

CH4 Authentification et annuaires

Utilisations

- Login
- Carnet d'adresses distribué
- Flexible: liste de ressources, ...


PAM

- **PAM** = *Pluggable Authentication Modules*
- Bibliothèques de fonctions que les applications peuvent utiliser pour demander l'authentification d'un utilisateur
- Permet de rendre les applications **indépendantes** du schéma d'authentification
 - /etc/passwd, shadow passwords, token, kerberos, LDAP...
- Présent dans toutes les distributions de Linux

LDAP

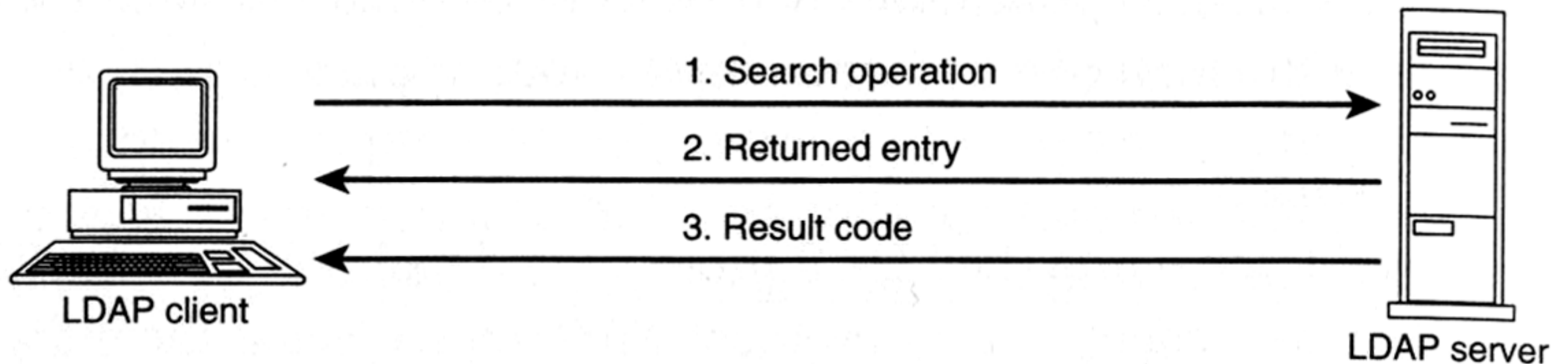
Annuaire vs bases de données

– Concept commun, le stockage de données mais...

- Ratio read/write 
- Extensibilité
- Niveau de distribution
- Réplication
- Performance
- Standard

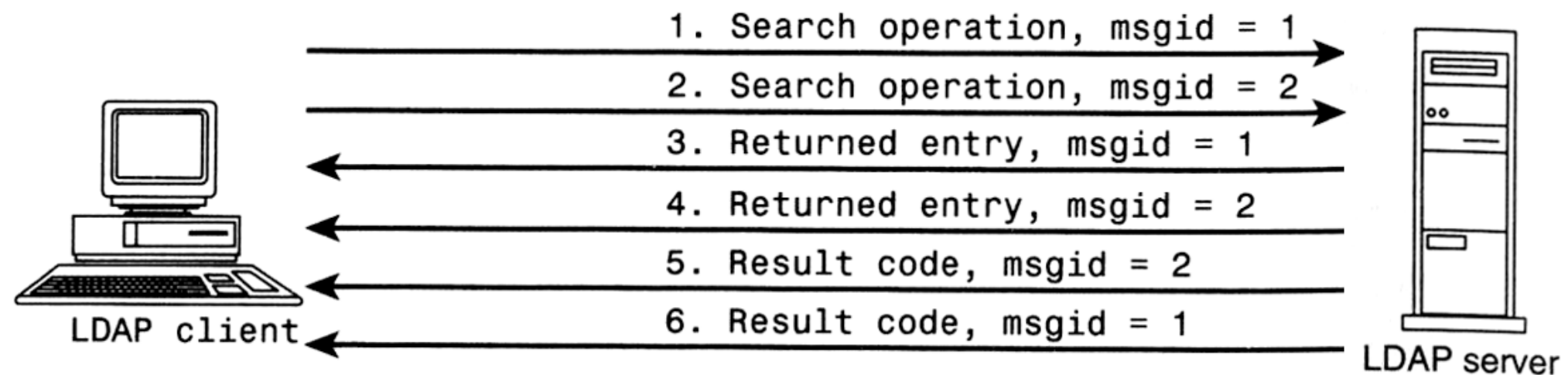
Modèle client/serveur (1/2)

