R304 - DNS : Domain Name Service

Guillaume Urvoy-Keller

October 10, 2022

DNS

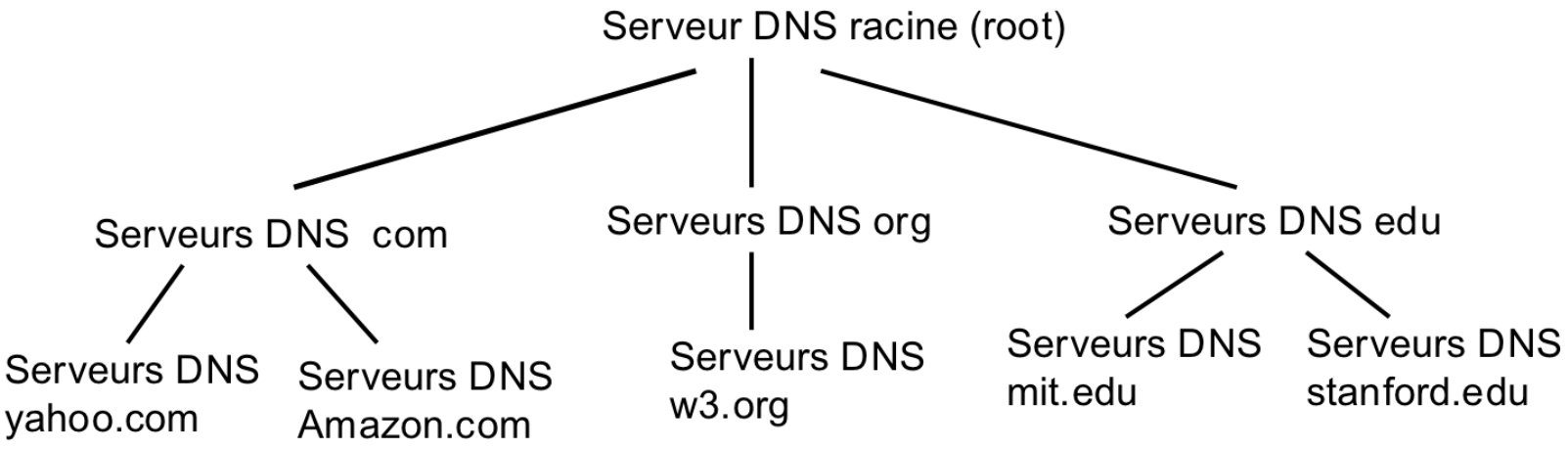
- Base de données distribuée implantée par une hiérarchie de serveurs de noms
- Protocole de niveau applicatif
  - Protocole client/serveur
  - Suite de requêtes/réponses
- Utilise UDP comme couche transport
  - Port : 53
  - UDP est suffisant car requêtes/réponses tiennent dans un datagramme
  - Avec TCP, il faudrait établir la cx $\rightarrow$ perte de temps!
  - Si requête (ou réponse perdue) le client réémet au bout d'un certain temps
- Pleins de clients : nslookup (windows/unix), host, dig

Exemple client : dig

dig montre bien la structure du paquet DNS avec ses différentes parties (question, réponse, etc.)

```
guillaumesmbp2:DNS urvoy$ dig www.unice.fr

; <<>> DiG 9.10.6 <<>> www.unice.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 420
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.unice.fr.                  IN      A

;; ANSWER SECTION:
www.unice.fr.           38902   IN      CNAME   sites.unice.fr.
sites.unice.fr.         38902   IN      A       134.59.204.9

;; Query time: 36 msec
;; SERVER: 192.168.1.1#53(192.168.1.1)
;; WHEN: Mon Jul 16 10:04:03 CEST 2018
;; MSG SIZE  rcvd: 74
```

Une base de données distribuées



Les serveurs roots connaissent les serveurs TLD qui connaissent les serveurs ayant autorités.
Seuls les serveurs ayant autorité contiennent les bases de données des sites $\rightarrow$ c'est ce que l'on veut apprendre à configurer en M3105

