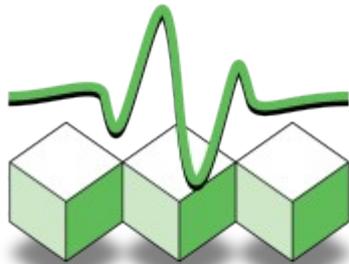
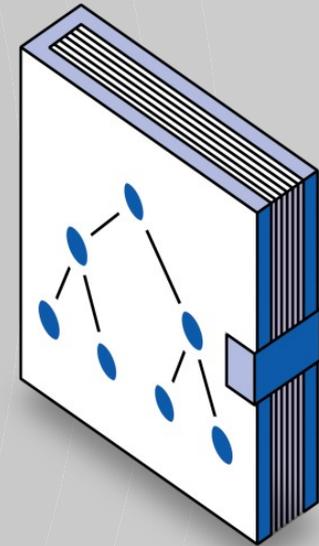


**LINAGORA**

*Formation*



Administration et sécurité



# Le protocole LDAP

**Auteurs :**

- Clément OUDOT, Raphaël OUAZANA et Sébastien BAHLOUL
- LINAGORA *Formation* : [formation@linagora.com](mailto:formation@linagora.com)

# Licence

## Paternité - Pas d'Utilisation Commerciale - Partage des Conditions Initiales à l'Identique 2.0 France

Vous êtes libres :

- de reproduire, distribuer et communiquer cette création au public,
- de modifier cette création.

Selon les conditions suivantes :

-  Paternité. Vous devez citer le nom de l'auteur original.
-  Pas d'Utilisation Commerciale. Vous n'avez pas le droit d'utiliser cette création à des fins commerciales.
-  Partage des Conditions Initiales à l'Identique. Si vous modifiez, transformez ou adaptez cette création, vous n'avez le droit de distribuer la création qui en résulte que sous un contrat identique à celui-ci.

À chaque réutilisation ou distribution, vous devez faire apparaître clairement aux autres les conditions contractuelles de mise à disposition de cette création.

Chacune de ces conditions peut être levée si vous obtenez l'autorisation du titulaire des droits.

Ce qui précède n'affecte en rien vos droits en tant qu'utilisateur (exceptions au droit d'auteur : copies réservées à l'usage privé du copiste, courtes citations, parodie...)

# Pourquoi LINAGORA met ce support sous licence Creative Commons

- Volonté de contribuer activement à l'essor du logiciel libre
- Promouvoir l'échange et favoriser l'émulation communautaire
- Assurer la pérennité de l'industrie logiciel libre et ne comptabiliser que la Valeur Ajoutée (le formateur)
- Partager le savoir et la connaissance à une vaste échelle

**LINAGORA croit au Libre !**

# Présentation du formateur

- Parcours du formateur

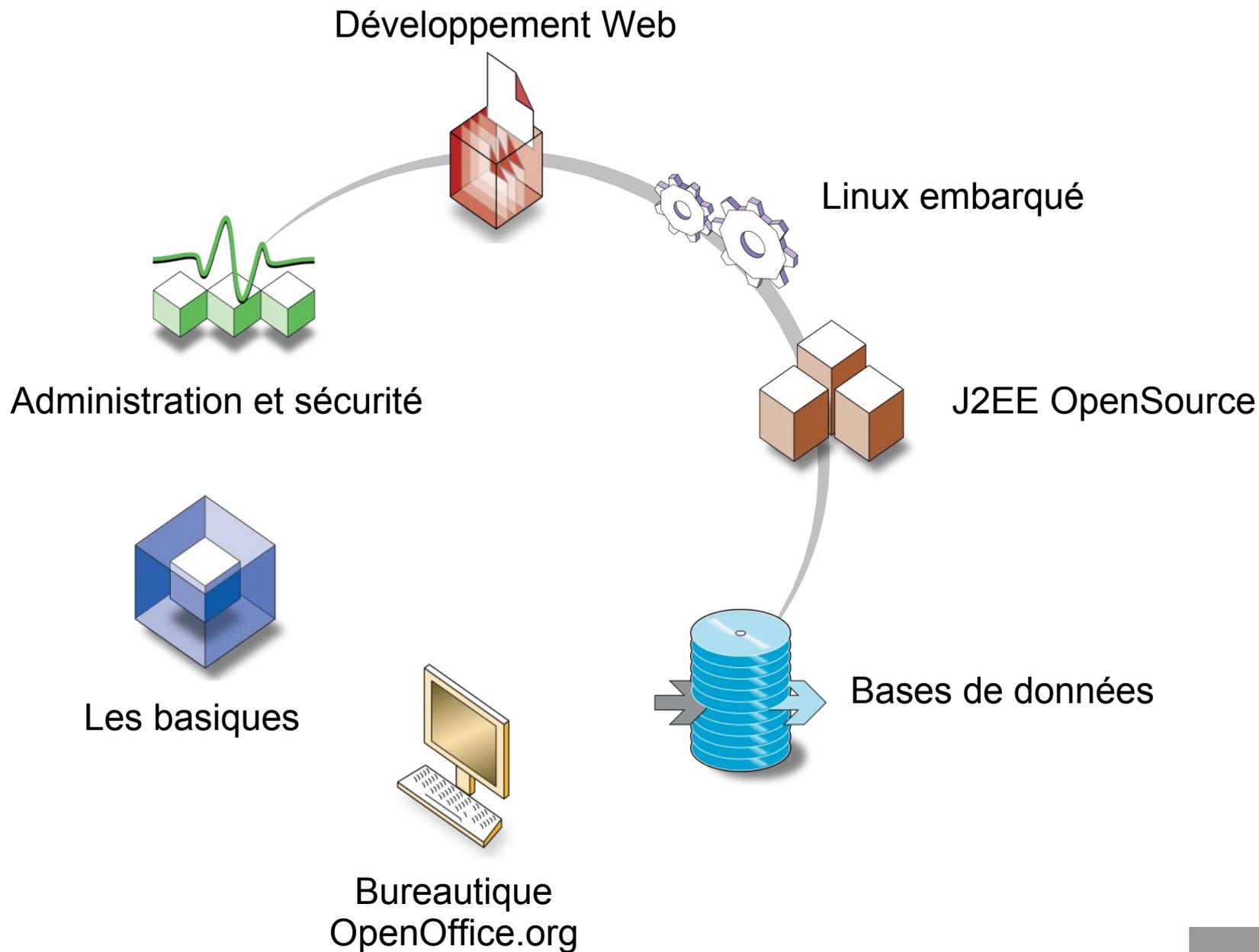
# LINAGORA, premier EOS

- Créateur des concepts SS2L (Société de Services en Logiciels Libres) et TM2L (Tierce Maintenance Logiciel Libre), LINAGORA se définit désormais comme un Éditeur Orienté Service (EOS).
- LINAGORA propose une **Open Source Software Assurance** (OSSA) sur 150 logiciels libres :
  - Prêts à l'industrialisation, sur une plate-forme unique : le [08000LINUX.com](http://08000LINUX.com).
  - Avec garantie de service contractuelle : en cas de bug, LINAGORA s'engage au résultat sur des délais de résolution.
  - Gestion de la feuille de route du logiciel pour le compte du client et s'engage au reversement des développements.
- LINAGORA apporte également son expertise sur toute une gamme de **services professionnels** et de **formations** au travers de **LINAGORA Formation**.

# LINAGORA Formation

- **7 années d'expérience**, au service des technologies libres et Open Source
- **40 modules** de formation répartis au travers de **7 filières**
- Un cadre agréable, au cœur de Paris
- Deux salles de formation climatisées pouvant accueillir jusqu'à 10 stagiaires.
- **2006 : Plus de 150 stages** effectués
- **2006 : Plus de 900 stagiaires**
- **Une satisfaction** moyenne client de **18/20**
- **Une note moyenne formateur** de **16,27/20**

# Filières de formations



# Organisation générale et planning

09h30 : début des cours

10h30 : pause du matin

10h45 : reprise des cours

12h00 : pause déjeuner

13h00 : reprise des cours

15h00 : pause de l'après-midi

15h15 : reprise des cours

17h30 : fin de journée

17h30 : libre discussion avec le formateur

Matin :

- Introduction
- Protocole
- Modèle d'information
- Modèle de nommage

Après-midi :

- Modèle fonctionnel
- Sécurité
- LDIF
- Annuaire LDAP

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes Cliquez pour ajouter un plan

# Définition d'un annuaire

- © Le Petit Robert :
  - Recueil publié annuellement et qui contient des renseignements remis à jour tous les ans
- Principales caractéristiques :
  - Ensemble de données
  - Beaucoup d'enregistrements de petite taille
  - Souvent lus, rarement mis à jour
  - Recherches simples

# Les annuaires électroniques

- Même principe que les annuaires papiers, mais avec les avantages du numérique :
  - Puissants : recherches multi-critères complexes
  - Dynamiques : mises à jour plus faciles
  - Souples : possibilité d'évolution de la structure des données
  - Sûrs : authentification, contrôles d'accès
  - Personnalisables : affichage en fonction de l'utilisateur

# Ce qu'un annuaire n'est pas

- Une base de données :
  - Rapport lecture/écriture beaucoup plus élevé
  - Structure plus facilement extensible
  - Diffusion à plus grande échelle (scalabilité)
  - Duplication des données plus simple
  - Respect des standards (LDAP)
- Un serveur de stockage :
  - Entrées de petite taille
  - Optimisé pour la lecture