- Modèle client/serveur
- TCP 389
- Protocole orienté messages
- Code de résultat

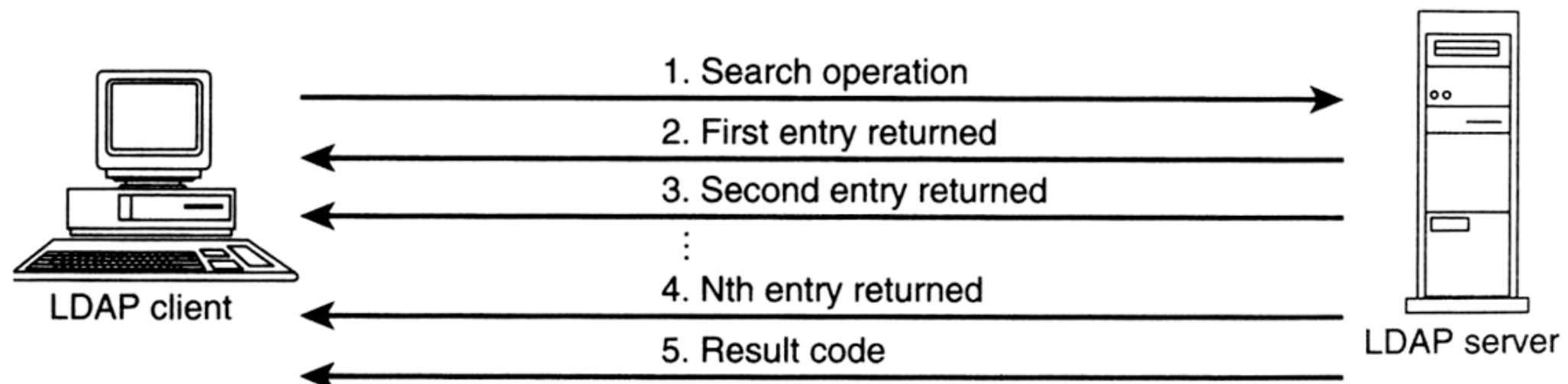


Modèle client/serveur (2/2)

→ requêtes multiples asynchrones (\neq HTTP)



