la possibilité d'utiliser l'héritage.

Ainsi la classe de plus haut niveau est la classe **top** dont toutes les classes d'objets dérivent. Avec LDAP seul l'héritage simple est autorisé (donc pas d'héritage multiple), c'est-à-dire qu'une classe ne peut dériver que d'une seule classe, mais qu'une classe peut avoir plusieurs filles. :

Les attributs sont caractérisés par :

- leur nom unique
- un Object Identifier (OID) qui permet de les identifier de façon unique
- une syntaxe et des règles de comparaison

- un indicateur d'usage
- un format ou une limite de taille

Il s'agit d'utiliser une série de paires clé/valeur permettant de repérer une entrée de manière unique. Voici une série de clés généralement utilisées :

- **uid** (userid) : identifiant unique obligatoire
- **cn** (common name) : nom de la personne
- **givenname** : prénom de la personne
- **sn** (surname) : nom usuel de la personne
- **o** (organization) : entreprise de la personne
- **ou** (organization unit) : service de l'entreprise dans laquelle la personne travaille
- **mail** : adresse de courrier électronique de la personne

2.5.2 Le modèle de nommage

Le modèle de nommage (aussi appelé modèle de désignation) a pour but de définir la façon selon laquelle les objets de l'annuaire sont nommés et classés.
Ainsi les objets LDAP sont classés hiérarchiquement et possèdent un espace de nom homogène. Cela signifie que d'un annuaire à un autre, un objet de la même classe possèdera le même nom afin de garantir une compatibilité (on parle d'interopérabilité) entre les annuaires.

2.5.2.1 L'arborescence d'informations (DIT)

LDAP présente les informations sous forme d'une arborescence d'informations hiérarchiques appelée **DIT** (Directory Information Tree), dans laquelle les informations, appelées entrées (ou encore **DSE**, Directory Service Entry), sont représentées sous forme de branches.

Une branche située à la racine d'une ramification est appelée racine ou suffixe (en anglais root entry).
Chaque entrée de l'annuaire LDAP contient à un objet abstrait ou réel (par exemple une personne, un objet matériel, des paramètres, ...). Ceci signifie que chaque

noeud de l'arbre correspond à un objet pouvant appartenir à n'importe quelle classe d'objets. Les classes d'objets peuvent donc être utilisées comme à n'importe quel niveau de la hiérarchie et même à la racine de l'arbre.
Il existe une entrée particulière de l'annuaire appelée **rootDSE** (root Directory Specific Entry) contenant la description de l'arbre.

2.5.2.2 La désignation des entrées

La norme LDAPv3 permet de désigner un objet de deux façons :

- grâce à son nom relatif (RDN - *Relative Distinguished Name*)
- grâce à son nom absolu (DN - *Distinguished Name*)

Le nom relatif (RDN) est composé d'une (ou plusieurs) paire(s) clé/valeur (attribut). Ainsi, un RDN sera de la forme cn=UMLV ou bien c=fr.
Un RDN doit respecter certains critères :

- Un objet ne doit posséder qu'un et un seul RDN
- Le RDN doit être un nom unique dans la branche de l'objet (à un même niveau)
- Un RDN peut être composé d'un ensemble d'attributs. Le RDN est alors dit *multivalué* (par exemple *cn=UMLV+sn=Ingenieurs2000*)

Ainsi il est conseillé de faire en sorte que l'attribut servant de RDN soit obligatoire.

Le DN (*Distinguished Name*) d'un objet est un moyen d'identifier de façon unique un objet dans la hiérarchie. Un DN se construit en prenant le nom relatif de l'élément (RDN - *Relative Distinguished Name*), et en lui ajoutant l'ensemble des noms relatifs des entrées parentes. Le DN d'un élément est donc la concaténation de l'ensemble des RDN de ses ascendants hiérarchiques.

Ainsi une entrée indexée par un nom absolu (DN, *distinguished name*) peut être identifiée de manière unique dans l'arborescence. Le nom absolu (DN) d'un objet ne comportant aucune information relative à la localisation de l'annuaire lui-même, la norme LDAPv3 recommande de le compléter par l'adresse DNS de l'annuaire.

2.5.3 Le modèle fonctionnel

LDAP fournit, à travers le modèle fonctionnel, un ensemble de neuf fonctions (appelées parfois procédures ou opérations) de base pour effectuer des requêtes sur les données afin de rechercher, modifier, effacer des entrées dans les répertoires.
Les opérations sont généralement classées en trois catégories :

- **les fonctions d'interrogation**: il s'agit des opérations permettant de rechercher ou comparer des entrées de l'annuaire (recherche, comparaison)
- **les fonctions de mise à jour**: il s'agit des opérations permettant de modifier des entrées de l'annuaire (ajout, suppression, modification, renommage)
- **les fonctions de session**: il s'agit des opérations permettant d'ouvrir une session (s'identifier), de la fermer ainsi que d'annuler une requête

Voici la liste des principales opérations que LDAP peut effectuer:

Opération	Description
Abandon	Abandonne l'opération précédemment envoyée au serveur
Add	Ajoute une entrée au répertoire
Bind	Initie une nouvelle session sur le serveur LDAP
Compare	Compare les entrées d'un répertoire selon des critères
Delete	Supprime une entrée d'un répertoire
Extended	Effectue des opérations étendues
Rename	Modifie le nom d'une entrée
Search	Recherche des entrées d'un répertoire
Unbind	Termine une session sur le serveur LDAP

Les commandes *search* et *compare* se font sous la forme d'une requête composée de 8 paramètres:

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 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

Le scope définit la profondeur de la recherche dans le **DIT**. Il peut prendre différentes valeurs selon la portée de la recherche souhaitée :

- **base** : recherche dans le niveau courant
- **one-level** : recherche uniquement dans le niveau inférieur au niveau courant
- **subtree**: recherche dans tout le sous-arbre à partir du niveau courant