Le *Resolver*

Ne fait pas partie de la hiérarchie

Proxy entre les clients locaux et les serveurs DNS

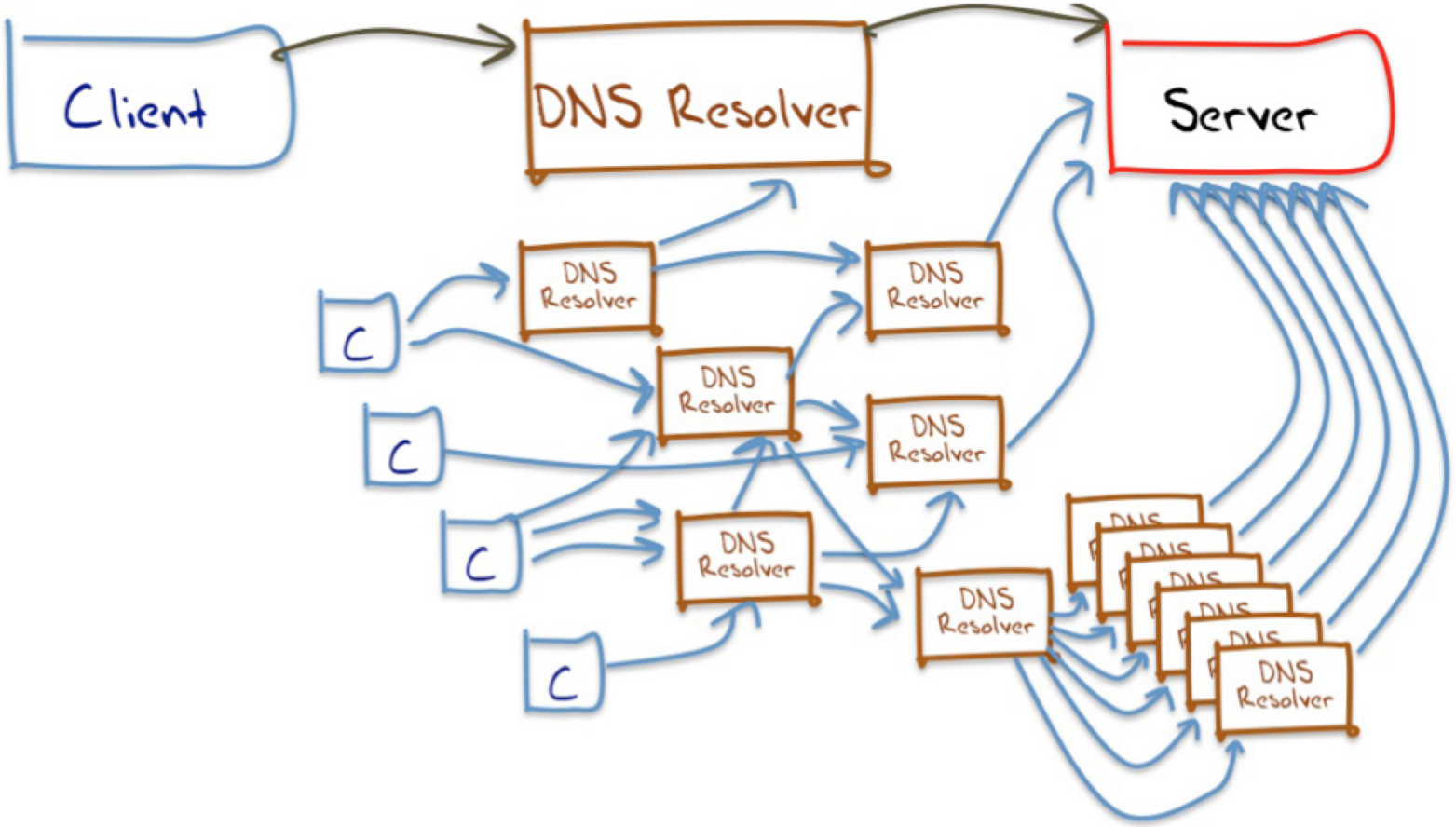


Figure: `https://labs.ripe.net/Members/gih/dnsfig2.png`

Base de donnée d'un serveur ayant autorité

Une compagnie possède

- Des adresses IPs
- Un nom de domaine

donc

Une compagnie possède

- Une zone directe nom $\rightarrow$ adresse IP
- Une zone inverse adresse IP $\rightarrow$ nom

Les zones (directes **et** inverses) sont dans le même arbre de référence.