→ réponses multiples



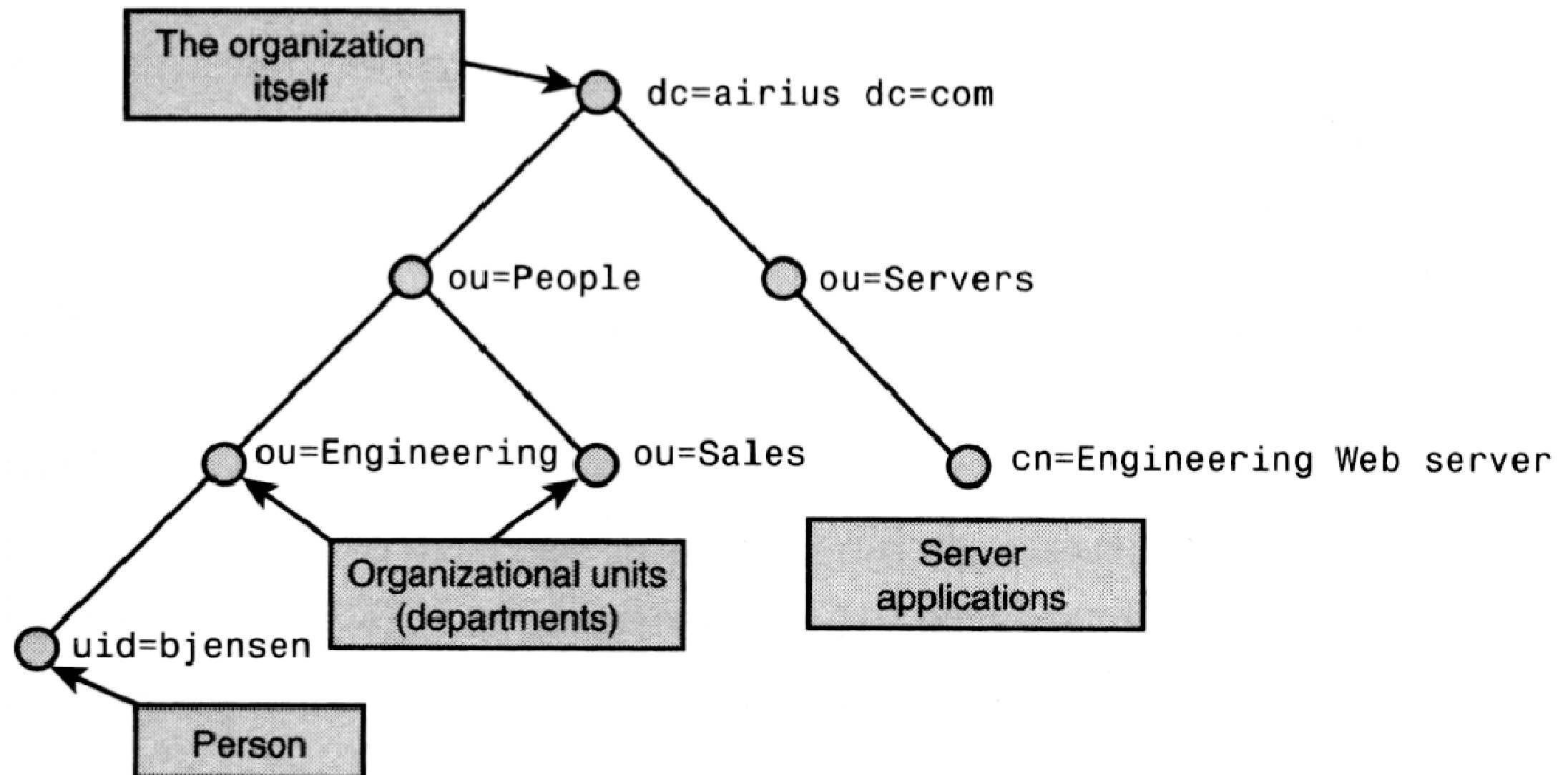
Opérations du protocole

- 9 opérations de base
 - Opérations d'**interrogation**
 - search, compare
 - Opérations de **mise à jour**
 - add, delete, modify, modify DN
 - Opérations d'**authentification** et de **contrôle**
 - bind, unbind, abandon

Le modèle d'information (1/5)

- **Définit les données et les unités d'information de base stockables dans un annuaire LDAP**
- DIT – structure arborescente
- Entries = objets
- **Objets** : notion de classes
- Les objets possèdent des **attributs**
- Chaque attribut a un **type** (**nom**) et une **valeur**
- La **syntaxe** d'un attribut est définie (ex. **caseIgnoreString**) et utilisée dans comparaisons (search, compare → matching rules)

Le modèle d'information (2/5)



Le modèle d'information – schéma (3/5)

- **Schéma** = définition des **classes**, c.à.d. des attributs de chaque type d'objet
- Exemple : **objectClass = person** implique la présence des attributs **cn** (common name) et **sn** (surname)
- Certains attributs sont **obligatoires**, d'autres **optionnels** ; certains sont **mutivalués**
- **Forme d'héritage multiple**
 - Une entrée peut posséder des attributs de plusieurs classes

Le modèle d'information – schéma (4/5)

```
dn: uid=rdevil, dc=ulb, dc=be
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Raymond Devillers
cn: Ray Devillers
sn: Devillers
mail: rdevil@ulb.ac.be
telephoneNumber: +32 650 12 34
description: professeur ordinaire
```