Il n'existe pas de fonction *read* dans LDAP. Cette fonction est simulée par la fonction *search* avec un *search scope* égal à base.
Le filtre de recherche s'exprime suivant une syntaxe spécifique dont la forme générale est : (< operator(< search operation)(< search operation)...))
Ce filtre décrit une ou plusieurs conditions exprimées sous forme d'expressions régulières sensées désigner un ou plusieurs objets de l'annuaire, sur lesquels on veut appliquer l'opération voulue. Le suivant récapitule les opérateurs de recherche disponibles :

Filtre	Syntaxe	Interprétation
Approximation	(sn~=UMLV)	nom dont l'orthographe est voisine de UMLV
Egalité	(sn=UMLV)	vaut exactement UMLV
Comparaison	(sn>UMLV) , <= , >= , <	noms situés alphabétiquement après UMLV
Présence	(sn=*)	toutes les entrées ayant un attribut sn
Sous-chaîne	(sn=UM*), (sn=*ML*), (sn=UM*V)	expressions régulières sur les chaînes
ET	(&(sn=UMLV) (ou=Ingenieurs2000))	toutes les entrées dont le nom est UMLV et du service Ingenieurs2000
OU	(|(ou=IGM) (ou=UMLV))	toutes les entrées dont le service est IGM ou UMLV
Négation	(!(tel=*))	toutes les entrées sans attribut téléphone

Lors de la connexion au serveur (*bind*), ce dernier demande une authentification. Le client doit alors fournir un *DN* et le *mot de passe* correspondant, celui-ci transitant en clair. Pour sécuriser les transactions, LDAPv3 fournit la possibilité d'utiliser du chiffrement (SSL ou TLS) et le mécanisme *Simple Authentification and Security Layer* (SASL) procurant des outils d'authentification plus élaborés à base de clés. Une fois connecté, le client peut envoyer autant de commandes qu'il souhaite jusqu'à ce qu'il ferme la session (*unbind*).

Chaque commande se voit attribuer un *numéro de séquence,* qui permet au client de reconnaître les réponses lorsque celles-ci sont multiples - ce qui peut parfois arriver lors d'une recherche simple qui peut renvoyer jusqu'à plusieurs milliers d'entrées. A

chaque opération, le serveur renvoie également un *acquittement* pour indiquer que sa tâche est terminée ou qu'il y a une erreur.

2.5.3.1 Le format d'échange de données LDIF

LDAP fournit un format d'échange (LDIF, Lightweight Data Interchange Format) permettant d'importer et d'exporter les données d'un annuaire avec un simple fichier texte. La majorité des serveurs LDAP supportent ce format, ce qui permet une grande interopérabilité entre eux.

La syntaxe de ce format est la suivante:

```
[<id>]
dn: <distinguished name>
<attribut> : <valeur>
<attribut> : <valeur>
...
```

Dans ce fichier id est facultatif, il s'agit d'un entier positif permettant d'identifier l'entrée dans la base de données.

- chaque nouvelle entrée doit être séparée de la définition de l'entrée précédente à l'aide d'un saut de ligne (ligne vide)
- Il est possible de définir un attribut sur plusieurs lignes en commençant les lignes suivantes par un espace ou une tabulation
- Il est possible de définir plusieurs valeurs pour un attribut en répétant la chaîne nom:valeur sur des lignes séparées
- lorsque la valeur contient un caractère spécial (non imprimable, un espace ou :), l'attribut doit être suivi de :: puis de la valeur encodée en base64.

2.5.4 Le modèle de sécurité

L'annuaire doit pouvoir être protégé contre des intrusions intempestives, et ce, de manière efficace. De plus, chaque acteur, suivant ses droits ne doit pouvoir effectuer que certaines actions.

Les aspects de 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.

La tâche primordiale réside dans l'établissement des règles d'accès aux données. Il faut déterminer pour chaque attribut quel est son niveau de confidentialité et quel utilisateur ou quelle application pourra y accéder en lecture ou en écriture.

Les mécanismes qui peuvent être mis en œuvre :

- L'authentification
- Les signatures électroniques
- La cryptographie
- Le filtrage réseau
- Les règles d'accès aux données (ACL ou access control list)
- L'audit des journaux

Les ACL permettent de décrire les habilitations de tout utilisateur référencé dans l'annuaire sur les autres objets de l'annuaire.

2.5.4.1 Authentification

LDAP propose plusieurs choix d'authentification :

- Anonymous authentification : correspond à un accès au serveur sans authentification, qui permet uniquement de consulter les données accessibles en lecture pour tous.
- Root DN Authentification : Utilisateur privilégié. Il a accès en modification à toutes les données.
- Mot de passe + SSL : La session entre le serveur et le client est chiffrée et le mot de passe ne transite plus en clair
- Simple Authentification and Security Layer : permet de faire des mécanismes d'authentification plus élaborés à base de clés
- Certificats sur SSL : Echange de certificats (clefs publiques/privées)

2.5.4.2 Définition des droits d'accès

Il s'agit de définir les droits d'accès des utilisateurs sur les ressources de l'annuaire (objets et attributs).

Préciser à qui appartiennent chaque attribut et chaque classe d'objet, ainsi que les droits d'accès des acteurs spécifiés. Ces droits comprennent essentiellement la lecture, la modification, la création, la suppression et la recherche.
Pour définir ces droits il faut commencer par lister les actions possibles :

- Rechercher et lister des données
- Comparer des données
- Modifier un objet
- Supprimer un objet
- Ajouter un objet
- Renommer le DN d'un objet

2.5.4.3 Protection réseau

Toute la panoplie d'outils de sécurité est à la disposition de l'administrateur pour sécuriser les accès réseau au service et la confidentialité des transactions. LDAP supporte les protocoles SSL et TLS pour chiffrer les données qui transitent sur le réseau.

2.5.5 Le modèle de réplication

La réplication est très importante pour plusieurs raisons. La première est la volonté d'assurer une disponibilité maximale des données gérées par l'annuaire. La seconde est l'optimisation des performances.

Cette optimisation d'accès à l'annuaire peut se faire de la façon suivante : plusieurs serveurs dédiés à la lecture et un seul dédié à l'écriture.