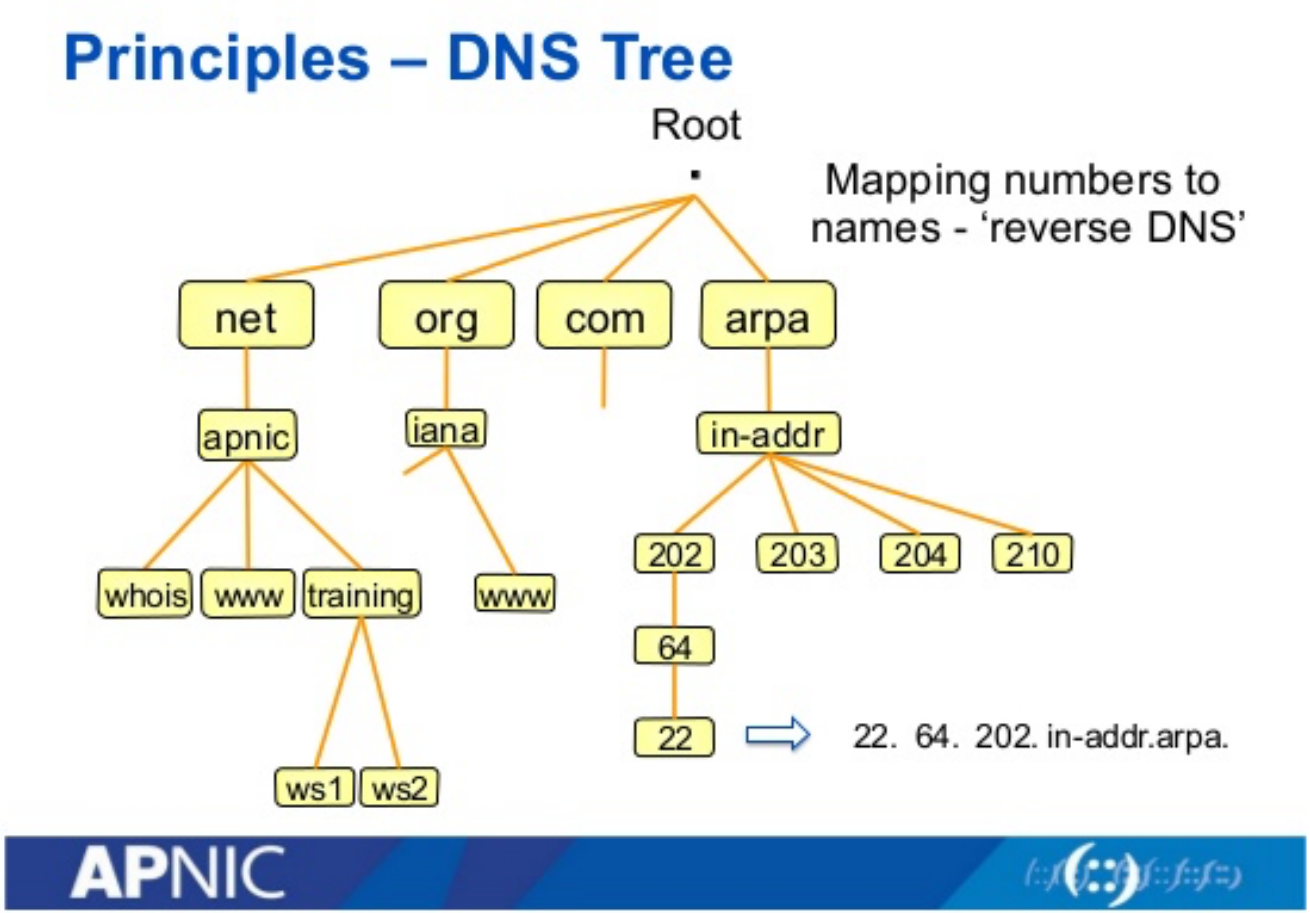


Figure: Source : `https://blog.apnic.net/author/roman/`

Fichier de zone directe

- @ est l'alias du nom de la zone
- Contient des enregistrements. Ex : *nom_machine A adresse_IP* ou *nom_domaine NS nom_serveur_DNS*
- Enregistrement important : le SOA en début de fichier

```
$TTL    86400 ; 24 hours could have been written as 24h or 1d
; $TTL used for all RRs without explicit TTL value
$ORIGIN example.com.
@  1D  IN  SOA ns1.example.com. hostmaster.example.com. (
                                2002022401 ; serial
                                3H ; refresh
                                15 ; retry
                                1w ; expire
                                3h ; minimum
                               )
@       IN  NS     ns1.example.com. ; in the domain
@       IN  NS     ns2.smokeyjoe.com. ; external to domain
@       IN  MX  10 mail.another.com. ; external mail provider
; server host definitions
ns1     IN  A      192.168.0.1  ;name server definition
www     IN  A      192.168.0.2  ;web server definition
```

Fichier de zone inverse. Ex: on suppose ici que c'est 192.168.0/24

Nom de zone inverse est écrit dans le même ordre de priorité que zone directe : example.com et 0.168.192.IN-ADDR.ARPA car ARPA ou .com sont les plus génériques.

Important

- Dans un enregistrement, par exemple : nom_serveur A IP_serveur, si nom_serveur ne finit par un point, le serveur ajoute le nom de la zone!
- Erreur typique : on écrit serveur.example.com qui sera traduit en serveur.example.com.example.com.
- Idem avec les adresses IPs à gauche des enregistrement. Ex: 1 se traduira en 192.168.0.1

Fichier de zone inverse. Ex: on suppose ici que c'est 192.168.0/24

```
$TTL 86400 ; 24 hours, could have been written as 24h or 1d
$ORIGIN 0.168.192.IN−ADDR.ARPA.
@ 1D IN SOA ns1.example.com. hostmaster.example.com. (
                                        2002022401 ; serial
                                        3H ; refresh
                                        15 ; retry
                                        1w ; expire
                                        3h ; minimum
                                       )
; Name servers for the zone − both out−of−zone − no A RRs required
          IN NS ns1.example.com.
          IN NS ns2.smokeyjoe.com.
; server host definitions
1 IN PTR ns1.example.com.
2 IN PTR www.example.com.
; non server domain hosts
3 IN PTR bill.example.com.
4 IN PTR fred.example.com.
```

Relation nom de zone et fichier de zone

Le nom de la zone est décrit dans un le fichier de config qui pointe vers le fichier de zone correspondant

```
// BIND configuration file

options {
        directory "/var/named";
        // Place additional options here.
};
zone "movie.edu" in {
        type master;
        file "db.movie.edu";
};

zone "249.249.192.in-addr.arpa" in {
        type master;
        file "db.192.249.249";
};

zone "253.253.192.in-addr.arpa" in {
        type master;
        file "db.192.253.253";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127.0.0";
};

zone "." in {
        type hint;
        file "db.cache";

};
```