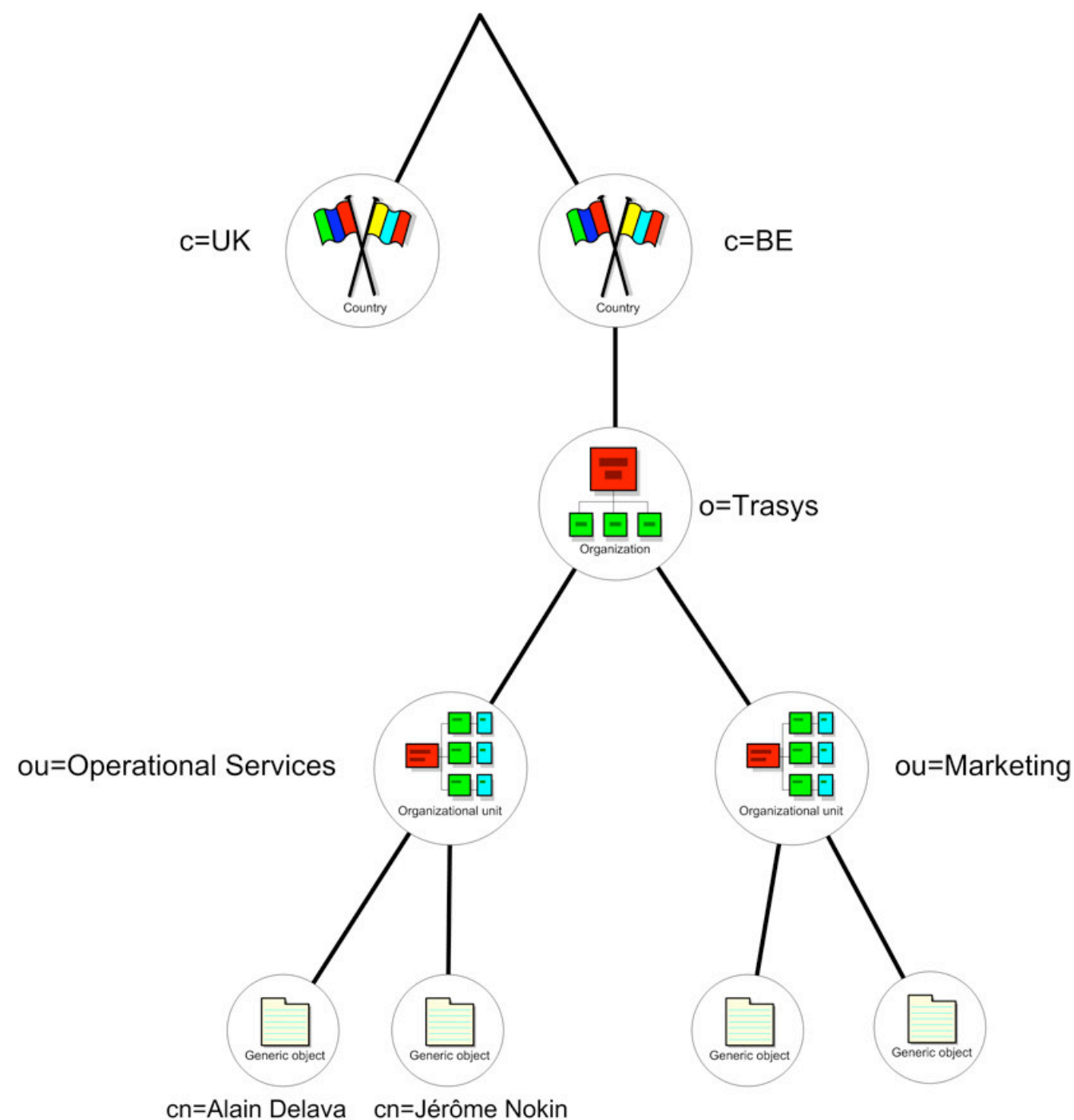
Le modèle de nommage (1/5)

- **Définit la manière dont les données sont organisées et référencées**
- DIT – structure arborescente semblable à un filesystem Unix mais...
 - **root conceptuelle** (ne contient pas d'information)
 - **containers** : pas de différence, ce sont des nœuds/**objets comme les autres** (mais par exemple de classe `organizationalUnit`)

Le modèle de nommage (2/5)

- DIT – structure arborescente semblable à un filesystem Unix mais...
 - **nommage** des nœuds « **de bas en haut** », c.à.d. du nœud à la racine (cf. DNS)
- **Distinguished Name (DN)**
 - `cn=Alain Delava, ou=Operational Services, o=Trasys, c=be`
 - Chaque entrée doit avoir un DN **unique**
 - Utilisé e.a. pour les opérations de mises à jour

Le modèle de nommage (3/5)



**cn=Alain Delava, ou=Operational Services,
o=Trasys, c=be**

Le modèle de nommage (4/5)