De plus, la réplication d'un serveur sur plusieurs serveurs peut pallier à :

- Une coupure du réseau

- Surcharge du service
- Une panne de l'un des serveurs

Une stratégie consiste à avoir un seul serveur maître sur le site principal qui centralise toutes les mises à jour. Cette stratégie est simple de mise en place. Le serveur maître va se charger des répliquer ses informations de manière régulière sur les serveurs dédiés à la lecture.

Toutefois si ce type de stratégie est simple à mettre en œuvre, il y a deux inconvénients :

1. Si le serveur maître tombe en panne, les mises à jour ne peuvent plus s'effectuer.

- Mise en place d'un doublon non accessible pour pallier à une éventuelle panne du serveur maître

2. Si le serveur maître est situé sur un site distant qui n'est pas relié en permanence avec les postes clients, la mise à jour ne pourra pas se faire à tout moment

- Ce point n'aura pas de répercussion étant donné que le serveur maître est à proximité et en réseau avec les postes clients.

Les possibilités de réplication :

- L'arbre entier ou seulement un sous arbre
- Une partie des entrées et de leurs attributs qu'on aura spécifiés via un filtre

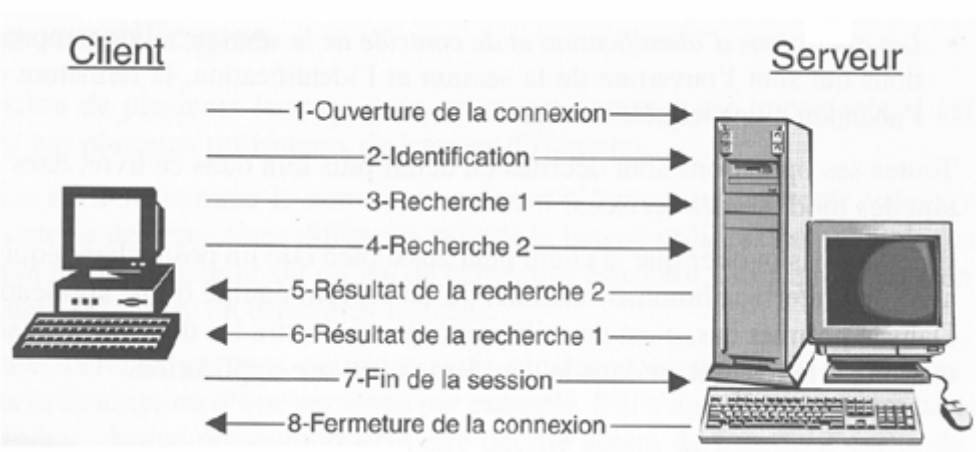
La synchronisation des serveurs peut se faire de façon totale ou incrémentale. La réplication se fait en temps réel ou à heure fixe.

2.6 Communication LDAP Client-Serveur

Le dialogue entre un client et un serveur LDAP est basé sur un protocole de type client-serveur. Sa particularité est de reposer sur un mécanisme de questions et de réponses sous forme de messages traités par le serveur de façon synchrone ou asynchrone.

Dans le cas de communication synchrone le client attend la réponse avant de transmettre une nouvelle requête.

Dans le cas de communications asynchrones, un numéro de contexte est associé à chaque requête, permettant au serveur d'envoyer ses réponses sans contrainte d'ordre.

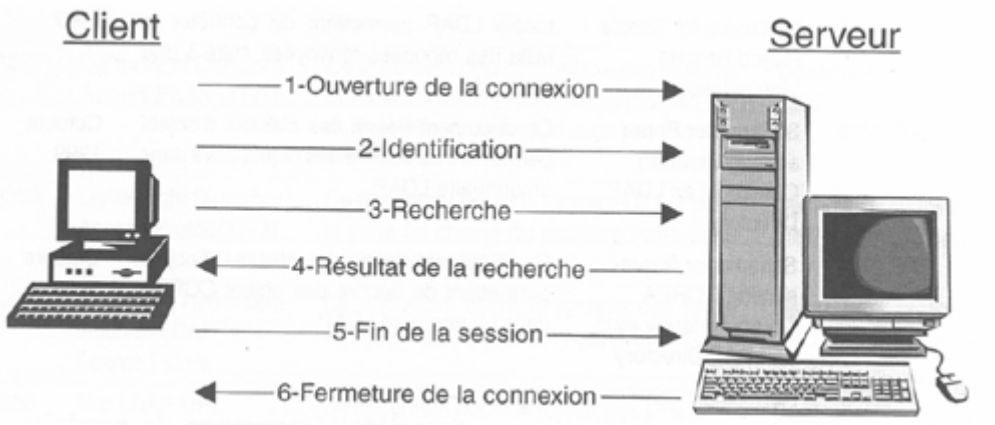


Dans ce cas, on observe que la réponse 2 arrive avant la réponse 1. Ceci permet donc d'optimiser le serveur si la réponse 2 est plus rapide à obtenir que la réponse 1.

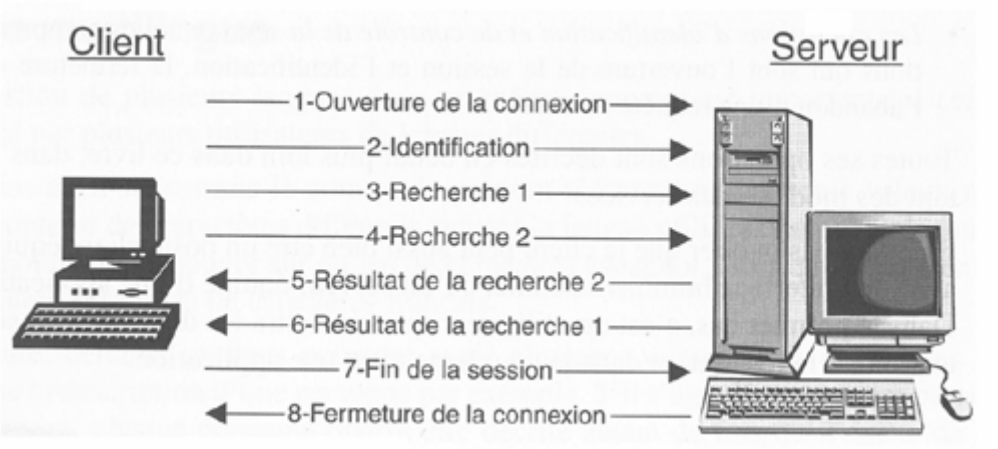
2.7 Communication LDAP Client-Serveur

Le dialogue entre un client et un serveur LDAP est basé sur un protocole de type client-serveur. Sa particularité est de reposer sur un mécanisme de questions et de réponses sous forme de messages traités par le serveur de façon synchrone ou asynchrone.