# Exemples d'annuaires

- Annuaires papiers :
  - Annuaire téléphonique (botin)
  - Carnet d'adresses
  - Dictionnaire
- Annuaires électroniques :
  - Annuaire de personnes (pages blanches)
  - Annuaire d'entreprise (pages jaunes)
  - Annuaire d'authentification (certificats, PKI, ...)
  - DNS

# Historique

- Évolution des standards dans le temps :
  - 1988 : X.500 v1 (ITU)
  - 1993 : X.500 v2 (ITU)
  - 1993 : LDAP v1 (IETF, RFC 1487)
  - 1995 : LDAP v2 (IETF, RFC 1777)
  - 1997 : X.500 v3 (ITU)
  - 1997 : LDAP v3 (IETF, RFC 2251)
  - 2001 : X.500 v4 (ITU)
- ITU : International Telecommunication Union
- IETF : Internet Engineering Task Force
- RFC : Request For Comments

# X.500 en bref

- Issu de la nécessité d'un service d'annuaire (protocole d'accès et modèle de données) pour les opérateurs de télécommunications
- Standard qui définit :
  - Un modèle de nommage des objets
  - Plusieurs protocoles de communication
  - Un mécanisme d'authentification
- Avantages :
  - Système distribué
  - Extensible
- Défauts :
  - Très lourd à implémenter

# Caractéristiques principales de X.500

- Standard OSI (protocole à 7 couches)
- Protocoles :
  - DAP : communication client-serveur (Directory Access Protocol)
  - DSP : communication serveur-serveur (Directory System Protocol)
- Données au format objets-attributs
- Objects Identifiers
- Structure hiérarchique : arbre, branches, feuilles
- Encodage ASN.1, format BER (Basic Encoding Rule)
- Sécurité : X.509 (certificats, PKI)
- Réplication

# LDAP v1

- Objectif :
  - Simplifier le protocole pour offrir la possibilité à des clients légers d'accéder à un annuaire X.500
- Implémentation technique :
  - Version légère du protocole X500 DAP
  - Frontal à X.500
- Premiers essais de normalisation :
  - Directory Assistance Service (DAS) : RFC 1202
  - Directory Interface to X.500 Implemented Efficiently (DIXIE) : RFC 1249
- Première norme LDAP :
  - RFC 1487
- LDAP : Lightweight Directory Access Protocol

# LDAP v2

- En 1995, le protocole devient un service d'annuaire :
  - RFC 1777 : Lightweight Directory Access Protocol
  - RFC 1778 : The String Representation of Standard Attribute Syntaxes
  - RFC 1779 : A String Representation of Distinguished Names
  - RFC 1959 : An LDAP URL Format
  - RFC 1960 : A String Representation of LDAP Search Filters
- Il définit déjà les éléments suivants :
  - Règles de nommage des classes d'objets et des attributs
  - Réplication
  - Les alias : liens entre entrées
  - Opérations de base
  - Authentification

# LDAP v3

- Apparu en 1998, c'est le standard actuel
- Les nouveautés par rapport à la v2 sont :
  - Les referrals : liens entre annuaires
  - La sécurité : ajout du support SASL, SSL/TLS
  - i18n : support des caractères internationaux
  - Extensibilité :
    - opérations et contrôles étendus
    - Possibilité de découverte du schéma
  - Renommage des entrées (au lieu de couper/coller)

# LDAP v3 : les RFC

- RFC 2251 : Lightweight Directory Access Protocol (v3)
- RFC 2252 : LDAP (v3) : Attribute Syntax Definitions
- RFC 2253 : LDAP (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 User Schema for use with LDAP (v3)
- RFC 2829 : Authentication Methods for LDAP
- RFC 2830 : LDAP (v3) : Extension for Transport Layer Security
- RFC 3377 : LDAP (v3) : Technical Specification
- RFC 2830 : LDAP (v3) : Extension for Transport Layer Security
- RFC 3377 : LDAP (v3) : Technical Specification
- RFC 1274 : The COSINE and Internet X.500 Schema

# LDAP v3 : les RFC

- RFC 2798 : Definition of the inetOrgPerson LDAP Object Class
- RFC 2820 : Access Control Requirements for LDAP
- RFC 2849 : The LDAP Data Interchange Format (LDIF) - Technical Specification
- RFC 3494 : LDAP v2 to Historic Status
- RFC 3671 : Collective Attributes in LDAP
- RFC 3672 : Subentries in LDAP
- RFC 3673 : LDAP v3 : All Operational Attributes
- RFC 3674 : Feature Discovery in LDAP
- Également en préparation :
  - draft-behera-ldap-password-policy-xx.txt
  - draft-zeilenga-ldap-managedit-xx.txt
  - draft-zeilenga-ldap-noop-xx.txt

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Ce que définit le protocole

- Communication client-serveur :
  - Au dessus de TCP/IP
  - LBER (Lightweight Basic Encoding Rules)
  - ASN.1 (Abstract Syntax Notation One)
  - Pas d'ASCII !
- Communication serveur-serveur :
  - Referrals : liens entre annuaires
  - Réplication : échange de données entre annuaires
- Les mécanismes de sécurité :
  - Authentification
  - Chiffrement des flux
  - Règles d'accès aux données

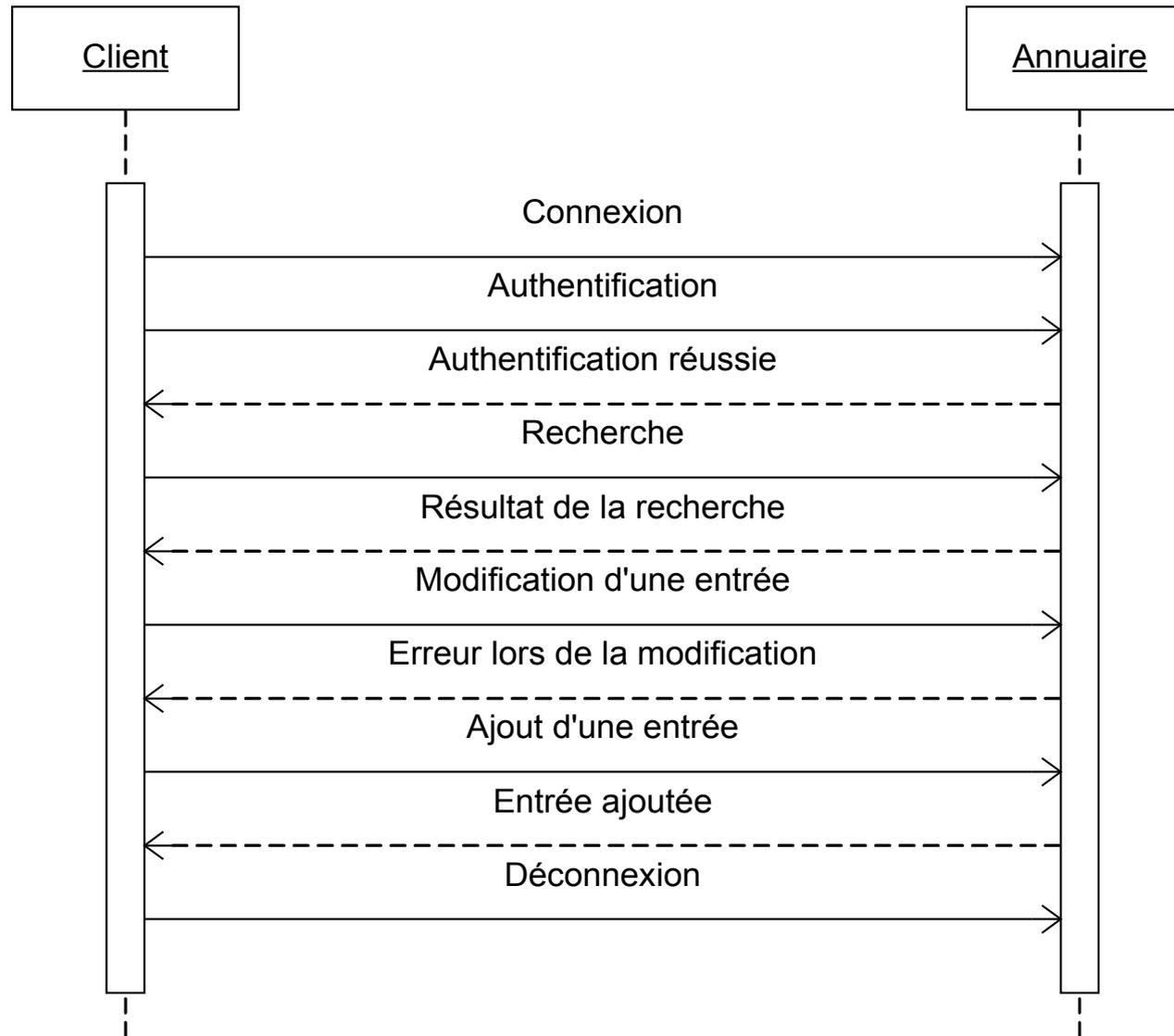
# Ce que définit le protocole

- Les opérations de base :
  - Accès au service :
    - *bind*
    - *unbind*
    - *abandon*
  - Interrogation :
    - *search*
    - *compare*
  - Mise à jour :
    - *add*
    - *delete*
    - *modify*
    - *modifyDN*

# Ce que définit le protocole

- Des règles d'extensibilité, qui permettent d'augmenter les possibilités du protocole sans avoir à modifier la norme :
  - Opérations étendues :
    - Offrir de nouvelles opérations par rapport aux opérations de base
    - Par exemple : changement de mot de passe, test d'identité...
  - Contrôles :
    - Ajouter de nouveaux paramètres aux opérations existantes
    - Par exemple : sélection de la plage d'enregistrements dans les recherches
  - SASL (Simple Authentication and Security Layer) :
    - Pouvoir utiliser de nouvelles couches de sécurité
    - Par exemple : Kerberos