R304 - LDAP : Lightweigth Directory Access Protocol

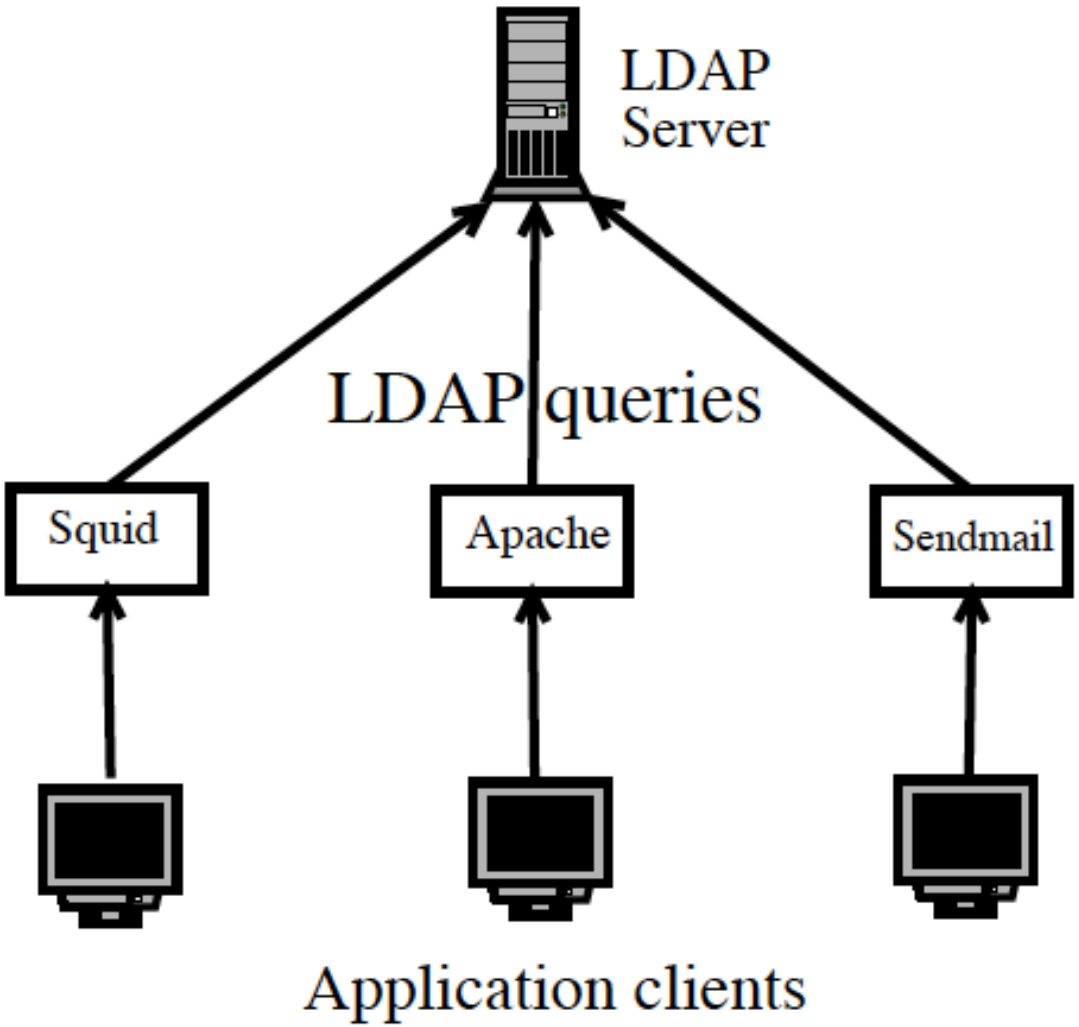
Guillaume Urvoy-Keller

October 10, 2022

Introduction

- Entreprises ont besoin de distribuer des données :
  - Email
  - Téléphone
  - Comptes informatique
- Solution incontournable : LDAP – Lightweigth Directory Access Protocol
- Standardisé par IETF
- Dans le monde Windows, Active Directory utilise LDAP pour gérer les machines et les comptes
- Les applications ont aussi besoin de vérifier les comptes... et donc d'accéder à LDAP

Besoin d'accéder à l'annuaire et/ou authentifier



Exemple d'informations stockées

Un exemple typique d'une entrée dans /etc/passwd exprimée sous forme LDAP

```
uid: ghopper
cn: Grace Hopper
userPassword: {crypt}$1$pZaGA2RL$MPDJoc0afuhHY6yk8HQFp0
loginShell: /bin/bash
uidNumber: 1202
gidNumber: 1202
homeDirectory: /home/ghopper
```

Exemple d'informations stockées

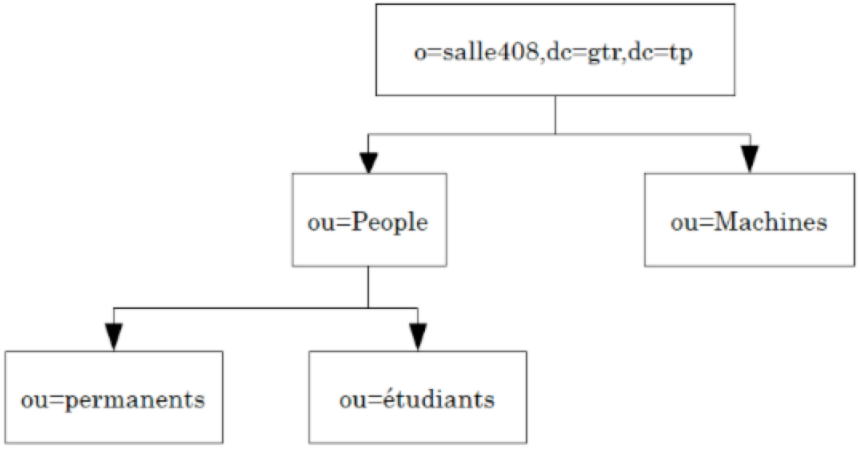
Dans l'exemple précédent :

- Les données sont au format LDIF
- Les données sont stockées dans la base LDAP sous une forme différente (binaire...)
- Mais la possibilité de convertir dans les deux sens "Base de données $\leftrightarrow$ LDIF" est simple et permet échange entre serveurs LDAP