Dans le cas de communication synchrone le client attend la réponse avant de transmettre une nouvelle requête.



Dans le cas de communications asynchrones, un numéro de contexte est associé à chaque requête, permettant au serveur d'envoyer ses réponses sans contrainte d'ordre.



Dans ce cas, on observe que la réponse 2 arrive avant la réponse 1. Ceci permet donc d'optimiser le serveur si la réponse 2 est plus rapide à obtenir que la réponse 1.

Implémentation de la norme LDAP

Nous présenterons ici diverses implémentations de la norme LDAP. Tout d'abord, cela correspond à introduire OpenLDAP, un projet libre de serveur d'annuaire conforme à la norme LDAP 3. Ceci fait, il sera intéressant d'étudier un exemple de cas d'utilisation de LDAP. Enfin, nous présenterons Active Directory, le standard de Microsoft.

2.8 OpenLDAP

OpenLDAP (http://www.OpenLDAP.org) est un projet libre de serveur d'annuaire conforme à la norme LDAP 3. Ce serveur, dérivé de l'implémentation mise au point par l'université du Michigan.
OpenLDAP est composé des éléments suivants :

- Le serveur LDAP : slapd
- La passerelle LDAP vers X500 : ldapd
- Le serveur de réplication : slurpd
- Des outils d'administration

OpenLDAP est un logiciel libre, au sens de la Free Software Foundation. Cela signifie qu'il respecte les quatre libertés fondamentales d'un logiciel libre: liberté d'exécution, liberté d'étude, liberté de modification et liberté de redistribution.

En revanche ce n'est pas un logiciel copyleft. C'est à dire qu'il n'impose pas sa licence d'utilisation aux logiciels qui dérivent de lui. Il peut donc être privatisé. Ce qui ne l'empêche pas d'être compatible avec la licence GPL. Le code d'OpenLDAP peut donc être intégré dans un logiciel sous GPL.

2.8.1 Installation

Comme pour la plupart des logiciels sous Linux, OpenLDAP peut-être installé de différentes façons :

- sous forme de fichiers RPM
- sous forme de packages Debian ou autres

- en compilant les sources

2.8.1.1 Installation de OpenLDAP sous forme de RPM

Pour installer le serveur LDAP sous forme de RPM, il s'agit dans un premier temps de récupérer le package (nommé OpenLDAP-version.rpm où version représente la version actuelle de OpenLDAP), puis d'exécuter la commande suivante :

```
rpm -ivh OpenLDAP-1.2.9-5mdk.i586.rpm
```

2.8.1.2 Installation de OpenLDAP sous forme de packages

Pour installer le serveur LDAP sous forme de package, il faut taper la commande suivante :

```
apt-get install slapd ldap-utils
```

2.8.1.3 Installation de OpenLDAP à partir des sources

Si vous avez choisi le mode d'installation à partir des sources, il vous faut donc suivre ces étapes : Décompression de l'archive, en remplaçant xxx, yyyy et zzzz par les bonnes valeurs pour l'archive que vous avez téléchargée :

```
tar xvzf OpenLDAP-xxx-yyyy.tgz
cd OpenLDAP-zzzz
```

Configuration de l'installation :

```
./configure
```

Il ne reste plus qu'à compiler les sources :

```
make depend $ make
```

Lancer les tests pour vérifier la bonne compilation du programme :

```
make tests
```

Ensuite il faut lancer l'installation en root :

```
su -c "make install"
```

2.8.2 Répertoires de OpenLDAP

L'installation génère un certain nombre de scripts de configuration et va créer les répertoires suivants :

- ***/etc/OpenLDAP*** : répertoire des fichiers de configuration
- ***/var/lib/ldap*** : répertoire par défaut où va être stocké l'annuaire
- ***/usr/share/OpenLDAP*** : répertoire contenant les documentations et les outils pour migrer par exemple un système NIS (yellow page) existant dans l'annuaire LDAP

Les traditionnelles pages de manuel et les commandes LDAP sont respectivement installées dans ***/usr/man*** et ***/usr/bin***.

2.8.3 Configuration

La configuration d'un serveur LDAP est la phase préliminaire à toute mise en oeuvre de celui-ci. Elle est bien entendue spécifique à l'outil que vous utilisez. Dans le cas de OpenLDAP, cela consiste à éditer un fichier : slapd.conf.

Le fichier slapd.conf est constitué de trois types d'informations de configuration : global, spécifique au backend, et spécifique à la base de données. L'information globale est spécifiée en premier, suivie par l'information associée à un type de backend particulier, qui est elle même suivie par l'information associée avec une instance de base de données particulière.

Les lignes blanches et les commentaires commençant par le caractère « # » sont ignorés. Si une ligne commence avec un espace, elle est considérée comme la continuation de la ligne précédente.