- **Distinguished Name (DN)**
 - `cn=Alain Delava, ou=Operational Services, o=Trasys, c=be`
- **Relative Distinguished Name (RDN)**
 - Partie la plus à gauche du DN : `cn=Alain Delava`
 - **Unicité locale** (i.e. parmi les nœuds de même parent)
 - Peut être **mutivalué** (permet d'avoir deux entrées avec, dans notre exemple, le même **cn**)
 - `cn=John Smith + mail=jsmith@company.com`

Le modèle fonctionnel

- **Définit les opérations pouvant être effectuées sur les informations**
- search, compare
- add, delete, modify, modify DN (rename / move)
- bind, unbind, abandon

Search – filtres

- Equality filters (**sn=Delava**)
- Substring filters (**sn=Van Den***)
- Approximate filters (**sn=~Delawa**)
- Comparison filters (<,>,<=,>=)
 - nécessite un ordre (numérique, lexicographique...)
- Presence filters (**telephoneNumber=***)
- Combinaisons : & (and) - | (or)
- Not : !
- (**& (sn=Delava) (givenname=Alain))**
(| (sn=Devillers) (sn=Roggeman))
(& (mail=*) (! (telephoneNumber=*))))

Le modèle de sécurité

- Protection des informations
 - Accès en lecture, en modification
 - Qui peut modifier quelles entrées, dans quelles branches... ?
 - Qui peut modifier quels attributs dans un objet... ?
- **LDAP v3 ne définit pas de processus standard de contrôle d'accès !**
 - ACL etc. dépendants de l'implémentation

LDIF

- **LDAP Data Interchange Format**
 - Permet de représenter des objets dans un format **texte**
 - Utile pour export/import entre serveurs LDAP
 - Deux sortes de fichiers LDIF
 - **Décrivant des objets**
 - Contenant des **opérations de mise à jour** à appliquer sur des entrées d'un annuaire LDAP

LDIF update statements

- Fichier LDIF contenant une liste de modifications
- Généralement utilisé comme input de l'outil `ldapmodify`
- Update statement =
 - **DN** de l'entrée à modifier + **change type**
[+ **modifytype** dans le cas du *modify*]
[+ ensemble des **modifications**]

Rajout d'une entrée

`dn: <dn of entry to be added>`

`changetype: add`

`<attribute1 name>: value`

`<attribute2 name>: value`

`...`

Outils

- Fournis avec OpenLDAP, avec Netscape Directory Server (entre autres)
 - `ldapsearch`
 - `ldapadd`
 - `ldapdelete`
 - `ldapmodify`
 - `ldapcompare`
 - `ldappasswd`
 - `ldapmodrdn`
 - et d'autres

NIS - Yellowpages

NIS - Yellowpages

- Partage de comptes/groupes entre ordinateurs, plus généralement partage de fichiers de configuration
 - Serveur
 - Clients
 - Ajoute virtuellement les entrées à `/etc/passwd`
 - Intégration via PAM