Les entrées LDAP

- Les entrées LDAP sont organisées de manière hiérarchique avec un **identifiant unique** par entrée appelé **dn**
- Données organisées en arbre $\rightarrow$ dn donne chemin dans l'arbre
- Racine de l'arbre : dc=navy, dc=mil :
  - schéma typique où on utilise le domaine DNS pour définir la racine, ici navy.mil

Base LDAP "complète"




```
dn: o=salle408,dc=RT,dc=tp
objectClass: top
objectClass: organization
o: salle408

dn: cn=admin,o=salle408,dc=RT,dc=tp
cn: admin
objectClass: organizationalRole
objectClass: top
objectClass: simpleSecurityObject
description: administrateur LDAP
userPassword: test

dn: ou=Machines,o=salle408,dc=RT,dc=tp
ou: Machines
objectClass: organizationalUnit
objectClass: top

dn: ou=People,o=salle408,dc=RT,dc=tp
ou: People
objectClass: organizationalUnit
objectClass: top

dn: ou=permanents,ou=People,o=salle408,dc=RT,dc=tp
ou: permanents
objectClass: organizationalUnit
objectClass: top

dn:: b3U9w6l0dWRpYW50cyxvdT1QZW9wbGU
        sbzlzYWxsZTQwOCxkYzlSVCxkYz10cA==
ou:: w6l0dWRpYW50cw==
objectClass: organizationalUnit
objectClass: top
```

6 dn $\Rightarrow$ 6 objets (certains écrits en base64)

Construction d'une base

On fait le schéma de la base voulue ....comme on veut

- Schéma appelé DIT : Directory Information Tree

Par contre, un objet de la base hérite de classes d'objets pré-existantes (on ne peut pas en définir!)

- Attributs obligatoires
- Attributs optionnels
- Notion d'héritages entre classes

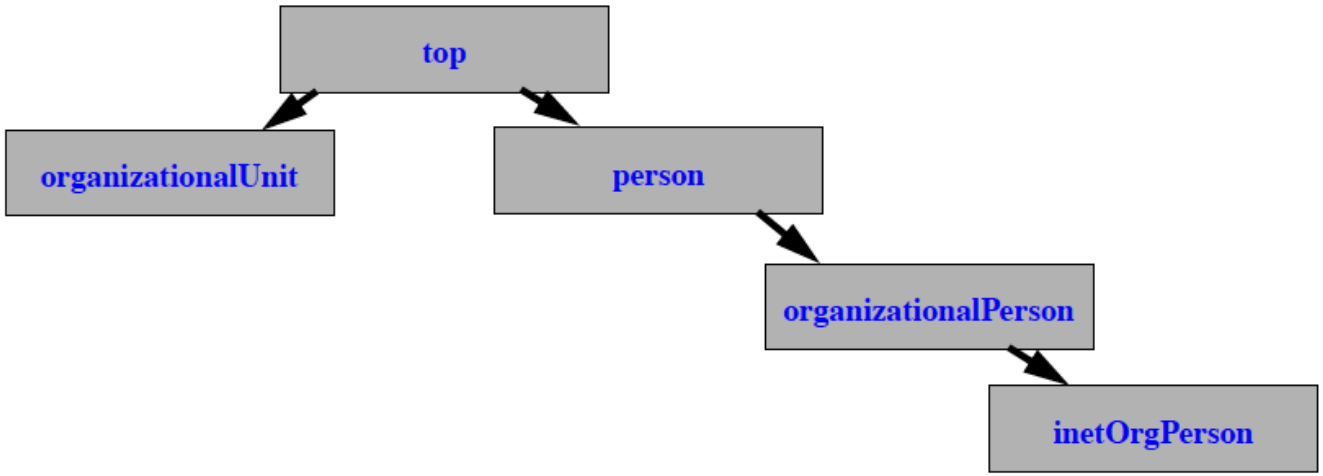
Classes d'objets

Exemples de classes d'objets

Entry Type	Required Attributes	Optional Attributes
inetOrgPerson (defines entries for a person)	•commonName (cn) •surname (sn) •objectClass	•businessCategory •carLicense •departmentNumber • description •employeeNumber •facsimileTelephone •Number •givenName •mail •manager •mobile •organizationalUnit (ou) •pager •postalAddress •roomNumber •secretary •seeAlso •telephoneNumber •title •labeledURI •uid
organizationalUnit (defines entries for organizational units)	•ou •objectClass	•businessCategory •description •facsimileTelephoneNumber •location (l) •postalAddress •seeAlso •telephoneNumber