2.8.3.1 Gestion des schémas

Rappelons qu'un objet LDAP est décrit par des attributs et des classes d'objets. Un schéma regroupe les attributs et les classes d'objet que pourront posséder les objets de l'annuaire. Il précise pour chaque attribut et chaque classe les contraintes, les héritages, les syntaxes, les règles de comparaison,…

Dans le cas de l'outil OpenLDAP, les schémas sont des fichiers textes. Les schémas nécessaires au bon fonctionnement du serveur sont inclus dans le fichier slapd.conf
La ligne suivante :

```
include /usr/local/etc/OpenLDAP/schema/core.schema
```

va permettre de spécifier quel schéma l'annuaire doit mettre en oeuvre. Ici ce fichier décrit le schéma de base de tous les annuaires LDAP. Il contient les définitions des attributs et classes d'objets standards. Ce schéma est obligatoire ; c'est le minimum attendu par un serveur LDAP.
Voici un extrait du fichier core.schema avec la classe 'person' :

```
ObjectClass (2.5.6.6 NAME 'person'
DESC 'RFC2256 : a person'
SUP top structural
MUST (sn, $ cn )
MAY ( userPasswd $ telephoneNumber $ description $ seeAlso ))
```

Légende:

MUST correspond aux attributs obligatoires et MAY à ceux facultatifs
ObjectClass est le nom de la classe qui descend elle même de la classe top
sn correspond à nom
cn correspond à prénom + nom

Pour créer un annuaire contenant des fiches de personnes, il sera nécessaire de rajouter plusieurs schémas complémentaires tels que :

```
include /usr/local/etc/OpenLDAP/schema/inetorgperson.schema
include /usr/local/etc/OpenLDAP/schema/cosine.schema
```

Ces schémas permettent d'avoir une description plus intéressante des objets de l'annuaire.

2.8.3.2 Gestion du serveur

Le serveur, lors de son démarrage, essaye d'écrire dans deux fichiers particuliers ; s'il échoue dans l'écriture, ceci n'empêche pas son fonctionnement. Mais il vaut mieux que ces fichiers soient présents en cas de problèmes ultérieurs.
Les lignes suivantes :

```
pidfile /var/run/slapd.pid
argsfile /var/run/slapd.args
```

Le fichier slapd.pid contient le numéro du premier processus UNIX sous lequel le serveur tourne.

Le fichier slapd.args contient la liste des arguments avec lesquels a été lancé le serveur.

2.8.3.3 Gestion de la base de donnée

La gestion de la base de données va permettre de préciser plusieurs choses :

- Le nom (suffixe) de la base de données
- L'identité (DN) du gestionnaire de la base
- L'endroit où seront stockés les différents fichiers représentant les données de l'annuaire

2.8.3.4 Le suffixe de la base de données

C'est en quelque sorte l'identifiant général de la base de données. Toutes les entrées de la base contiendront ce suffixe.
Il est définit ainsi : **dc=my-domain ,dc=com**

2.8.3.5 Le gestionnaire de la base

C'est une entrée spéciale de la base. Elle peut être virtuelle. Elle est gérée par la ligne **rootdn**. La solution la plus simple consiste à utiliser une forme en fonction du choix du suffixe.

```
rootdn « cn=Manager, dc=my-domain , dc=com »
```

Le gestionnaire de la base doit se connecter à l'aide d'un mot de passe ; celui-ci est décrit par la ligne suivante

```
rootpw secret
```

Le répertoire de stockage des données de l'annuaire est indiqué par la line suivante :

2.8.3.6 Gestion des contrôles d'accès

L'accès aux entrées et attributs slapd est contrôlé par la directive de configuration d'accès. La forme générale d'une ligne d'accès est la suivante :

```
<access directive ::= access to <what>
[by <who> <access><control>]
<what> ::=* | [dn.<target style>]=<regex>]
[filter=<ldapfilter>] [attrs=<attrlist>]
<who> ::= [* | anonymous | users | self | [dn.<subject style>]
```

Où le <what> définit les entrées et/ou les attributs sur lesquelles les règles s'appliquent, le <who> définit quelles identités ont accès, et le <acces> définit le type d'accès.

Exemples :

 # Autorise la visualisation d'une entrée comprenant l'attribut organizationalStatus avec comme valeur parti uniquement à l'admin de l'annuaire et personne d'autre

```
access to attr=entry filter=(organizationalStatus=parti)
by dn="cn=Manager,dc=my-domain.com" read
by dn="cn=Manager,dc=my-domain.com" write
by * none
```

 # Autorise la consultation de toutes les entrées à tout le monde

```
access to * by * read
```

Extrait du fichier slapd.conf :

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include /usr/local/etc/OpenLDAP/schema/core.schema
include /usr/local/etc/OpenLDAP/schema/inetorgperson.schema
include /usr/local/etc/OpenLDAP/schema/cosine.schema
include /usr/local/etc/OpenLDAP/schema/nis.schema
pidfile /var/run/slapd.pid
argsfile /var/run/slapd.args
access to attr=entry filter=(organizationalStatus=parti)
by dn="cn=Manager,dc=my-domain.com" read
```

```
by dn="cn=Manager,dc=my-domain.com" write
by * none
access to * by * read
rootdn can always write!
database bdb
suffix "dc=my-domain ,dc=com"
rootdn "cn=Manager, dc=my-domain, dc=com"
rootpw secret
directory /var/db/OpenLDAP-data
index default pres,eq
index objectClass
index cn, s ,mail eq, sub, approx
```

Le fichier ldap.conf va permettre de définir l'URL vers laquelle on peut accéder au serveur LDAP. Il est important d'y spécifier également l'adresse IP de la machine ainsi que le DN de l'annuaire et le port de connexion.

Extrait du fichier ldap.conf :

```
#
# LDAP Defaults
#
# See ldap.conf(5) for details
# This file should be world readable but not world writable.
BASE dc=my-domain, dc=com
URI ldap://xxx.xxx.xxx.xxx/dc=my-domain, dc=com ldap://localhost:389
#SIZELIMIT 12
#TIMELIMIT 15
#DEREF never
```

2.8.4 Utilisation

Il existe différents moyens de manipuler des données de l'annuaire :

- En ligne de commande
- A l'aide du client LDAP Browser, un client Java Swing

Nous allons détailler la manipulation des données de l'annuaire en ligne de commande.
Les commandes LDAP existantes sont les suivantes :

- ldapadd,
- ldapdelete,
- ldapsearch,
- ldapcompare,
- ldapmodify,
- ldappasswd,
- ldapmodrdn.

2.8.4.1 Démarrage du service slapd

Nous démarrerons le service en mode « debug » numéro 1. Cela signifie que toutes les informations telles que les connexions ou les requêtes coté serveur s'affichent dans la fenêtre de son exécution.