/etc/nsswitch.conf

- Demo

ypcat

- Simule la lecture d'un fichier local, combiné avec les instructions NIS

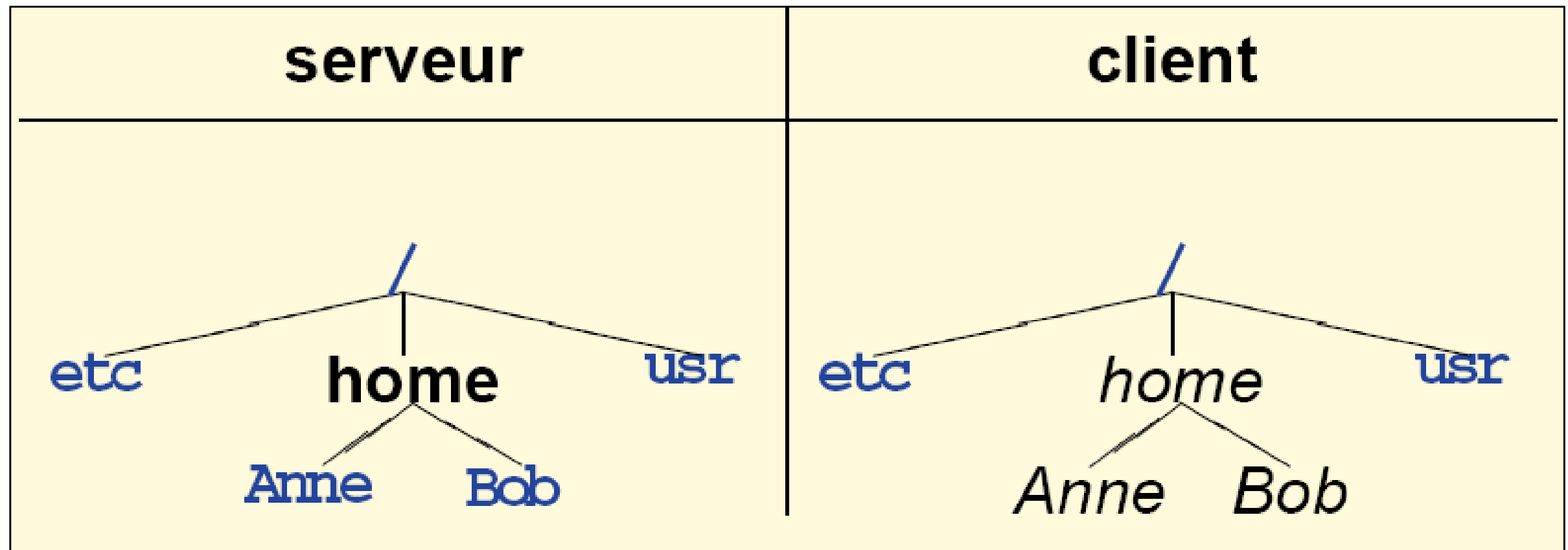
Problème

- Partager les comptes
 - N'implique pas le partage de données
 - NFS
 - Mais implique une structure similaire
 - inhérente au contenu de `/etc/passwd`

NFS

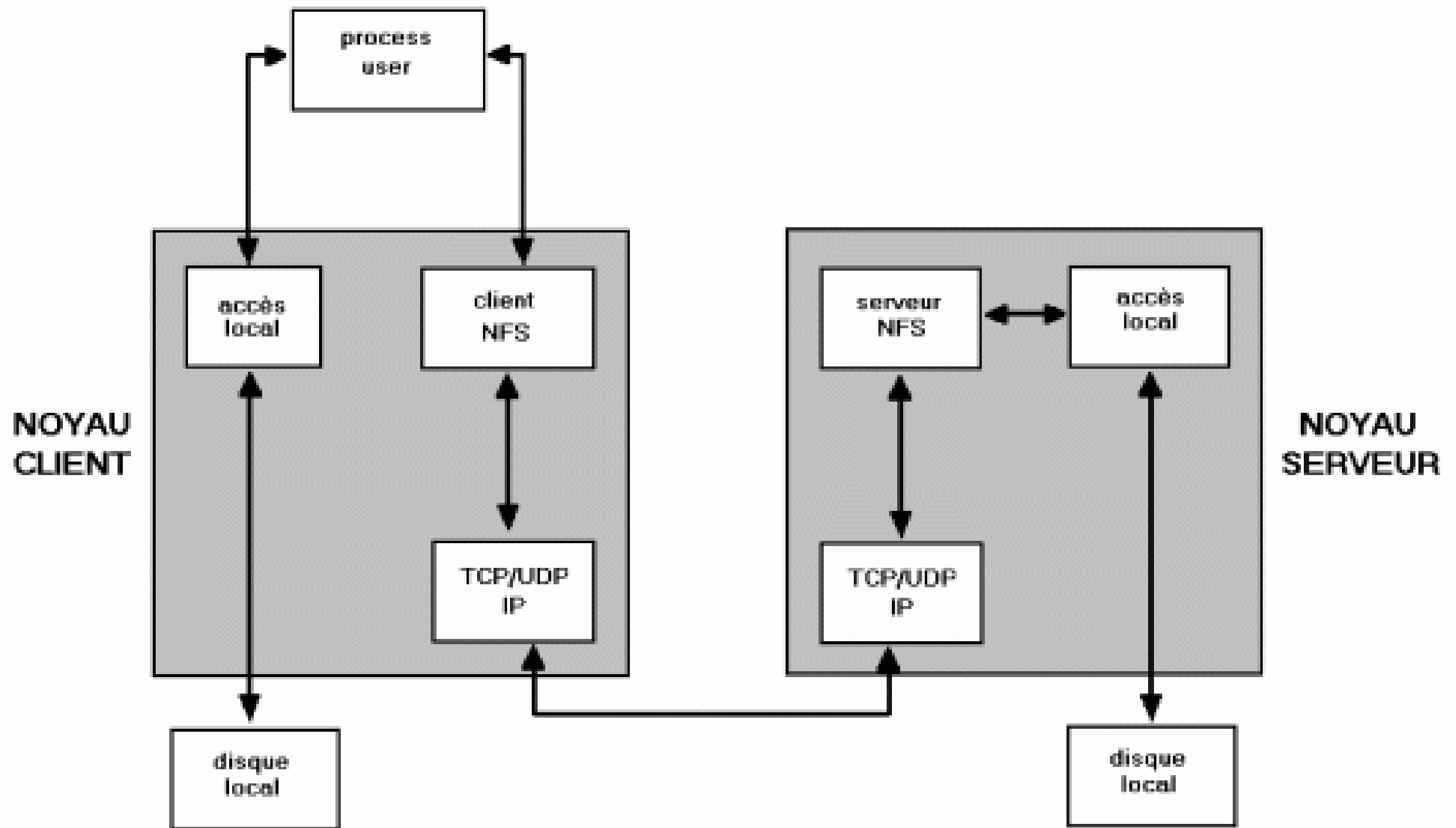
- **NFS = Network File System**
- Permet d'accéder aux fichiers d'un serveur distant
 - En montant un(e partie de) file system distant dans la structure du filesystem local
 - De manière transparente pour les applications
- Implémenté dans le noyau
 - Pour la transparence
 - Pour la performance (évite de passer fréquemment du mode user au mode kernel)