Héritage entre classes

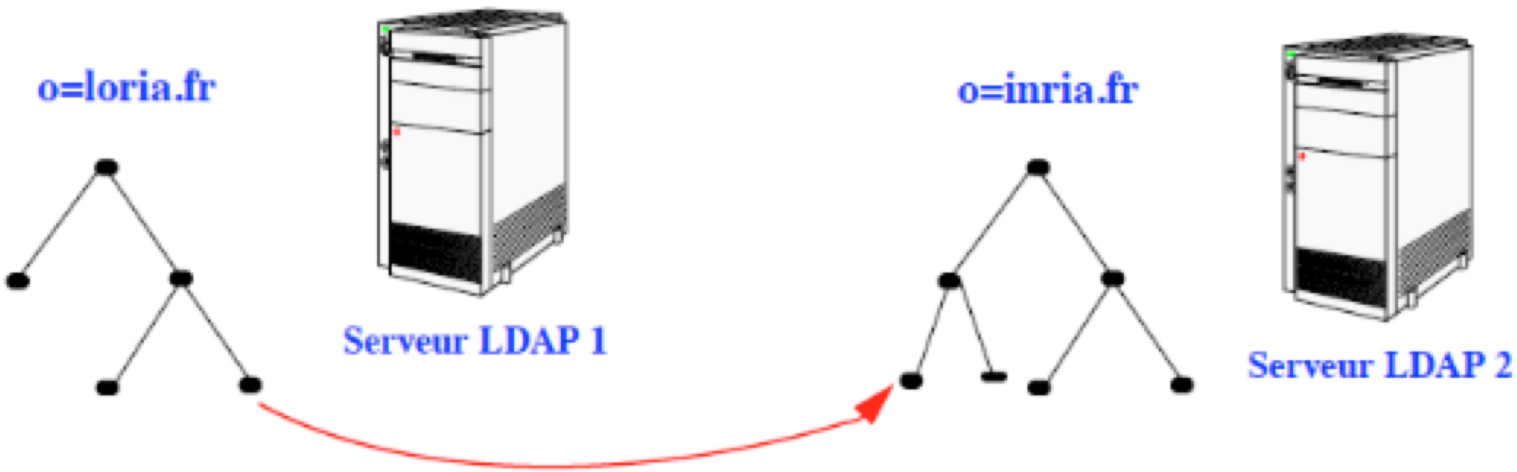
Les classes d'objets forment une hiérarchie, au sommet de laquelle se trouve l'objet `top`.



- **Chaque objet hérite des propriétés (attributs) de l'objet dont il est le fils.**
- **On précise la classe d'objet d'une entrée à l'aide de l'attribut `objectClass`.**
- **Il faut obligatoirement indiquer la parenté de la classe d'objet en partant de l'objet `top` et en passant par chaque ancêtre de l'objet.**

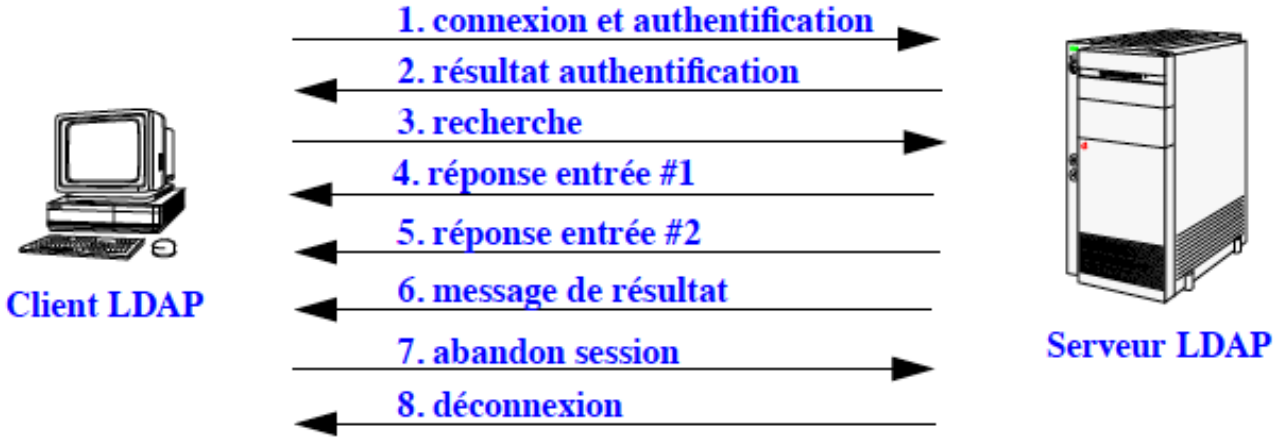
La base peut être distribuée - entreprise multi-sites

Le DIT est découpé en sous-arbre, chacun géré par un serveur distinct



Procole LDAP

Il faut se connecter avant d'accéder aux données



Authentification

L'authentification

LDAP est un protocole avec connexion : il faut s'authentifier pour ouvrir la connexion (`bind`) en fournissant une identité.

LDAPv3 propose plusieurs choix d'authentification :

- ***Anonymous authentification*** **- accès sans authentification permettant de consulter les données accessibles en lecture pour tous.**
- ***Root DN authentification*** **- accès administrateur (tous les droits).**
- ***Mot de passe en clair*** **- un DN plus un password qui transite en clair sur le réseau.**
- ***Mot de passe + SSL ou TLS*** **- la session est chiffrée et le mot de passe ne transite plus en clair.**
- ***Certificats sur SSL*** **- échange de certificats SSL (clefs publiques/privées).**
- ***Simple Authentification and Security Layer (SASL)*** **- mécanisme externe d'authentification.**

Contrôle d'accès (aux données)