```
./slapd -d 1
```

2.8.4.2 Ajouter un enregistrement

Pour ajouter des données au serveur LDAP il faut fournir un fichier au format LDIF, le format est un format texte facilement lisible au contraire du format interne de l'annuaire. Le format d'un fichier *.ldif est le suivant :

```
Dn : description du distinguished name
ObjectClass : classe d'objet d'origine
…
objectClass : classe d'objet d'arrivée
type attribut : valeur
```

Exemple :

Création du rootdn avec la syntaxe suivante dc=my-domain , dc=com représentant l'organisation root (o)

```
dn: dc=my-domain, dc=com
objectClass: dcObject
objectClass: organization
dc: my-domain
o: root
```

Création de l'objet SERVICE qui est de type OrganizationalUnit. Cette entité appartient à l'organisation root.

```
dn: ou=SERVICE, dc=my-domain, dc=com
objectClass: organizationalUnit
ou: SERVICE
```

Création de la fiche d'une personne qui appartient au SERVICE, ce même service appartenant à l'organisation root.

```
dn: cn=Henri LEBLANC, ou=PARIS, ou=SERVICE, dc=my-domain, dc=com
cn: Henri LEBLANC
objectClass: person
objectClass: inetOrgPerson
sn: leblanc
mail: leblanc@root.com
```

telephoneNumber:0890048015

REMARQUE : Chaque enregistrement dans le fichier est séparé du précédent et du suivant par une ligne vierge. Les espaces sont pris en compte.

ATTENTION : Il est important qu'il n'y ait aucun espace en fin de ligne.

Pour ajouter l'enregistrement on utilisera la syntaxe suivante :

```
ldapadd -D « <description du DN de l'administrateurs> » -w -f <nom du fichier>.ldif
```

Exemple :

On souhaite rajouter le fichier test.ldif crée ci-dessus

```
ldapadd –D « cn=Manager, dc=my-domain, dc=com » –w –f test.ldif
Enter LDAP password : secret
Adding new entry « dc=my-domain ,dc=com ».
Adding new entry « ou=SERVICE ,dc=my-domain, dc=com ».
Adding new entry « cn=Henri LEBLANC, ou=SERVICE, dc=my-domain ,dc=com
».
```

REMARQUE: Attention l'ajout d'un fichier de type *.ldif ne fonctionne qu'une seule fois : ceci est du qu'au second lancement du fichier, il va percevoir une redondance des objets crées et va annuler l'ajout même si le fichier comporte de nouvelles informations. Pour le réutiliser, il faudra vider la base auparavant, sinon utiliser un autre fichier de type *.ldif pour accroître l'annuaire

2.8.4.3 Rechercher un enregistrement

On utilisera la fonction ldapsearch. Pour visualiser tout l'annuaire on peut taper :

```
ldapsearch –b « dc=my-domain, dc=com » '(objectClass=*)'
```

2.8.4.4 Modifier un enregistrement

Ajouter un attribut à un enregistrement

Pour rajouter l'attribut facultatif location (**l**) à l'enregistrement **Henri LEBLANC.** On va créer un fichier *modif.ldif contenant :*

```
dn: cn= Henri LEBLANC, ou=PARIS ,ou=SERVICE, dc=my-domain, dc=com
add : l
title : extensionDOC
```

On tape ensuite :

```
ldapmodify –D "cn=Manager, dc=my-domain, dc=com » –w –f modif.ldif
Enter LDAP password : secret
Modifying entry « cn=Henri LEBLANC, ou=SERVICE, dc=my-domain ,dc=com
».
```

Modifier un attribut existant

On va modifier l'attribut titre (**title**) à l'enregistrement **Henri LEBLANC.** On va créer un fichier modif.ldif contenant :

```
dn: cn=Henri LEBLANC, ou=PARIS ,ou=SERVICE, dc=my-domain, dc=com
changetype: modify
replace: telephoneNumber
```

```
telephoneNumber: 0984532527
```

On tape ensuite :

```
ldapmodify -D "cn=Manager, dc=my-domain, dc=com » -w -f modif.ldif
Enter LDAP password : secret
Modifying entry « cn=Henri LEBLANC, ou=SERVICE, dc=my-domain ,dc=com
».
```

Supprimer un attribut

On veut supprimer l'attribut location (l) à l'enregistrement Henri LEBLANC. On va créer un fichier modif.ldif contenant :

```
dn: cn=Henri LEBLANC, ou=PARIS ,ou=SERVICE, dc=my-domain, dc=com
delete : l
```

On tape ensuite la commande suivante :

```
ldapmodify -D "cn=Manager, dc=my-domain, dc=com » -w -f modif.ldif
Enter LDAP password : secret
Modifying entry « cn=Henri LEBLANC, ou=SERVICE, dc=my-domain ,dc=com
».
```

2.8.4.5 Supprimer un enregistrement

On veut supprimer l'entrée Henri LEBLANC :