# Exemple de flux LDAP

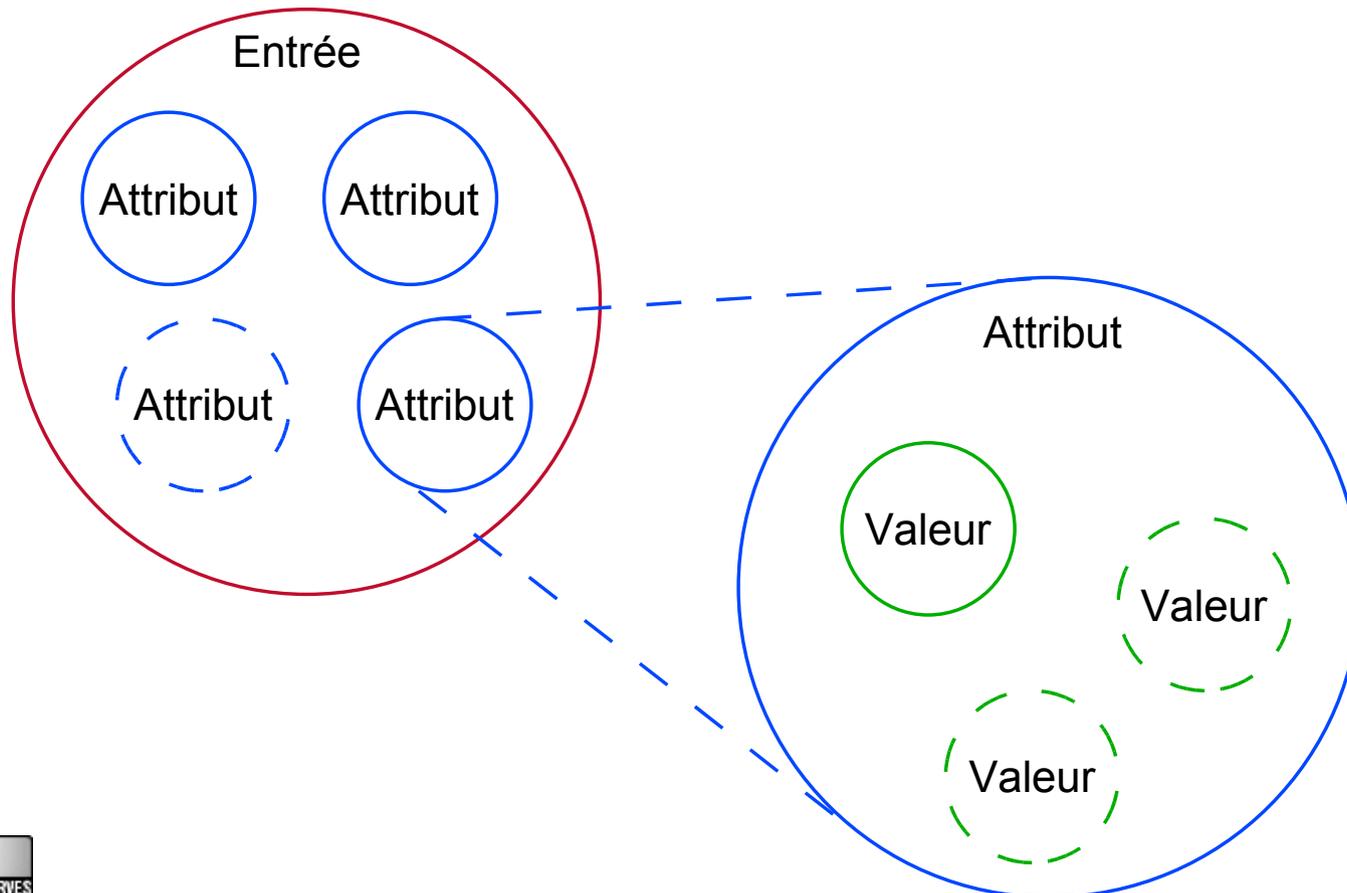


# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Le modèle d'information

- Le modèle d'information définit le type de données pouvant être stockées dans l'annuaire
- L'élément de base est l'entrée, qui contient des attributs
- Les informations sont des valeurs stockées dans les attributs



# Le modèle d'information

- Chaque entrée représente une instance d'une classe d'objet particulière
- Chaque classe d'objet est composée d'attributs (obligatoires ou optionnels)
- Chaque attribut possède une syntaxe particulière
- C'est le schéma qui spécifie le modèle d'information

# Le modèle d'information

- Le schéma spécifie :
  - Les classes d'objets (objectClasses)
  - Les attributs associés à ces classes (attributeTypes)
  - La syntaxe des attributs (ldapSyntaxes)
  - Les règles de comparaison (matchingRules)
- Il ne spécifie pas :
  - Des règles supplémentaires sur le contenu de l'annuaire
  - Des contraintes sur la structure de l'annuaire

# Schéma : classe d'objet

- Une classe d'objet, c'est :
  - Un OID
  - Un nom
  - Une hiérarchie de classes d'objets supérieures
  - Un type (ABSTRACT, STRUCTURAL ou AUXILIARY)
  - Une liste d'attributs obligatoires
  - Une liste d'attributs facultatifs

# Hiérarchie des classes d'objet

- Une classe d'objet enfant hérite des attributs de toutes ses classes supérieures
- Par exemple, la classe *organizationalPerson* hérite des attributs de la classe *top* et de la classe *person* :

2.5.6.0

NAME 'top'

ABSTRACT

MUST objectClass

2.5.6.6

NAME 'person'

SUP top

STRUCURAL

MUST (sn \$ cn)

MAY (...)

2.5.6.7

NAME 'organizationalPerson'

SUP person

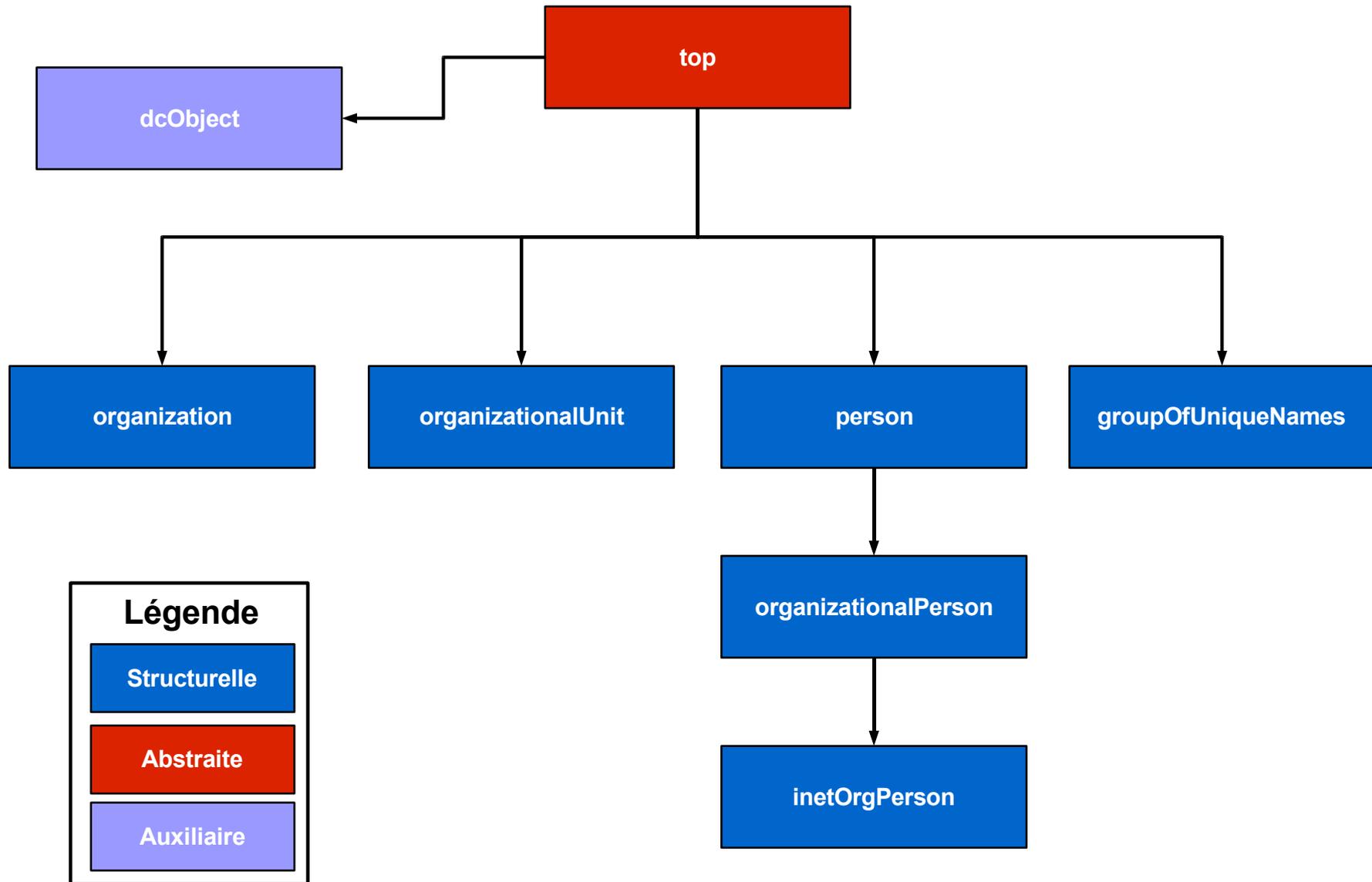
STRUCURAL

MAY (...)

