Le protocole LDAP

Lightweight Directory Access Protocol

HEPIA
Année académique 2014/2015

Contenu

- Introduction
- Modèle des données LDAP
- Espace de nommage LDAP
- Aperçu du protocole

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

- Un conteneur d'informations organisées
- Un service d'annuaire électronique c'est en plus :
  - Un format de message pour accéder au contenu des entrées de l'annuaire à distance
  - Une syntaxe de représentation des données pour interroger/stocker la base de donnée qui contient l'annuaire
  - Un protocole de mise à jour du contenu.
- Et aussi
  - Un modèle de duplication des données
  - Un modèle de distribution des données

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

- Spécificités des annuaires électroniques
  - Dynamiques (si les informations changent -> il faut mettre l'annuaire à jour)
  - Souples (changement aisé, typage et organisation des données)
  - Peuvent être sécurisés (qui peut voir quoi)
  - Peuvent être personnalisés (façon de présenter les données, actions sur ses propres données...

Introduction : les annuaires d'entreprise

- Les plus classiques :
  - L'annuaire téléphonique des employés
  - Le répertoire des fournisseurs
  - La base clients
  - Le catalogue des produits
  - L'inventaire
  - ...
- Les annuaires d'entreprise peuvent être :
  - + ou - nombreux (> 100 dans grandes entreprises)
  - + ou - informatisés
  - + ou - facilement consultables
  - gérés dans des services différents
  - dans des formats différents
  - + ou - à jour
  - + ou - redondants
  - + ou - incohérents

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

- Caractéristiques comparées des annuaires et des bases de données
  - Rapport lecture/écriture (beaucoup) élevé pour les annuaires.
  - Annuaires plus facilement extensibles : présence de types définis par l'utilisateur (certificats sécurité par ex).
  - Les annuaires diffusent leur données à plus large échelle (cf DNS)
  - Distribution des données entre les serveurs plus facile avec les annuaires
  - Plus grande duplication des informations des annuaires.
  - Importance des standards (LDAP)
  - Performances en lecture des annuaires plus élevées

Introduction : qu'est-ce que n'est pas un annuaire

- Approprié à de fréquentes écritures
- Destiné à manipuler des données volumineuses
- Un substitut à un serveur FTP, un système de fichiers...

Contenu

- Introduction
- Modèle des données LDAP
- Espace de nommage LDAP
- Aperçu du protocole