```
ldapdelete -v -D « cn=Manager,dc=my-domain.com» -w « cn=Henri
LEBLANC,dc=my-domain.com »
```

2.8.5Avantages et Inconvénients

2.8.5.1 Avantages

Parmi les atouts d'OpenLDAP on distingue facilement ses atouts techniques :

- De nombreux backends. Ils permettent au serveur slapd d'être utilisé à de nombreux usages, (comme un proxy ou un meta annuaire par exemple).
- Des options de sécurité avancées. Le serveur slapd est compatible avec les protocoles SSL et SASL.
- De nombreuses extensions implémentées. Chaque nouvelle version amène son lot de nouveautés et d'extensions supplémentaires implémentées.

L'autre grand atout d'OpenLDAP qu'il ne faut pas négliger c'est sa qualité de logiciel libre. Évidemment son coût s'en trouve réduit, puisqu'il n'y a aucun coût de licence, ni à l'achat, ni à l'exploitation, et quelle que soit la quantité de données gérées. Le seul

coût est donc celui de son déploiement et de sa maintenance. En tant que logiciel libre, ses bugs sont corrigés très rapidement, en particulier les bugs de sécurité.

Mais le principal avantage de ce genre de logiciel reste l'indépendance qu'il assure vis à vis d'un fournisseur ou d'un prestataire, la liberté de le modifier (ou de le faire modifier) pour l'adapter à ses propres besoins, sans avoir à en référer à personne.

2.8.5.2 Inconvénients

Les principaux reproches que l'on faire à OpenLDAP sont d'ordres techniques. L'obligation de redémarrer le serveur à chaque changement de configuration est assez pénible. En effet le fichier de configuration n'est lu qu'au démarrage, et il contient les règles d'accès et le schéma. Ce qui nécessite l'arrêt puis le redémarrage après chaque modification d'une règle ou du schéma. Ceci n'est au fond pas très gênant dans la mesure où schéma et règle ne devraient pas être modifiés très souvent, et qu'un annuaire qui doit être toujours accessible devrait être répliqué.

L'autre faiblesse, qui peut se révéler plus gênante, concerne les RFCs optionnelles non implémentées, et dont certaines organisations ne peuvent pas se passer.

Enfin, le dernier petit reproche concerne la documentation. La documentation en ligne n'est pas la plus complète. Pour avoir accès à toutes les directives de configuration, il faut télécharger le logiciel pour consulter les pages de manuels. Éventuellement, certaines pages de la FAQ peuvent se révéler d'un grand secours.

Sources

Installation, Configuration, Avantages et Inconvénients :

- http ://www.commentcamarche.net/ldap/ldapinst.php3
- http ://mparienti.developpez.com/cours/openldap/ ?page=page_4
- http ://articles.mongueurs.net/magazines/linuxmag65.html
- http ://www-id.imag.fr/Laboratoire/Membres/Varrette_Sebastien/download/polys/Tutorial_LDAP_HTML/node5.html

Utilisation : http://www-igm.univ-mlv.fr/~dr/XPOSE2003/HERVE/

2.9 Active Directory

2.9.1 Présentation

Active Directory est le nom du service d'annuaire de Microsoft apparu dans le système d'exploitation Microsoft Windows Server 2000. Le service d'annuaire Active Directory est basé sur les standards TCP/IP : DNS, LDAP, Kerberos, etc.

Le service d'annuaire Active Directory doit être entendu au sens large, c'est-à-dire qu'Active Directory est un annuaire référençant les personnes (nom, prénom, numéro de téléphone, etc.) mais également toute sorte d'objet, dont les serveurs, les imprimantes, les applications, les bases de données, etc.

Caractéristiques d'Active directory

Active Directory permet de recenser toutes les informations concernant le réseau, que ce soient les utilisateurs, les machines ou les applications. Active Directory constitue ainsi le moyeu central de toute l'architecture réseau et a vocation à permettre à un utilisateur de retrouver et d'accéder à n'importe quelle ressource identifiée par ce service.

Active Directory est donc un outil destiné aux utilisateurs mais dans la mesure où il permet une représentation globale de l'ensemble des ressources et des droits associés il constitue également un outil d'administration et de gestion du réseau. Il fournit à ce titre des outils permettant de gérer la répartition de l'annuaire sur le réseau, sa duplication, la sécurisation et le partitionnement de l'annuaire de l'entreprise.

La structure d'Active Directory lui permet de gérer de façon centralisée des réseaux pouvant aller de quelques ordinateurs à des réseaux d'entreprises répartis sur de multiples sites.

Source

Introduction à Active Directory :
http://www.commentcamarche.net/activedirectory/active-directory-intro.php3

2.10 Application de LDAP : Authentification des utilisateurs

Name Service Switch (NSS) permet de se passer de nombreux fichiers de configuration comme /etc/passwd, /etc/group, /etc/hosts avec plusieurs données ou une base centralisée. Il a tout d'abord été développé par Sun Microsystem pour Solaris puis a été porté sur FreeBSD, Linux, HP-Unix, IRIX et AIX.

NSS est habituellement configuré grâce au fichier /etc/nsswitch.conf. Celui-ci liste les bases de données regroupant des informations tels que les groupes d'utilisateurs et leur mot de passe et d'autres sources pour obtenir davantage d'informations.

La technologie Pluggable Authentification Module est une création de Sun Microsystems supportée par les architectures Solaris, Linux, FreeBSD et NetBSD. Elle permet d'intégrer différents schéma d'authentification de bas niveau en se passer des schémas des logiciels habituels.
L'administrateur système peut alors définir une stratégie d'authentification sans devoir recompiler les programmes en nécessitant une. PAM permet de contrôler la manière dont les modules sont enfichés dans les programmes en modifiant un fichier de configuration.

C'est ici qu'intervient LDAP. On peut stocker dans un annuaire toutes les informations relatives à un utilisateur dans un réseau seulement NSS supporte simplement les données suivantes :

- Alias : les alias mail
- Ethernet : les adresses Ethernet
- Groupes : des groupes d'utilisateurs
- Hôtes : les noms d'hôtes et leurs adresses

Lorsque celui-ci se connecte sur une machine, il est authentifié avec PAM et, il aura les droits sur son compte ou aura un client mail proprement configuré pour consulter ses mails grâce à NSS.