# Types des classes d'objet

- Le type d'une classe d'objet :
  - ABSTRACT : désigne une classe abstraite, dont d'autres classes vont pouvoir hériter
  - STRUCTURAL : désigne une classe concrète, l'annuaire pourra contenir des entrées du type de cette classe
  - AUXILIARY : désigne une classe auxiliaire, ce qui permet de rajouter le même type d'informations à des entrées de classe structurelle différente
- Contrainte des classes structurelles :
  - Un objet doit posséder au moins une classe structurelle
  - Un objet ne peut posséder plusieurs classes structurelles que si elles appartiennent à la même hiérarchie
  - Il n'est pas possible de rajouter une classe structurelle une fois l'entrée insérée dans l'annuaire

# Exemple de hiérarchie



# Attributs obligatoires

- Une classe d'objet peut posséder un ou plusieurs attributs obligatoires
- La liste de ces attributs est constituée des attributs obligatoires de la classe elle-même augmentés des attributs obligatoires de toutes les classes supérieures et auxiliaires
- Pour chaque attribut obligatoire, une entrée doit, à tout moment, posséder au moins une valeur

# Attributs facultatifs

- Une classe d'objet peut posséder un ou plusieurs attributs facultatifs
- La liste de ces attributs est constituée des attributs facultatifs de la classe elle-même augmentés des attributs facultatifs de toutes les classes supérieures et auxiliaires
- Pour chaque attribut facultatif, une entrée peut avoir une ou plusieurs valeurs (en fonction de la définition de l'attribut), ou bien pas de valeur du tout

# Schéma : type d'attribut

- Un type d'attribut, c'est :
  - Un OID
  - Un ou plusieurs noms
  - Des règles de comparaison (EQUALITY, ORDERING et SUBSTR)
  - Une syntaxe
  - Un indicateur de multi-valuation
  - Un indicateur pour la modification par l'utilisateur
  - Un indicateur d'usage

# Règles de comparaison d'un type d'attribut

- EQUALITY matching\_rule\_name :
  - Règle à appliquer lors d'une recherche de type égalité
- ORDERING matching\_rule\_name :
  - Règle à appliquer pour trier les entrées possédant cet attribut
- SUBSTR matching\_rule\_name :
  - Règle à appliquer lors d'une recherche de type sous-chaîne

# Exemple de type d'attribut

2.5.4.20

NAME 'telephoneNumber'

EQUALITY telephoneNumberMatch

SUBSTR telephoneNumberSubstringsMatch

SYNTAX 1.3.6.1.4.1.1466.115.121.1.50{32}

# Autres notions sur les types d'attributs

- Comme pour les classes d'objet, les attributs peuvent avoir des supérieurs et hériter de leurs propriétés :
  - Syntaxe
  - Règles de comparaison
- La mention « OBSOLETE » signifie que l'attribut est toujours dans le schéma mais qu'il ne devrait plus être utilisé

# Schéma : syntaxe

- La syntaxe d'un type d'attribut :
  - chaque type d'attribut possède une syntaxe
  - Cette syntaxe spécifie le format et l'encodage des valeurs des attributs associés à ce type
  - De plus, une contrainte de longueur peut être spécifiée
- Les syntaxes LDAP :
  - Associées à un type d'attribut, elles définissent le format et l'encodage des valeurs des attributs associés à ce type
  - Elles sont spécifiées dans la RFC 2252
  - D'après la RFC, les serveurs « devraient » reconnaître l'ensemble des syntaxes décrites dedans
  - Ajouter une syntaxe nécessite de modifier le code source du serveur d'annuaire

# Exemples de syntaxe

- Les syntaxes les plus courantes sont :
  - 1.3.6.1.4.1.1466.115.121.1.7 : Boolean
  - 1.3.6.1.4.1.1466.115.121.1.12 : DN
  - 1.3.6.1.4.1.1466.115.121.1.15 : Directory String (chaîne de caractères encodée en UTF-8)
  - 1.3.6.1.4.1.1466.115.121.1.26 : IA5String (chaîne de caractères ASCII)
  - 1.3.6.1.4.1.1466.115.121.1.27 : INTEGER

# Schéma : règle de comparaison

- Les règles de comparaison :
  - Associées à un traitement particulier sur un type d'attribut (comparaison, tri), elles définissent l'algorithme à utiliser lors de ces opérations
  - Elles dépendent de la syntaxe
  - Elles sont aussi spécifiées dans la RFC 2252
- Exemples de règles de comparaison :
  - 2.5.13.0 : objectIdentifierMatch
  - 2.5.13.1 : distinguishedNameMatch
  - 2.5.13.2 : caseIgnoreMatch
  - 2.5.13.3 : caseIgnoreOrderingMatch
  - 2.5.13.4 : caseIgnoreSubstringsMatch
  - 2.5.13.8 : numericStringMatch
  - 2.5.13.10 : numericStringSubstringsMatch

# Découverte du schéma

- Le schéma, une entrée de l'annuaire :
  - Dans le standard LDAP v3, le schéma d'un annuaire fait partie intégrante de celui-ci
  - Il est possible de récupérer le schéma comme toute autre entrée, certains annuaires permettent même de le modifier
  - L'attribut *subschemaSubentry* d'une entrée classique contient le nom de l'entrée du schéma par lequel elle est régie
  - La branche du schéma est annoncé dans la racine de l'annuaire (RootDSE)
- Commande LDAP pour lecture du schéma :

```
ldapsearch -H ldap://IP:PORT -b cn=subschema -s base +
```

# Le schéma : un standard

- Les schémas par défaut sont issus des standards :
  - RFC 2256 : core
  - RFC 1274 : cosine
  - RFC 2798 : inetOrgPerson
- Ces schémas sont généralement décrits dans des fichiers chargés au démarrage de l'annuaire
- Il est bien évidemment possible de les compléter avec son propre schéma, mais il faut pour cela obtenir son propre préfixe d'OID auprès de l'IANA :

<http://www.iana.org/cgi-bin/enterprise.pl>

# Particularités

- Classe d'objet sans supérieur : top
- Classes d'objets sans héritier possible : alias, referral
- Classe d'objet possédant tous les attributs : extensibleObject
- Quelques attributs opérationnels :
  - creatorsName :
    - DN du créateur de l'entrée
  - createTimestamp :
    - date de création de l'entrée
  - modifiersName :
    - DN du dernier utilisateur qui à mis à jour l'entrée
  - modifyTimestamp :
    - date de dernière mise à jour de l'entrée