Modèle des données LDAP

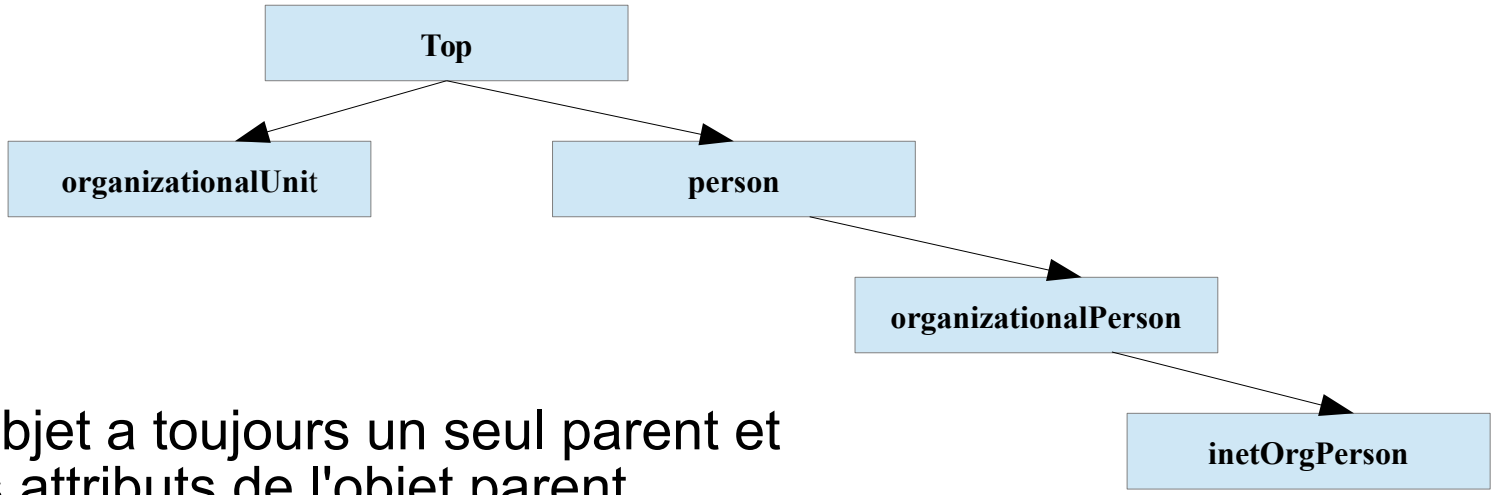
- Le Modèle de données définit le type de données pouvant être stockées dans l'annuaire
  - Basé sur des objets appelées *entrées* (entry)
  - Une *entrée* contient une séquence d'attributs
  - Chaque *entrée* est identifiée de manière unique par l'attribut distinguished name (`dn`)
  - Chaque entrée est typée et définie par l'attribut de *classe d'objet* (`objectClass`)
    - Chaque classe d'objet a elle même aussi des *attributs*
    - Chaque attribut a un type et une ou plusieurs valeurs autorisées

Modèle des données LDAP : Classes d'objets

- Modélisent des objets réels ou abstraits en les caractérisant par une liste d'attributs optionnels ou obligatoires. Une classe d'objet est définie par :
  - Un Nom, qui l'identifie
  - Un OID (Object Identifier) qui l'identifie aussi
  - Des attributs obligatoires
  - Des attributs optionnels
  - Un type (structuré, auxiliaire ou abstrait)
- Exemples :
  - Une organisation (`o`)
  - Ses départements (`ou`)
  - Son personnel (`organizationalPerson`)
  - Ses imprimantes (`device`)
  - Ses groupes de travail (`groupofnames`)

Hiérarchie des classes d'objets

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



- Chaque objet a toujours un seul parent et hérite des attributs de l'objet parent
- On précise la classe d'objet d'une entrée à l'aide de l'attribut d'entrée `objectClass`
- Il faut obligatoirement indiquer la parentée de la classe d'objet en partant de l'objet `top` et en passant par chaque ancêtre de l'objet

Exemple pour une entrée de type inetOrgPerson

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

- L'objet `person` a comme attributs `commonName(cn)`, `surname(sn)`, `description`, `seeAlso`, `telephoneNumber`, `userPassword`
- L'objet fils `organizationalPerson` ajoute des attributs comme : `organizationUnitName`, `title`, `postalAddress`, ...
- L'objet petit-fils `inetOrgPerson` lui rajoute des attributs comme : `mail`, `labeledURI`, `userID (uid)`, `photo`, ...

- Les objets ont une forme standard et sont définis en ASN.1 dans des RFC (ex : RFC 4517, RFC 2798)
- Une entrée peut appartenir à un nombre non limité de classes d'objets
- Les attributs obligatoires de l'entrée sont la réunion des attributs obligatoires de chaque classe

Attributs de classe d'objets

- Ils sont aussi typés et caractérisés par
  - Un nom qui l'identifie
  - Un OID (Object Identifier) qui l'identifie
  - Si il est mono ou multi-valué
  - Une syntaxe et des règles de comparaison
  - Un format ou une limite de taille de valeur qui lui est associée

Type d'attribut	Valeur d'attribut
cn:	Bob John
uid:	bjohn
telephonenumber:	+41 (0) 111 111 111
mail:	bob.john@hepia.ch
roomnumber	A408

OIDs (Object Identifiers)