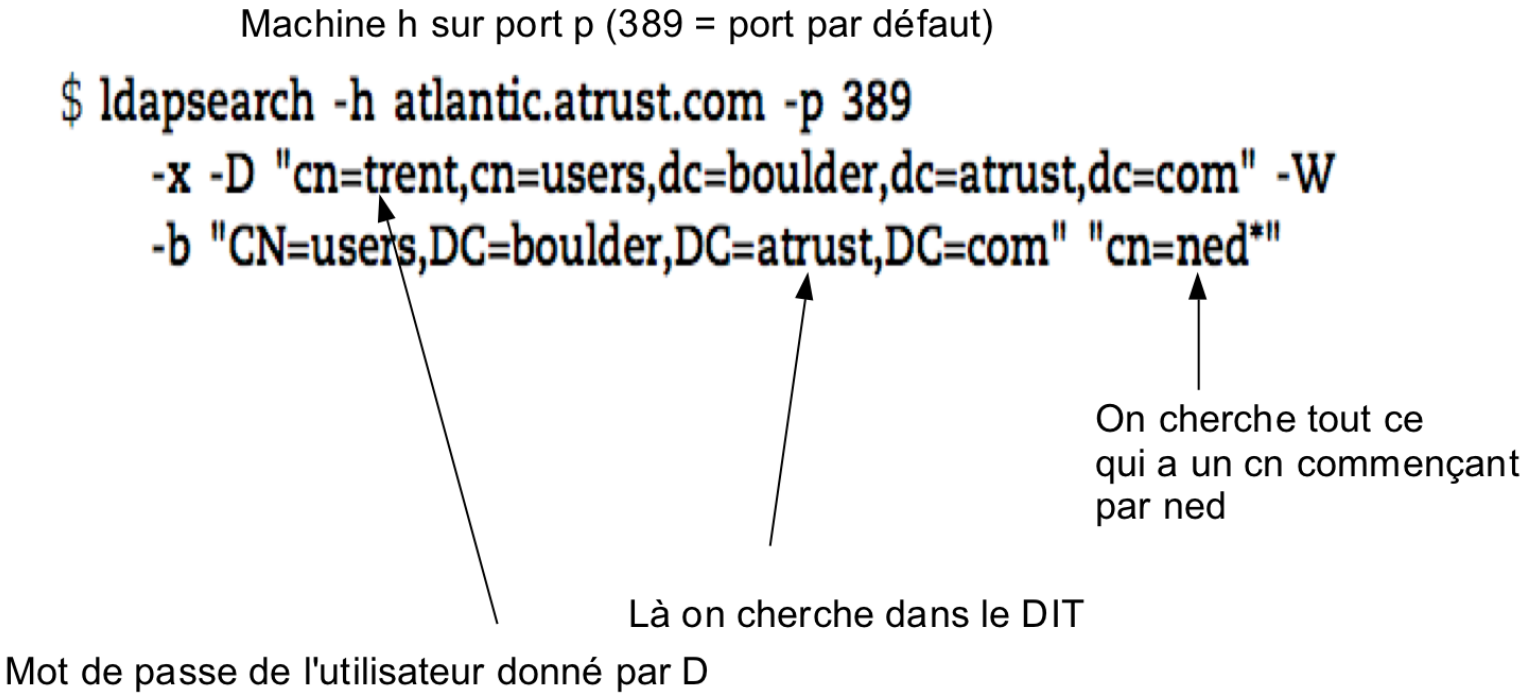
Une fois authentifié, on peut controller l'acccès aux données, c.a.d. ce que l'utilisateur voit : tout ou partie du DIT, en lecture, écriture, etc.

Ces ACLs s'expriment sous la forme canonique :

```
<target> <permission> <bind rule>
        <target> : point d'entrée de l'annuaire auquel s'applique la règle
        <permission> : permet ou refuse un type d'accès (lecture, écriture...)
        <bind rule> : identifie le bindDN utilisé en connexion
```

<permissions>	<bind rules>
Read	Un utilisateur
Write	Un groupe d'utilisateur
Search	
Compare	
Selfwrite	
Add	
Delete	

Exemple d'accès aux données avec utilitaire ldapsearch



Comment un utilitaire, par ex. ssh, est raccordé à LDAP?

PAM: Pluggable Authentication Module

- Idée derrière PAM : une librairie dynamique pour les commandes/démons et un fichier de config par commande/démon. Délégation de l'authentification
- Seul le fichier de config doit être modifié, aucune recompilation!
- Comment savoir si une commande est pam-ifiée : ldd nom_commande | grep pamlib.so

```
root@stretch:/home/vagrant# ldd /usr/bin/passwd
        linux-vdso.so.1 (0x00007ffc6b282000)
        libpam.so.0 => /lib/x86_64-linux-gnu/libpam.so.0 (0x00007fdd6a421000)
        libpam_misc.so.0 => /lib/x86_64-linux-gnu/libpam_misc.so.0 (0x00007fdd6a21d000)
        libaudit.so.1 => /lib/x86_64-linux-gnu/libaudit.so.1 (0x00007fdd69ff5000)
        libselinux.so.1 => /lib/x86_64-linux-gnu/libselinux.so.1 (0x00007fdd69dcd000)
        libc.so.6 => /lib/x86_64-linux-gnu/libc.so.6 (0x00007fdd69a2e000)
        libdl.so.2 => /lib/x86_64-linux-gnu/libdl.so.2 (0x00007fdd6982a000)
        libcap-ng.so.0 => /lib/x86_64-linux-gnu/libcap-ng.so.0 (0x00007fdd69624000)
        libpcre.so.3 => /lib/x86_64-linux-gnu/libpcre.so.3 (0x00007fdd693b1000)
        /lib64/ld-linux-x86-64.so.2 (0x00007fdd6a83e000)
        libpthread.so.0 => /lib/x86_64-linux-gnu/libpthread.so.0 (0x00007fdd69194000)
```

PAM : fichier de configuration

- Stockés dans le repertoire /etc/pam.d.
- Un fichier contient 3 parties en général : passwd, auth, session, account