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

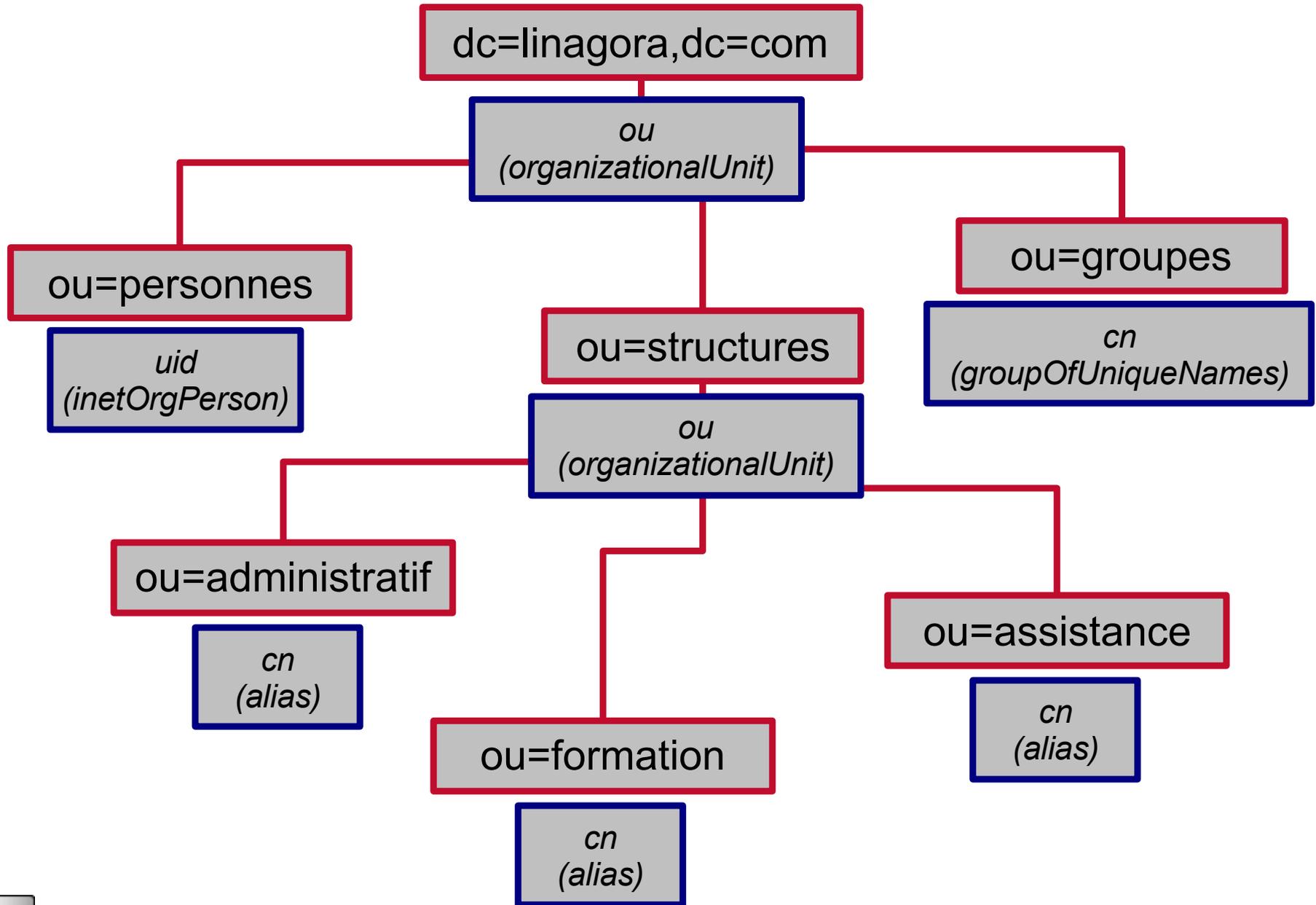
# Le modèle de nommage

- Le modèle de nommage définit comment sont organisées les entrées de l'annuaire et comment elles sont référencées
- Les entrées sont stockées selon une structure logique hiérarchique appelée DIT (Directory Information Tree)
- Au sein de cet arbre, les entrées (soit nœud, soit feuille) sont identifiées par un nom, le DN (Distinguished Name)
- Un DN est unique dans l'annuaire, un RDN (Relative Distinguished Name) est unique dans sa branche

# Le DIT

- Le DIT est un outil de conception d'annuaire, mais n'a (actuellement) aucune implémentation technique
- Le DIT définit :
  - Le contexte de nommage
  - Les branches
  - Les entrées des branches :
    - Leur RDN
    - Leurs classes d'objets

# Exemple de DIT



# Le suffixe ou contexte de nommage

- C'est l'entrée de plus haut niveau dans l'arbre des données
- Conventions d'écriture :
  - Sous X.500 :
    - Suffixe géographique avec l'ajout de l'organisation :
    - Exemple : o=linagora,c=com
  - Depuis LDAP v3 :
    - Suffixe basé sur le nom DNS
    - Exemple : dc=linagora,dc=com
- Un même serveur peut gérer plusieurs suffixes
- Un suffixe doit avoir la syntaxe d'un DN
- Il peut être aussi long que voulu
- Il peut utiliser n'importe quel attribut du schéma

# La racine

- La racine est aussi appelée RootDSE (Root DSA Specific Entry)
- Elle contient des informations importantes sur l'annuaire :
  - *namingContexts* : suffixes gérés par cet annuaire
  - *supportedExtension* : liste des opérations étendues supportées
  - *supportedControl* : liste des contrôles supportés
  - *supportedSaslMechanisms* : mécanismes SASL supportés
  - *supportedLdapVersion* : versions du protocole supportées
  - ...
- La racine est située au-dessus des contextes de nommage
- L'entrée peut être lue en spécifiant un DN vide comme base de recherche :  
`ldapsearch -H ldap://IP:PORT -b "" -s base +`

# Le DN

- Le DN ou Distinguished Name :
  - Référence, de manière unique, une entrée du DIT
  - Est composé du DN de l'entrée supérieure, préfixé par le RDN (Relative Distinguished Name) de l'entrée courante
- Sous une branche donnée, chaque RDN est unique
- Format définit dans la RFC 2253 :
  - Pas d'espace ou de # au début
  - Pas d'espace à la fin
  - Pas les caractères , + " \ < > ;
  - Il est possible d'échapper ces caractères avec \
- Exemple :
  - ou=personnes, dc=linagora, dc=com

# Les alias

- Lors d'une recherche, le client choisit comment l'annuaire doit traiter les alias :
  - never : ne jamais déréférencer les alias
  - searching : déréférence tous les alias, sauf la base de recherche
  - finding : déréférence uniquement la base de recherche
  - always : déréférence tous les alias
- Exemple d'entrée alias :

dn: cn=abc1234,ou=formation,ou=structures,dc=linagora,dc=com

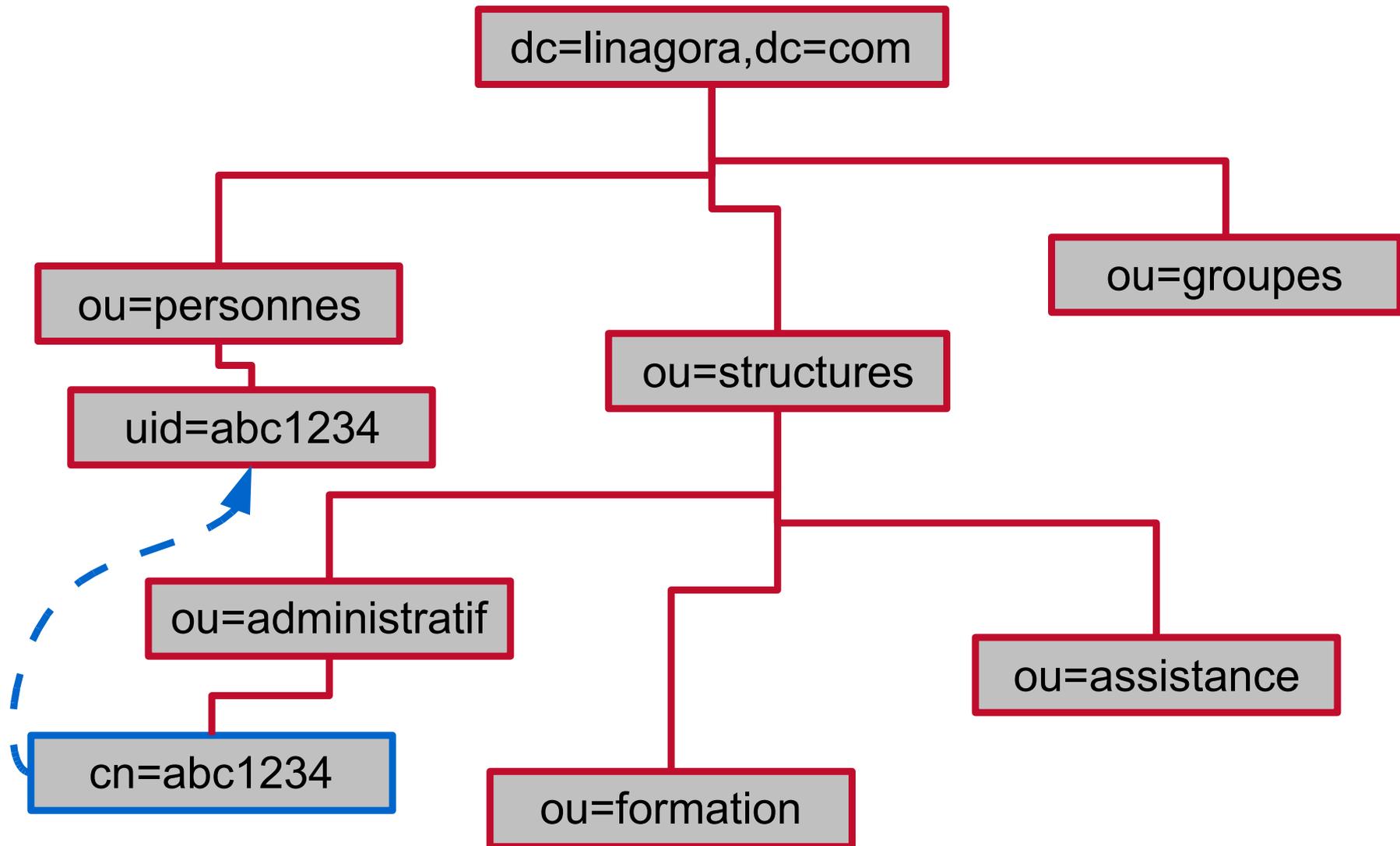
objectClass: alias

objectClass: extensibleObject

cn: abc1234

aliasedObjectName: uid=abc1234,ou=personnes,dc=linagora,dc=com

# Exemple d'alias



# Les referrals

- Lors d'une recherche, c'est le client qui décide comment traiter les referrals :
  - follow : suivre les referrals
  - throw : renvoyer une erreur (et l'URL LDAP du referral)
- Les referrals sont apparus dans LDAP v3, et permettent le chaînage des annuaires, calqué sur le modèle du DNS
- Exemple d'entrée referral :