- Les classes d'objets et les attributs
  - Sont normalisés (cf RFCs) afin de garantir l'interopérabilité entre logiciels
  - Sont référencés par un Object Identifier (OID) unique donc la liste est tenue à jour par l'IANA[1]
- Un OID est une séquence de nombres entiers séparés par des points. Les OIDS sont alloués de manière hiérarchique :
  - Seule l'autorité qui a délégation sur la hiérarchie x.y.z peut définir la signification de l'objet x.y.z.t. Par exemple :
    - 2.5 - fait référence au service X.500 (l'ancêtre de LDAP)
    - 2.5.4 - est la définition des types d'attributs
    - 2.5.6 - est la définition des classes d'objets
    - 1.3.6.1 - Internet OID
    - 1.3.6.1.4.1 - OIDs alloués par l'IANA aux entreprises privées

[1] http://fr.wikipedia.org/wiki/Internet_Assigned_Numbers_Authority

Modèle des données LDAP : le schéma

- Le *directory schema* définit l'ensemble des objets qui peuvent être présents dans l'annuaire
- Il décrit les *classes d'objet,* Les types des attributs et leur syntaxe
- Chaque entrée de l'annuaire fait obligatoirement référence à une classe d'objet du schéma et ne doit contenir que des attributs rattachés au type d'objet en question

Contenu

- Introduction
- Modèle des données LDAP
- Espace de nommage LDAP
- Aperçu du protocole

Espace de nommage LDAP

- Chaque entrée à un nom unique : le Distinguished Name (`dn`)
- L'espace de nommage est structuré par un arbre, appelé le *Directory Information Tree* (DIT)
- Chaque noeud de l'arbre est une *entrée*
- La racine de l'arbre est aussi une *entrée*
- Chaque entrée doit être connectée à une entrée déjà existante

Le Distinguished name (dn)

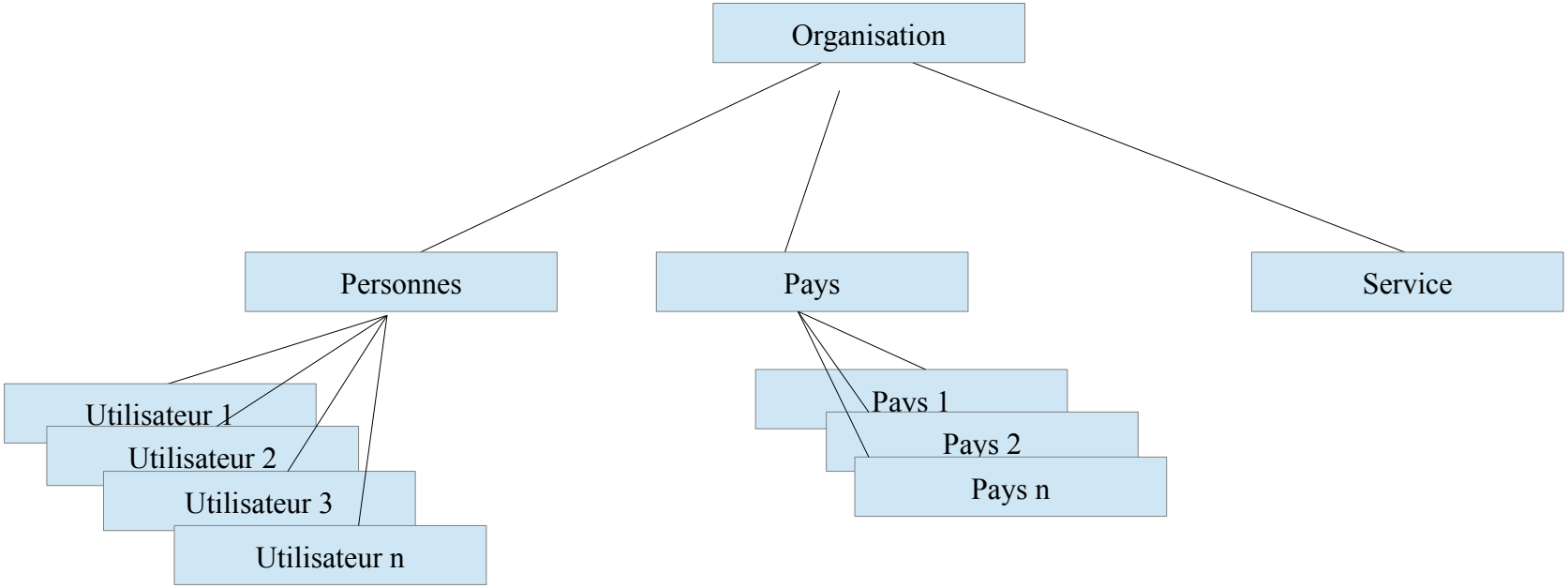
- Un dn est composé de plusieurs Relative Distinguished Names (rdn)