NFS



- Le serveur exporte **/home**
- Le client monte explicitement ce dossier
- Ensuite tout est transparent

NFS



NFS

Lorsqu'un processus client demande un accès à un fichier:

1. il sollicite le **kernel** (system call)
2. le kernel détermine si le fichier est **local ou non**
3. si le fichier est **distant**, le kernel passe au **client NFS** les références du fichier distant
4. le client NFS contacte le **portmapper** du **serveur** concerné
5. le portmapper du serveur met le **client** en relation avec le **serveur NFS**
6. le **serveur NFS** opère la requête via les routines d'accès aux **fichiers locaux**

NFS

- Daemons côté serveur
 - `portmap`
 - `nfsd` : traite les requêtes des clients d'accès aux fichiers
 - `mountd` (`rpc.mountd` sous Linux) : traite les requêtes des clients pour les montages
- Client côté client
 - *`mount`*
- Daemons côté serveur et client
 - `rpc.lockd`
 - `rpc.statd`

NFS – configuration serveur

- `/etc/exports`
- `FS host1(options) ... hostN(options)`
 - `FS` = le filesystem / répertoire à exporter
 - `hostX` : hostname, IP, subnet, wildcards
- Options
 - `ro, rw, rw=list`
 - `root_squash` : `uid=0,gid=0` → `anonuid,anongid`
 - `no_root_squash`
 - `all_squash`
 - `noaccess`
 - ...

NFS – configuration serveur

```
# /etc/exports
```

```
...
```

```
/home 172.19.3.0/24(ro,root_squash) gollum(rw)
```

```
...
```

NFS – côté client

- **Montage manuel**

`mount -t nfs server:/dir /mountpoint`

- **Options**

- ro
- rw (doit être exporté en rw)
- bg
- hard / soft
- noexec
- intr / nointr
- retrans=n
- timeo=m
- ...

NFS – côté client

- Montage automatique

```
# /etc/fstab
```

```
...
```

```
remote_host:/rep_dist rep_local nfs 0 0
```

```
...
```

NFS – divers

- Statistiques
 - `nfsstat -c`
 - `nfsstat -s`
- Sécurisation
 - Protection du client (qui n'a pas confiance en le serveur) : `nosuid`, `noexec`
 - Protection du serveur (qui n'a pas confiance en le client) : `root_squash`
 - Restrictions au niveau de portmap
 - `/etc/hosts.deny` et `/etc/hosts.allow`
- Montage automatique (*automount*, *amd*)