dn: ou=structures,dc=linagora,dc=com

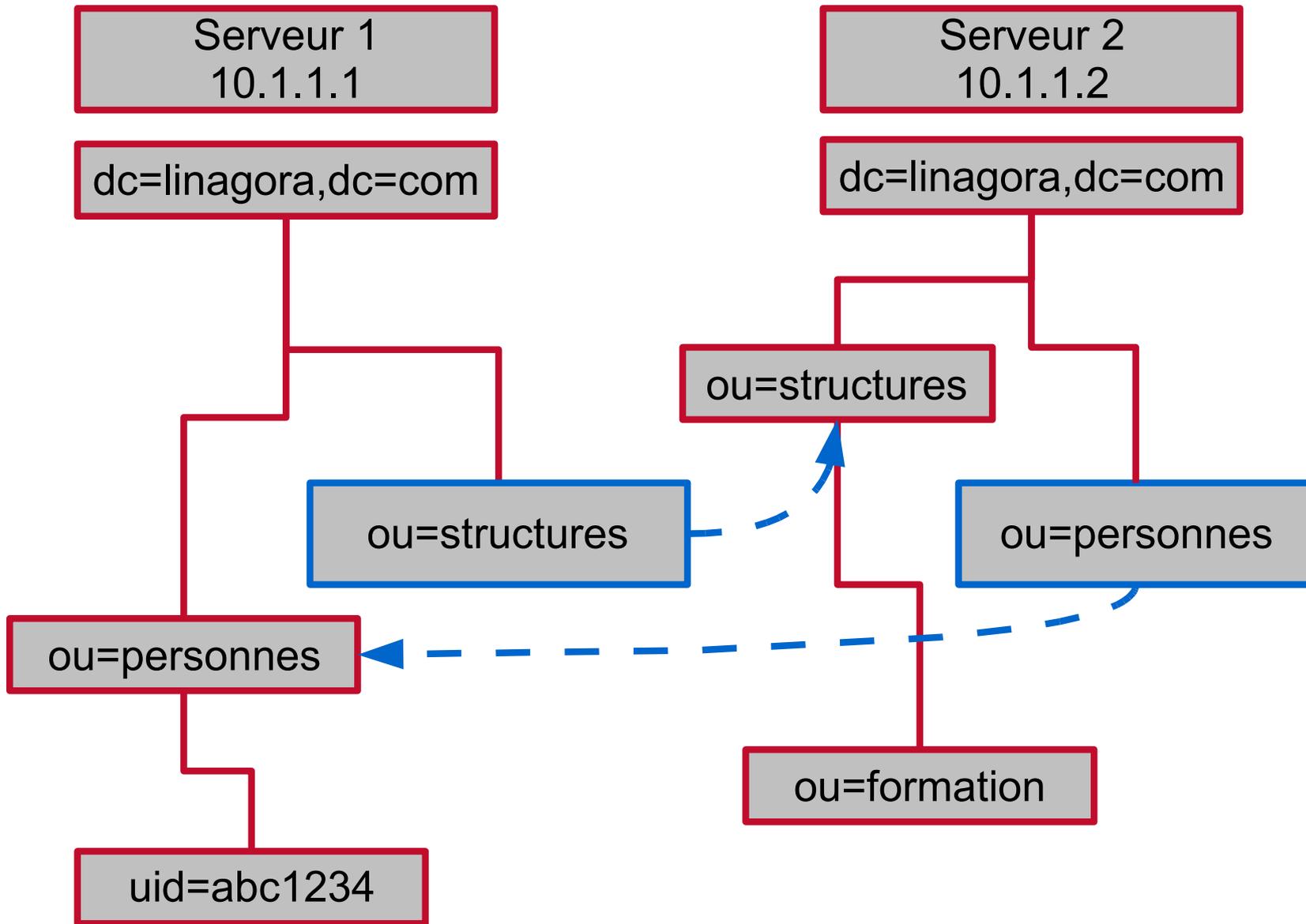
objectClass: referral

objectClass: extensibleObject

ou: structures

ref: ldap://10.1.1.2:389/ou=structures,dc=linagora,dc=com

# Exemple de referral



# URL LDAP

- Syntaxe d'une URL LDAP :  
ldap[s]://server:port/base?attributes?scope?filter
- Paramètres de l'URL :
  - *server* : nom ou IP du serveur LDAP
  - *port* : port d'écoute de l'annuaire
  - *base* : DN de base de la recherche
  - *attributes* : liste des attributs à récupérer, séparés par une virgule
  - *scope* : étendue de la recherche
  - *filter* : filtre de la recherche

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Le modèle fonctionnel

- Le modèle fonctionnel définit 9 opérations de base :
  - *bind*
  - *unbind*
  - *abandon*
  - *search*
  - *compare*
  - *add*
  - *delete*
  - *modify*
  - *modifyDN*
- Il définit aussi la possibilité d'avoir des opérations étendues (*extended*), et des contrôles étendus appliqués aux opérations de base

# Opérations de base

- L'accès au service :
  - *bind* :
    - S'authentifie auprès de l'annuaire
    - Paramètres : DN et mot de passe (en clair)
    - Si pas de paramètres, la connexion est anonyme
  - *unbind* :
    - Se déconnecte auprès de l'annuaire
    - Termine la connexion TCP/IP
    - Pas de paramètres
    - Obligatoire pour mettre fin à une connexion
  - *abandon* :
    - Abandonne une opération asynchrone en cours
    - Paramètres : identifiant de l'opération

# Opérations de base

- L'interrogation :
  - *search* :
    - Effectue une recherche
    - Seule opération pouvant obtenir plusieurs réponses
    - Opération la plus utilisée sur les annuaires
    - Nombreux paramètres
  - *compare* :
    - Exécute un test d'égalité sur la valeur d'un attribut
    - Paramètres : DN, nom de l'attribut à tester, valeur à tester
    - Résultat :
      - TRUE si l'entrée possède bien le couple attribut-valeur
      - FALSE autrement

# Paramètres de la recherche

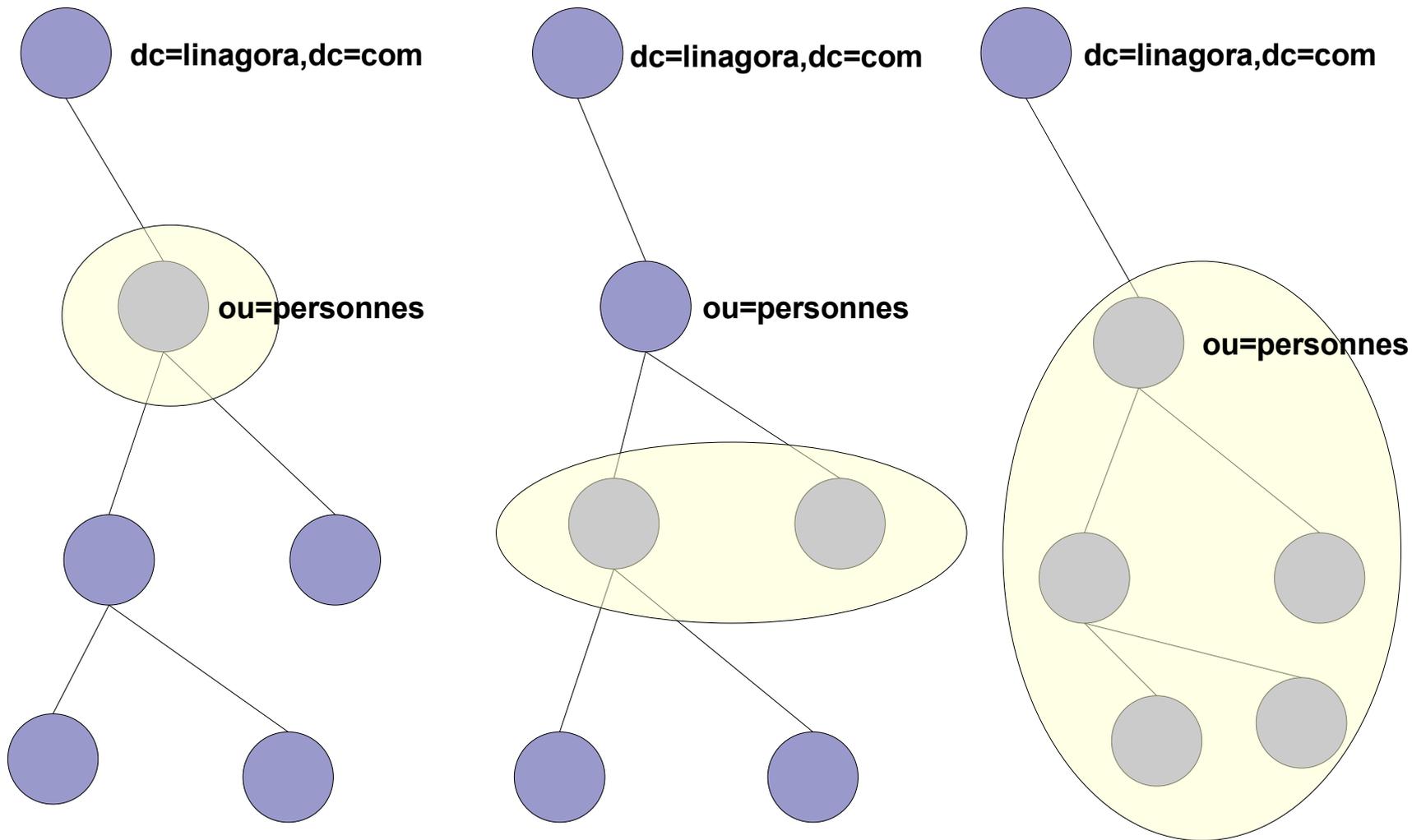
- Les paramètres d'une recherche :
  - Base : DN de l'entrée à partir de laquelle se fera la recherche
  - Étendue (scope) : périmètre de la recherche
  - Filtre (filter) : basé sur les valeurs des attributs
  - Attributs à récupérer
  - Nombre maximum d'entrées à retourner
  - Durée maximale de la recherche
  - Comment traiter les alias
  - Comment traiter les referrals
  - Contrôles étendus

# Étendue d'une recherche

BASE

ONE

SUB



Base de la recherche : `ou=personnes,dc=linagora,dc=com`

# Filter d'une recherche

- Les filtres sont définis dans la RFC 2254
- Un filtre simple se compose de :
  - une parenthèse ouvrante
  - un attribut
  - un type
  - une valeur
  - une parenthèse fermante
- Un filtre complexe se compose de :
  - une parenthèse ouvrante
  - un opérateur
  - un ou plusieurs filtres simples
  - une parenthèse fermante

# Types et opérateurs

- Les types de filtres sont :
  - Présence : attribut=\*
  - Égalité : attribut=valeur
  - Sous-chaîne : attribut=v\*le\*
  - Approximation : attribut=~valeur
  - Comparaison :
    - Plus petit ou égal : attribut<=valeur
    - Plus grand ou égal: attribut>=valeur
- Les opérateurs sont :
  - Et : &
  - Ou : |
  - Non : !