- Un rdn est une paire (nom d'attribut, valeur)

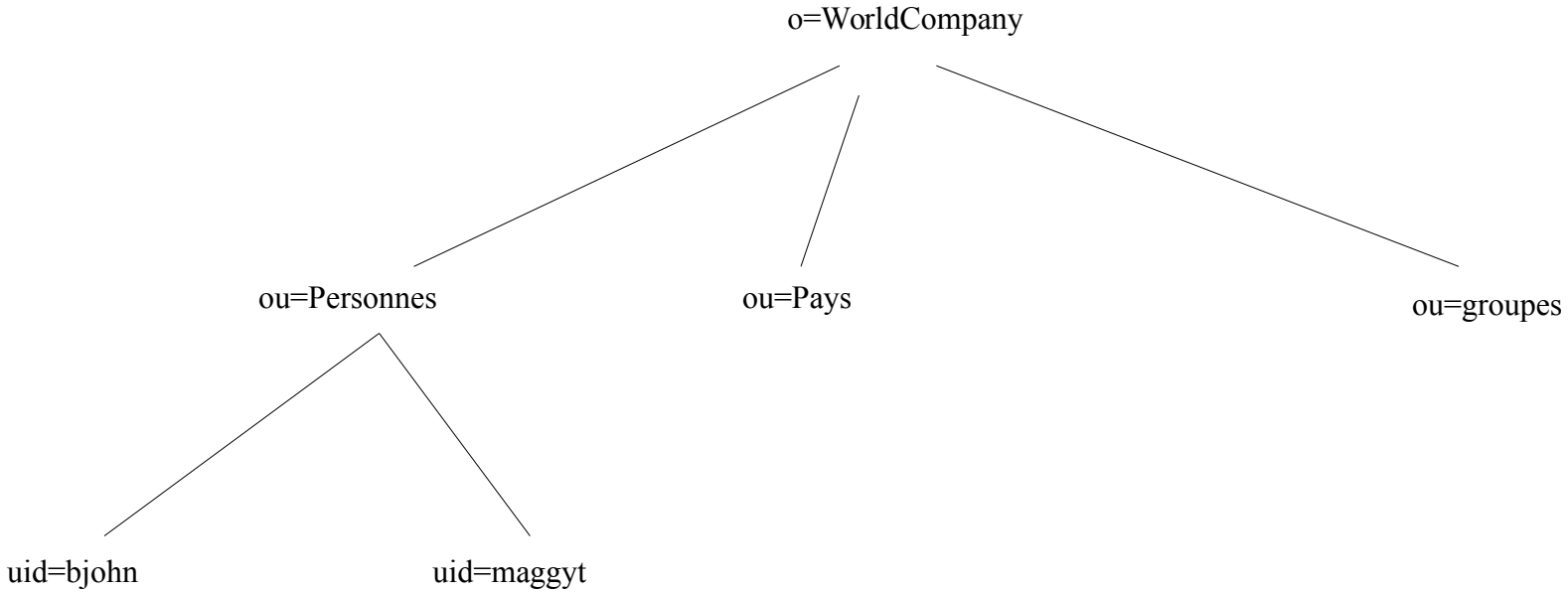
- Exemples de rdn :
  - cn=printer
  - o=Nations Unies
  - c=FR