Sources

Name Server Switch, Pluggable Authentication Modules et LDAP sur RedHat 6.0 : http ://jfgiraud.free.fr/programmation/ldapauth/vinitial/

Authentification des utilisateurs via LDAP : http://www-id.imag.fr/Laboratoire/Membres/Varrette_Sebastien/download/polys/Tutorial_LDAP_HTML/node8.html#SECTION00082000000000000000

Définition NSS : http ://en.wikipedia.org/wiki/Name_Service_Switch

3 Synthèse de la Technologie LDAP

3.1 Avantages

Les avantages de la technologie LDAP se résument sur les points suivants :

- ***Centralisation***

  Aujourd'hui, il existe de nombreuses applications capables d'interagir avec un même annuaire LDAP. On peut citer, par exemple, les systèmes d'authentification d'utilisateur Unix (*pam_ldap*) ou Windows (*Samba*), proxy (Squid), web (module Apache AuthLDAP), POP3, IMAP. Cette liste est bien évidemment pas exhaustive.

  Les utilisateurs de tous ces services ne s'identifient alors qu'avec un seul identifiant pour tous ces services.

- ***Fiabilité***

  Des mécanismes de réplication (en cours de standardisation) entre des annuaires maîtres et des serveurs miroirs permettent d'assurer une grande fiabilité au système.

- ***Sécurisation***

  Les annuaires supportent pour la plupart des mécanismes de chiffrage des connexions (SSL TLS).

- ***Support de nombreux environnements de développement***

  Des librairies pour accéder aux annuaires LDAP existent dans la plupart des langages tel que (une fois encore cette liste n'est pas exhaustive) :

  - C /C++
    - l'API d'OpenLDAP
    - Sun ONE Directory SDK for C
    - Netscape Directory SDK pour C
    - Innosoft LDAP Client Software Development Kit (ILC-SDK)
  - Java
    - Netscape Directory SDK pour Java
    - Sun ONE Directory SDK for Java

- Java Naming and Directory Interface (JNDI), de SUN : API Java multi-annuaires (NIS, DNS, LDAP,...)
- JLDAP : Classes LDAP Java, contribution de Novell pour OpenLDAP

- Perl
  - PerLDAP : librairie en C et Perl fournissant une API Perl d'accès à un annuaire LDAP
  - Net::LDAPapi : ancienne librairie Perl remplacée par PerLDAP
  - Perl-LDAP : librairie Perl avec API orientée objet
- PHP
  - CruLDAP
  - Extensions LDAP de PHP3 : API LDAP pour le langage de script PHP

3.2 Inconvénients

- ***Un langage d'interrogation pauvre***

  Comparé à SQL le langage d'interrogation de LDAP est plutôt pauvre, mais c'est aussi ce qui rends le code aussi rapide.

3.3 LDAP contre d'autres technologies

Tableau comparatif LDAP / Base de Données

Critère	LDAP	Base de Données
Rapport lecture/écriture	optimisé en lecture	lecture/écriture
Extensibilité	facile (schéma LDAP)	difficile (via schéma entité-association)
Distribution des tables	inhérente	rare
Réplication	possible	possible
Modèle transactionnel	simple	avancé
Standard	oui	non (spécifique à un SGBD)

Tableau 1: Avantages/inconvénients de LDAP sur les bases de données

Tableau comparatif LDAP / NIS (Network Information Services)

Critère	LDAP	NIS
Port	spécifique (389/636 par défaut)	arbitraire (appel RPC)
Chiffrement des données	possible	impossible
Mécanisme de contrôle d'accès	oui	non
Distribution des tables	oui	non
Réplication	oui (réplication partielle possible)	oui (uniquement totale)
Sémantique des recherches	avancée	simple

Tableau 2: Avantages/inconvénients de LDAP sur NIS

Conclusion

Ce dossier sur l'étude du protocole LDAP nous a permis de comprendre que cette technologie est inévitable dans de nombreux domaines informatiques, comme par exemple dans les systèmes

- d'authentifications de personnes des Systèmes d'Informations
- ou encore de localisation de personnes ou de ressources.

Le principal point fort de cette technologie qui se dégage de cette étude est, de loin, la formidable souplesse de LDAP. En effet de nombreuses autres technologies sont capables d'interagir avec et, c'est ce qui permet de dire que LDAP va devenir la clé de voûte de la plupart des Systèmes d'Informations contemporains.

Ce dossier présente les principes fondamentaux du protocole LDAP qu'il faut connaître pour mettre en place une implémentation en oeuvre, mais au fur et à mesure que nous avancions dans notre étude, nous avons vite compris que ce dossier ne pouvait pas couvrir toutes les caractéristiques de LDAP.

Il serait particulièrement intéressant de proposer aux futures promotions d'Ingénieur 2000, d'étudier les autres aspects que propose LDAP comme par exemple les systèmes distribués LDAP, la sécurisation d'une architecture LDAP.

Glossaire

IETF : ***Internet Engineering Task Force***. Organisation préparant les principaux standards de l'Internet.

ISO : ***International Standard Organization***. Organisation internationale de standardisation, réunissant les organismes de normalisation de pas mal de pays dans le monde, et qui travaille dans tous les domaines.

Kerberos : protocole d'authentification réseau créé au MIT. Kerberos utilise un système de tickets au lieu de mots de passe en texte clair. Ce principe renforce la sécurité du système et empêche que des personnes non autorisées interceptent les mots de passe des utilisateurs.

OSI : ***Open System Interconnect***. Modèle en couches fournissant un cadre conceptuel et normatif aux échanges entre systèmes hétérogènes. Le modèle OSI comporte 7 couches : 1. Physique, 2. Liaison de données, 3. Réseau, 4. Transport, 5. Session, 6. Présentation, 7. Application.

PAM : ***Pluggable Athentication Modules***. Technologie de gestion des interfaces d'authentification pour les applications en nécessitant sur les systèmes Unix.

NSS : ***Name Server Switch***. Technologie de resolution de nom sur les systèmes Unix.

RFC : ***Request For Comment***. Référence auprès de la Communauté Internet, qui décrivent, spécifient, aident à la mise en œuvre, standardisent et débattent de la majorité des normes, standards, technologies et protocoles liés à Internet et aux réseaux en général.

RPM : ***Red Hat Package Manager*** est un système de gestion de paquets de logiciels utilisé sur certaines distributions GNU/Linux.

TCP/IP : ***Transmission Control Protocol / Internet Protocol***. Les deux protocoles de communication qui forment les fondements de l'Internet, spécifiés dans la RFC 793.