# Exemples de filtres

- Exemples de filtres de recherche simples :
  - (mail=\*)
  - (sn=durand)
  - (objectClass=inetOrgPerson)
  - (givenName=pat\*)
  - (givenName=\*ice)
  - (givenName=\*marie\*)
- Exemples de filtres de recherche complexes :
  - (&(mail=\*)(telephoneNumber=\*))
  - (|(sn=durand)(sn=durant))
  - (!(givenName=\*marie\*))

# Opérations de base

- La mise à jour :
  - *add* :
    - Ajoute une nouvelle entrée
    - Vérification par rapport au schéma
    - Paramètres : totalité de l'entrée (DN + contenu)
  - *delete* :
    - Supprime une entrée
    - Il n'est pas possible de supprimer une entrée de type noeud
    - Paramètres : DN de l'entrée

# Opérations de base

- La mise à jour :
  - *modify* :
    - Modifie une entrée existante
    - Vérification par rapport au schéma
    - Plusieurs variantes possibles :
      - *add* : ajoute une ou plusieurs valeurs à l'attribut
      - *delete* : supprime une ou plusieurs valeurs de l'attribut ou l'attribut
      - *replace* : remplace toutes les valeurs de l'attribut
    - Paramètres : DN de l'entrée, variante, attribut et valeurs
  - *modifyDN* :
    - Renomme une entrée
    - Il n'est pas possible de renommer une entrée de type noeud
    - Par défaut l'ancienne valeur du RDN est conservée dans l'entrée
    - Paramètres : DN, nouveau RDN, nouveau supérieur, conservation de l'ancienne valeur du RDN

# Opérations étendues

- Les opérations étendues gérées dans un annuaire sont listées dans son RootDSE
- Un client peut appeler une opération étendue par son OID et ses paramètres en conservant son API standard
- Certaines opérations étendues sont normalisées :
  - RFC 4532 : Who am I
  - RFC 3909 : Cancel
  - RFC 2830 : StartTLS
  - RFC 3062 : Password modify

# Principaux codes de retour

- LDAP\_SUCCESS (0) : tout va bien.
- LDAP\_TIMELIMIT\_EXCEEDED (3)  
LDAP\_SIZELIMIT\_EXCEEDED (4) : une limite spécifiée par le client ou par le serveur a été dépassée.
- LDAP\_REFERRAL (9) : un *referral* a été renvoyé.
- LDAP\_INVALID\_SYNTAX (21) : une valeur spécifiée n'a pas la syntaxe autorisée.
- LDAP\_NO\_SUCH\_OBJECT (32) : recherche sans résultat ou authentification erronée.
- LDAP\_INVALID\_DN\_SYNTAX (34) : la syntaxe du DN est incorrecte.
- LDAP\_INVALID\_CREDENTIALS (49) : identifiant ou mot de passe incorrect.
- LDAP\_INSUFFICIENT\_ACCESS (50) : droits insuffisants pour l'opération demandée, généralement lorsque l'on est authentifié en anonyme.
- LDAP\_UNAVAILABLE (52) : le serveur LDAP ne répond pas.

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Le modèle de sécurité

- Le modèle de sécurité décrit les moyens de protéger les données de l'annuaire
- La sécurité peut se faire à différents niveaux :
  - Lors de l'accès à l'annuaire, en nécessitant une authentification
  - Lors de l'accès aux données, par des contrôles d'accès
  - Lors du transfert des données, par le chiffrement des flux

# Accès à l'annuaire

- Par défaut, accès en anonyme :
  - Opération *bind* sans paramètres
  - Opération *bind* avec le DN
- Méthodes d'authentification disponibles :
  - Mot de passe en clair
  - Mot de passe dans une connexion SSL/TLS
  - Certificats sur SSL/TLS
  - Kerberos V4
  - SASL
- Autorisation mandataire :
  - Connexion en administrateur (*rootdn*) puis acquisition de l'identité d'un utilisateur

# Accès aux données

- ACLs :
  - Access Control Lists
  - Permettent d'écrire des règles du type que-qui-quoi :
    - que : entrées et attributs concernés
    - qui : utilisateurs pour lesquels s'applique cette règle
    - quoi : opérations autorisées
  - Le format des ACLs n'est pas encore standardisé
- ACIs :
  - Access Control Items
  - Intégrées aux entrées de l'annuaire
  - Le format des ACIs n'est pas encore standardisé

# Transfert des données

- Objectif :
  - Assurer la confidentialité et l'intégrité des données
- Solution :
  - Chiffrement des flux
- Méthodes :
  - Tunnel SSL :
    - Ouverture d'une connexion sécurisée sur le port 636
    - Utilisation des URL ldaps://
  - Opération étendue startTLS :
    - Connexion standard sur le port 389
    - Chiffrement de la connexion après son établissement sur TCP/IP

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Le modèle de duplication

- Le modèle de duplication décrit comment organiser les données entre différents serveurs
- Utilisation :
  - backup : sauvegarder les données
  - distribution : rapprocher les données des utilisateurs
  - fail-over : continuité du service en cas de panne
  - load-balancing : répartition de la charge
- **ATTENTION** : le modèle de duplication n'est pas encore standardisé

# Répartition et réplication

- Répartition :
  - Les données sont morcelées sur plusieurs annuaires
  - Les liens entre les annuaires sont faits par referrals
  - Possibilité d'utiliser un méta-annuaire pour fédérer les arbres
- Réplication :
  - Les données sont dupliquées sur les annuaires
  - Partielle ou totale
  - Deux modes :
    - SIR : Server Initiated Replication, le maître alimente les esclaves
    - CIR : Consumer Initiated Replication, les esclaves contactent le maître
  - Possibilité de faire de la réplication mutli-maîtres, mais sans garantir l'intégrité des données

# Précautions

- Les annuaires doivent posséder le même schéma
- Les règles d'accès aux données doivent être dupliquées
- La reprise sur échec ou la répartition de charge doivent être assurés par d'autres équipements
- La réplication multi-maîtres :
  - Peut permettre une haute-disponibilité des écritures derrière un équipement de reprise sur échec
  - Est dangereuse avec de la répartition de charge

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# LDIF

- LDAP Data Interchange Format
- LDIF est le standard de représentation des entrées sous forme texte (RFC 2849)
- Il permet aussi d'exécuter des opérations LDAP
- Le fichier est au format ASCII
- L'encodage des caractères internationaux est l'UTF-8
- Chaque entrée est séparée par une ligne blanche
- Les commentaires doivent être précédés du caractère #

# Le format LDIF

- Les valeurs ASCII sont indiquées ainsi :
  - attribut: valeur
- Les valeurs non ASCII doivent être codées en base64 :
  - attribut:: Ndbhs34cde673hd638NC8Ehhede73be\_
- En théorie un fichier LDIF récent commence par :
  - version: 1
- Une ligne ne doit pas excéder 80 caractères pour assurer la compatibilité avec les anciens systèmes.
- Un espace en début de ligne signifie qu'elle est la suite de la ligne précédente
- Un fichier peut être inclus avec < en début de valeur :
  - jpegphoto:< file:///usr/local/directory/photos/linagora.jpg

# LDIF simple

dn: uid=1234,ou=Personnes,dc=linagora,dc=com

objectClass: top

objectClass: person

objectClass: organizationalPerson

objectClass: inetOrgPerson

uid: 1234

uid: ddurand

cn: Denis DURAND

sn: DURAND

description: Formateur

title: Monsieur

# Le LDIF pour modifier

- Le LDIF permet de représenter les opérations modify, add, delete et modrdn (« changetype »).
- Les opérations à l'intérieur d'une même entrée sont séparées par un -
- Pour la modification (modify), les options suivantes sont disponibles :
  - add
  - delete
  - replace
- Pour modrdn :
  - newrdn
  - deleteoldrdn
  - newsuperior

# LDIF de modification

dn: uid=1234,ou=personnes,dc=linagora,dc=com

changetype: modify

replace: uid

uid: test

-

add: description

description: beau, grand et intelligent

-

# DSML

- Directory Services Markup Language
- DSML v1 est le standard de représentation des données d'un annuaire au format XML
- DSML v2 permet de décrire des opérations LDAP (interrogations et mises à jour) au format XML
- La communication avec le serveur se fait soit par SOAP (à travers un réseau), soit par des scripts (en local)
- Les 2 versions sont complémentaires

# Exemple DSML v1

```
<dsml:directory-entries>
  <dsml:directory-entry dn="uid=2345,ou=personnes,dc=linagora,dc=com">
    <dsml:objectclass>
      <dsml:oc-value>top</dsml:oc-value>
      <dsml:oc-value>person</dsml:oc-value>
      <dsml:oc-value>organizationalPerson</dsml:oc-value>
      <dsml:oc-value>inetOrgPerson</dsml:oc-value>
    </dsml:objectclass>
    <dsml:attr name="sn"><dsml:value>Ball</dsml:value></dsml:attr>
    <dsml:attr name="uid"><dsml:value>2345</dsml:value></dsml:attr>
    <dsml:attr name="mail">
      <dsml:value>mball@linagora.com</dsml:value></dsml:attr>
    <dsml:attr name="givenname">
      <dsml:value>Michael</dsml:value></dsml:attr>
    <dsml:attr name="cn">
      <dsml:value>Michael Ball</dsml:value></dsml:attr>
    </dsml:directory-entry>
  </dsml:directory-entries>
```