Le Directory Information Tree (DIT)

- Classifie les entrées dans une arborescence (comparable au système de fichier Unix)



Exemple de DN dans le DIT



- Forme du `dn`: suite des noms des entrées (les `rdns`), en partant de l'entrée elle-même et en remontant vers la racine du DIT, séparées par des ","

- Ex de `dn` dont le `rdn` est `uid=bjohn` :

```
uid=bjohn, ou=Personnes, o=WorldCompany
```

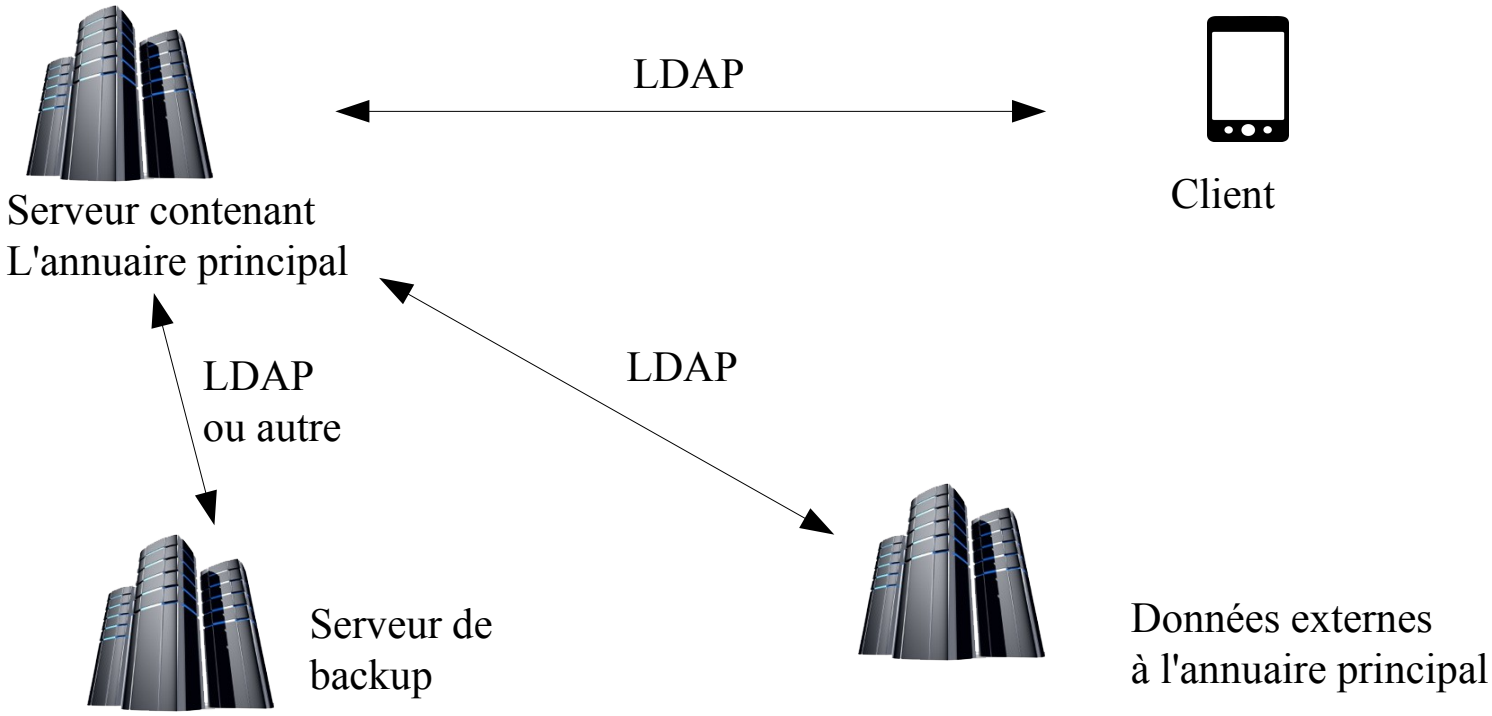
Contenu

- Introduction
- Modèle des données LDAP
- Espace de nommage LDAP
- Aperçu du protocole

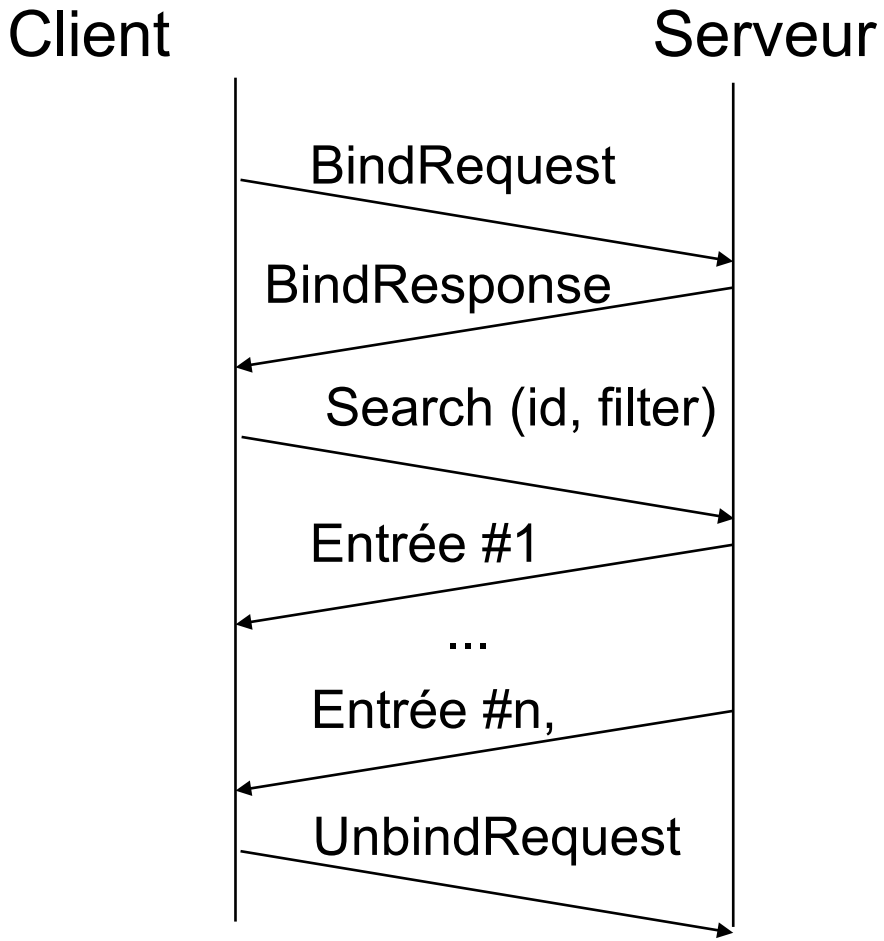
Aperçu du protocole LDAP (RFC 4511)

- Le protocole permet d'accéder à un annuaire via TCP/IP. Il définit :
  - Comment s'établit la communication client-serveur
    - i.e quelles sont les commandes pour se connecter, se déconnecter, pour rechercher, comparer, créer, modifier ou effacer des entrées.
  - Comment s'établit la communication serveur-serveur
    - Échanger leur contenu et le synchroniser, le copier.
    - Créer des liens permettant de relier des annuaires les uns aux autres
  - Le format de transport des données (voir cours sur ASN.1/BER)
    - Décrit en utilisant le standard ASN.1
    - Encodé sur le réseau en utilisant Basic Encoding Rules (BER)
  - Les mécanismes de sécurité
    - Méthodes de chiffrement et d'authentification
    - Mécanismes de règles d'accès aux données