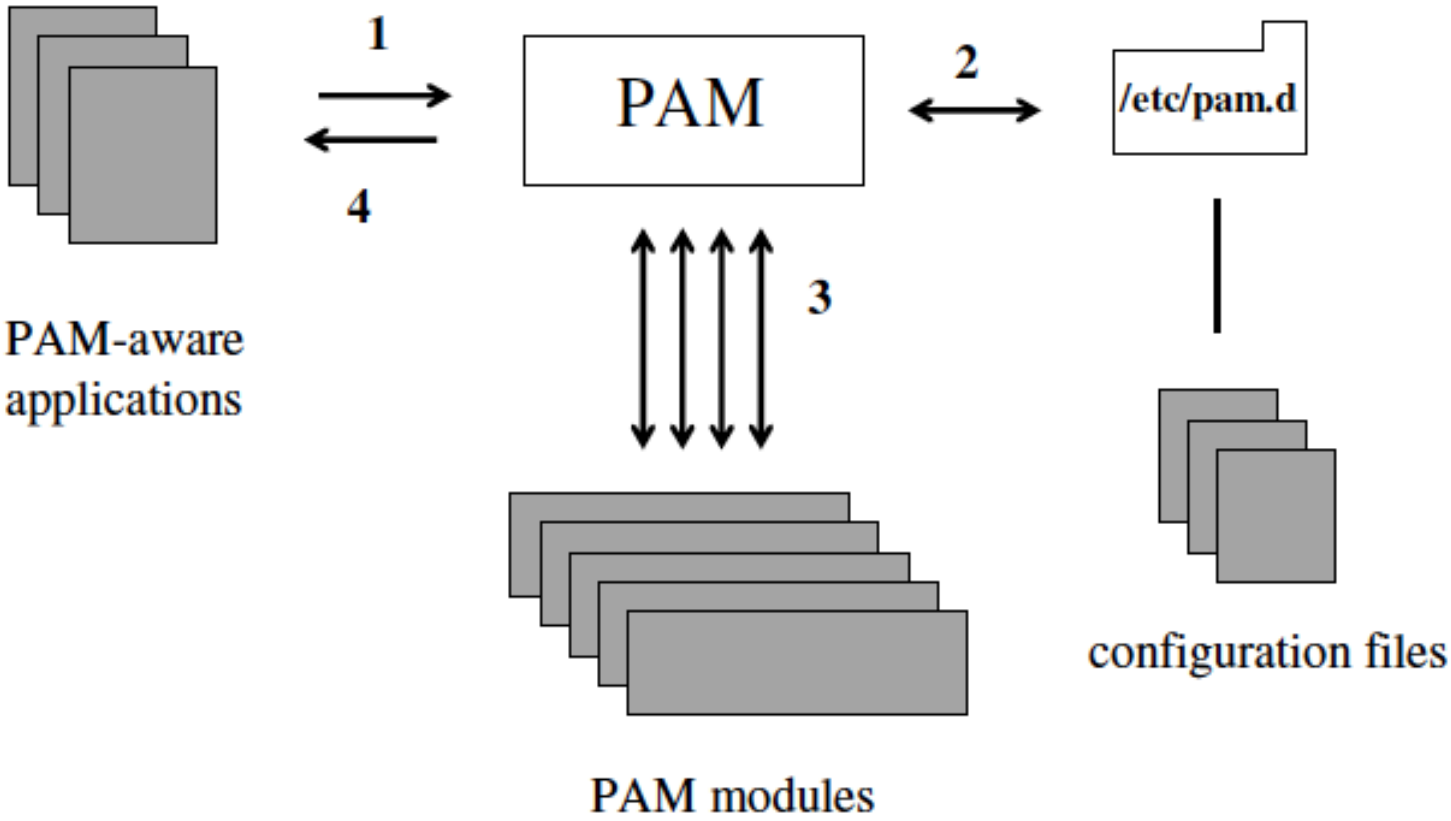
  - Passwd : changement de mot de passe
  - Auth : vérification du couple (nom, mdPasse)
  - Account : vérification des éléments du compte (horaires, droit, ...)
  - Session : mise en place de la session

PAM : fichier de configuration

- Chaque partie contient des lignes du type
  auth sufficient pamrootok.so
- En fin de ligne, on trouve un module PAM qui va retourner :
  succès ou erreur
- Le « control-flag » qui vaut sufficient, required,requisite, optional, etc
- A chaque invocation, retour d'une seule valeur : succès ou erreur

```
guillaumesmbp2:pam.d urvoy$ ls
authorization           authorization_lacont    login                   screensaver
authorization_aks       checkpw                 login.term              screensaver_
authorization_ctk       chkpasswd               other                   screensaver_
authorization_la        cups                    passwd                  screensaver_
guillaumesmbp2:pam.d urvoy$ more passwd
# passwd: auth account
auth       required       pam_permit.so
account    required       pam_opendirectory.so
password   required       pam_opendirectory.so
session    required       pam_permit.so
```

Architecture PAM



NSswitch

- Name Service Switch
- Wikipedia : autorise le remplacement des traditionnels fichiers Unix de configuration (par exemple /etc/passwd, /etc/group, /etc/hosts) par une ou plusieurs bases de données centralisées (ldap, etc)

NSswitch

```
root@stretch:/etc/pam.d# more /etc/nsswitch.conf
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference' and `info' packages installed, try:
# `info libc "Name Service Switch"' for information about this file.

passwd:         compat
group:          compat
shadow:         compat
gshadow:        files

hosts:          files dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

Compat = chercher les logins dans les fichiers classiques /etc/passwd et /etc/shadow