# Exemple DSML v2

```
<dsmlEnvelopeRequest xmlns="http://www.dsml.org/DSML/v2">
  ...
  <searchRequest dn="ou=personnes,dc=linagora,dc=com">
    <scope>singleLevel</scope>
    <derefAliases>neverDerefAliases</derefAliases>
    <sizeLimit>1000</sizeLimit>
    <filter>(sn=john*)</filter>
    <control>
      <controlType>1.2.840.113556.1.4.612</controlType>
      <criticality>true</criticality>
      <controlValue>U2VhcmNoIFJlcXVlc3QgRXhhbXBsZQ==
    </controlValue>
    </control>
  </searchRequest>
  ...
</dsmlEnvelopeRequest>
```

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Marché des annuaires

- OpenLDAP :
  - Annuaire LDAP natif
  - Gratuit, libre et bon respect des standards
  - Peu de documentation et petite communauté
- Sun Java System Directory Server :
  - Annuaire LDAP natif
  - Successeur de Netscape et iPlanet Directory Server
  - Très peu de contrôle sur le format des données
- OpenDS :
  - Nouveau projet Open Source de SUN
  - Entièrement en java
  - Encore en développement

# Marché des annuaires

- Microsoft Active Directory :
  - Version Windows 2000 et XP très peu compatible avec le standard
- Novell eDirectory :
  - Annuaire basé sur X.500 utilisé, à l'origine, pour Netware
  - Impose énormément de contraintes
- Red Hat/Fedora Directory Server :
  - Issu de Netscape Directory Server 4.16
  - Libre mais interface d'administration dépendante de Java
- Apache Directory Server :
  - Libre, vient d'arriver sur le marché
  - Codé en java
  - Supporte aussi les protocoles DNS, NTP, etc.

# Quelques clients LDAP

- LDAP Browser/Editor : navigateur graphique LDAP en Java
- JXplorer : autre navigateur graphique LDAP en Java
- GQ : navigateur graphique s'appuyant sur GTK
- Luma : gestionnaire graphique, basé sur python-ldap, extensible via des greffons
- phpLDAPadmin : client web d'administration/exploration LDAP
- ldapvi : pour modifier des entrées LDAP avec un éditeur de texte
- Calendra Directory Manager® : gestionnaire de contenu d'annuaires orienté métier, propriétaire

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Exemples de mise en œuvre

- Annuaire de données Samba 3 :
  - Classes d'objet posixAccount et sambaSamAccount
  - Stocke les utilisateurs, les groupes, les machines...
  - Authentification sur des attributs spécifiques :
    - sambaLMPassword
    - sambaNTPassword
  - Lecture à 99%
  - Écriture :
    - Création des comptes utilisateurs, groupes et machines
    - Rattachement d'une machine à un domaine
  - Penser au cache sur les serveurs Samba pour alléger la charge sur les annuaires LDAP

# Exemples de mise en œuvre

- Annuaire d'authentification :
  - Utilisation de l'attribut userPassword
  - Droits de lecture/écriture à limiter
  - Peu de données
  - Beaucoup de requêtes
  - Exemples :
    - Serveurs de messagerie
    - Module d'authentification applicatif
    - Interfaces Web (intranet, extranet, etc.)

# Exemples de mise en œuvre

- Annuaire « ressources humaines » :
  - Lecture importante
  - Classes d'objets « métier » à définir
  - Chaînes d'alimentation à développer
  - Interfaces de consultation à étudier :
    - Carnets d'adresses Outlook/Thunderbird
    - Intranet
    - Navigateurs LDAP

# Exemples de mise en œuvre

- Annuaire « gestion des applicatifs » :
  - Lecture importante
  - Stockage de données applicatives
  - Stockage des rôles
  - Modèle Identification -> Authentification -> Autorisation
  - Exemples :
    - Portail SSO
    - Application métier
    - Logiciel RH

# Les APIs

- La mise en œuvre d'annuaires passe souvent par la création de programmes ou d'interfaces. Cela est rendu possible par les différentes APIs (Application Programming Interface) :
  - C :
    - API d'OpenLDAP
    - Sun ONE Directory SDK for C
  - Java :
    - Sun ONE Directory SDK for Java
    - Java Naming and Directory Interface (JNDI), de SUN
    - JLDAP : classes LDAP Java, contribution de Novell pour OpenLDAP
  - Autres langages :
    - Perl : API Net::LDAP
    - Python : API Python LDAP
    - PHP : API PHP LDAP

# Exemple de code JLDAP

```
import java.io.UnsupportedEncodingException;
import com.novell.Idap.LDAPConnection;
import com.novell.Idap.LDAPException;

public static void main( String[] args ) {

    int version  = LDAPConnection.LDAP_V3;
    int port     = LDAPConnection.DEFAULT_PORT;
    String host  = args[0];
    String dn    = args[1];
    String passwd = args[2];
    LDAPConnection conn = new LDAPConnection();

    conn.connect( host, port );
    conn.bind( version, dn, passwd.getBytes("UTF8") );

    System.out.println((conn.isBound()) ?
        "\n\tAuthenticated to the server ( simple )\n":
        "\n\tNot authenticated to the server\n");
}
```

# Exemple de code JNDI

```
import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NamingException;

Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY,
        "com.sun.jndi.fscontext.RefFSContextFactory");

try {
    Context ctx = new InitialContext(env);
    Object obj = ctx.lookup(name);

    System.out.println(name + " is bound to: " + obj);

} catch (NamingException e) {
    System.err.println("Problem looking up " + name + ": " + e);
}
```

# Exemple de code PHP

```
$server = "localhost";
$port = "389";
$racine = "dc=linagora,dc=com";
$rootdn = "cn=admin,dc=linagora,dc=com";
$rootpw = "secret";

echo "Connexion...<br>";
$ds=ldap_connect($server);
if ($ds==1)
    {
        $r=ldap_bind($ds,$rootdn,$rootpw);
        echo "Déconnexion...<br>";
        ldap_close($ds);
    }
else {
    echo "Impossible de se connecter au serveur LDAP";
}
```

# Exemple de code Perl

```
#!/usr/bin/perl -w
use strict;
use Net::LDAP;

my $mesg;
my $ldap = Net::LDAP->new("ldap1.example.com", port => 389, version => 3)
    or die "Can't contact master ldap server ($@)";

$mesg = $ldap->bind( "cn=Manager,dc=example,dc=com", password =>
    'secret' );
$mesg->code && die $mesg->error;

$ldap->unbind();
```

# Sommaire

- Introduction et historique des annuaires
- LDAP :
  - Le protocole
  - Le modèle d'information
  - Le modèle de nommage
  - Le modèle fonctionnel
  - Le modèle de sécurité
  - Le modèle de duplication
- LDIF et DSML
- OpenLDAP et les annuaires existants
- Exemples de mise en œuvre
- Résumé des acronymes

# Acronymes

- ACI : Access Control Item
- ACL : Access Control List
- API : Application Programming Interface
- ASCII : American Standard Code for Information Interchange
- ASN.1 : Abstract Syntax Notation One
- BER : Basic Encoding Rule
- DAP : Directory Access Protocol
- DIT : Directory Information Tree
- DN : Distinguished Name
- DNS : Domain Name System
- DSA : Directory System Agent
- DSE : DSA Specific Entry
- DSML : Directory Services Markup Language

# Acronymes

- IANA : Internet Assigned Numbers Authority
- IETF : Internet Engineering Task Force
- IP : Internet Protocol
- ISO : International Organization for Standardization
- ITU : International Telecommunication Union
- LBER : Lightweight Basic Encoding Rule
- LDAP : Lightweight Directory Access Protocol
- LDIF : LDAP Data Interchange Format
- OID : Object Identifier
- OSI : Open System Interconnection
- PKI : Public Key Infrastructure
- RDN : Relative Distinguished Name
- RFC : Request For Comments

# Acronymes

- SASL : Simple Authentication and Security Layer
- SOAP : Simple Object Access Protocol
- SSL : Secure Sockets Layer
- TCP : Transmission Control Protocol
- TLS : Transport Layer Security
- URL : Uniform Resource Locator
- UTF : Unicode Transformation Format
- XML : eXtensible Markup Language

# Et si vous souhaitez continuer votre apprentissage...

Détachez ce coupon et adressez-le au pôle **Formation** :

Yves MIEZAN EZO  
 Email : [formation@linagora.com](mailto:formation@linagora.com)  
 Tél : 01 58 18 68 28  
 Fax : 01 58 18 68 29



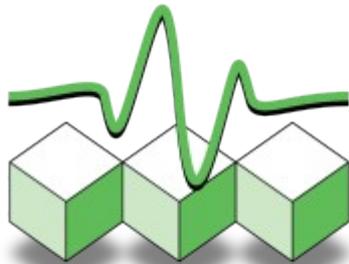
**Bénéficiez de  
 100 €  
 de réduction  
 sur votre prochaine  
 formation !**

Nom :	.....
Prénom :	.....
Société :	.....
Mail :	.....
Tél :	.....
Stage :	.....Date : .....
Tarif catalogue :	.....€.....Réduction : - 100 €.....Tarif final :.....€

Code Opération « **LNGFetdevientfortenlibre** »

**LINAGORA**

*Formation*



Administration et sécurité

**M**erci de votre attention

LINAGORA *Formation*  
[formation@linagora.com](mailto:formation@linagora.com)