Protocole : infrastructure type



Protocole : ex. de communication client-serveur



- L'opération Bind est optionnelle C'est l'opération d'authentification

- Le client peut envoyer plusieurs requêtes en même temps

- Chaque requête dispose d'un Identifiant

- Une requête peut générer une réponse sur plusieurs messages : il Faut indiquer la fin de la réponse.

Aperçu du protocole (RFC 4511)

- Définition de la syntaxe des messages en ASN.1

```
LDAPMessage ::= SEQUENCE {
             messageID       MessageID,
             protocolOp      CHOICE {
                  bindRequest           BindRequest,
                  bindResponse          BindResponse,
                  unbindRequest         UnbindRequest,
                  searchRequest         SearchRequest,
                  searchResEntry        SearchResultEntry,
                  searchResDone         SearchResultDone,
                  searchResRef          SearchResultReference,
                  modifyRequest         ModifyRequest,
                  modifyResponse        ModifyResponse,
                  addRequest            AddRequest,
                  addResponse           AddResponse,
                  delRequest            DelRequest,
                  delResponse           DelResponse,
                  modDNRequest          ModifyDNRequest,
                  modDNResponse         ModifyDNResponse,
                  compareRequest        CompareRequest,
                  compareResponse       CompareResponse,
                  abandonRequest        AbandonRequest,
                  extendedReq           ExtendedRequest,
                  extendedResp          ExtendedResponse,
                  ...,
                  intermediateResponse  IntermediateResponse },
             controls       [0] Controls OPTIONAL }

MessageID ::= INTEGER (0 ..  maxInt)

maxInt INTEGER ::= 2147483647 -- (2^^31 - 1) --
```

Aperçu du protocole – authentification

- Type BindRequest en ASN.1 :

```
BindRequest ::= [APPLICATION 0] SEQUENCE {
             version                 INTEGER (1 ..  127),
             name                    LDAPDN,
             authentication          AuthenticationChoice }


          LDAPDN ::= LDAPString
                   -- Constrained to <distinguishedName> [RFC4514]


          LDAPString ::= OCTET STRING -- UTF-8 encoded,
                                      -- [ISO10646] characters


          AuthenticationChoice ::= CHOICE {
             simple                  [0] OCTET STRING,
                                     -- 1 and 2 reserved
             sasl                    [3] SaslCredentials,
             ...  }

                   SaslCredentials ::= SEQUENCE {
                        mechanism               LDAPString,
                        credentials             OCTET STRING OPTIONAL }
```

Aperçu du protocole - Recherche

- Type SearchRequest en ASN.1 :

```
SearchRequest ::= [APPLICATION 3] SEQUENCE {
             baseObject      LDAPDN,
             scope           ENUMERATED {
                  baseObject              (0),
                  singleLevel             (1),
                  wholeSubtree            (2),
                  ...  },
             derefAliases    ENUMERATED {
                  neverDerefAliases       (0),
                  derefInSearching        (1),
                  derefFindingBaseObj     (2),
                  derefAlways             (3) },
             sizeLimit       INTEGER (0 ..  maxInt),
             timeLimit       INTEGER (0 ..  maxInt),
             typesOnly       BOOLEAN,
             filter          Filter,
             attributes      AttributeSelection }

Filter ::= CHOICE {
             and             [0] SET SIZE (1..MAX) OF filter Filter,
             or              [1] SET SIZE (1..MAX) OF filter Filter,
             not             [2] Filter,
             ...
```

Pour une spécification complète des messages : se référer au RFC 4511