

# Introduction à l'administration des systèmes & réseaux

Christian Bulfone

[christian.bulfone@gipsa-lab.fr](mailto:christian.bulfone@gipsa-lab.fr)

[www.gipsa-lab.fr/~christian.bulfone/MIASHS-DCISS](http://www.gipsa-lab.fr/~christian.bulfone/MIASHS-DCISS)

Adaptation : J.M. Adam



**Master MIASHS/DCISS**  
**Année 2020/2021**

# Qu'est-ce qu'un administrateur système ?



- Un administrateur système n'est qu'un utilisateur ayant des privilèges spéciaux et des devoirs (c'est le « super-utilisateur » ou superuser) :
  - « administrateur » depuis Windows NT et suivants
  - « root » sous Unix
- Son rôle :
  - maintenir le bon fonctionnement du parc de machines
  - configurer au mieux les machines
  - résoudre tout type d'incident
  - installer et mettre à jour le système et les nouveaux logiciels
  - administrer les disques (partitions, systèmes de fichiers)
  - gérer les utilisateurs (création, expiration, limitations)
  - guider et conseiller les utilisateurs
  - surveiller la sécurité du système
  - administrer le réseau local et l'accès au réseau public
  - organiser la sauvegarde des données
  - planifier l'évolution de son parc informatique

# Le compte root



- L'administration d'un système Unix, se fait en prenant l'identité de *root*
- Souvent l'invite de commandes est le symbole « # » et le répertoire de connexion `/root`
- Il est aussi possible de prendre temporairement, quand on est déjà connecté, l'identité de l'administrateur grâce à la commande `su`
- Remarques
  - Il n'est pas conseillé de toujours travailler connecté en tant que *root*. Il est préférable de disposer d'un compte ordinaire et d'exécuter la commande `su`
  - Par défaut, la commande `su` change l'identité mais pas l'environnement (sauf avec `su -`)
  - La commande `sudo` permet d'exécuter certaines commandes avec les privilèges de *root*

# Mises en garde



- L'utilisateur *root* a **tous les privilèges**  $\Rightarrow$  il est très facile de détériorer le système sous le compte *root*
- Quelques conseils utiles :
  - Bien vérifier les commandes tapées avant de presser sur la touche ENTREE. Pour l'effacement de fichiers, utiliser `rm -i` plutôt que `rm`
  - Utiliser les commandes en `sudo` pour les tâches courantes d'administration (mise à jour des applications ...)
  - réserver le compte *root* pour les opérations spécifiques (gestion des disques, des utilisateurs ...)

# Les fichiers d'administration



- C'est le moyen primitif imaginé par les créateurs du système pour administrer un Unix
  - Consiste à éditer, avec l'éditeur `vi`, le fichier d'administration concerné
- Tous les fichiers de configuration et d'administration du système Linux sont de type texte et sont stockés dans le répertoire `/etc`
- Cette méthode d'administration nécessite de bien connaître la structure du fichier et les liens éventuels avec d'autres fichiers, sous peine d'introduire des incohérences graves.

# Les commandes d'administration



- Appelées le plus souvent commandes systèmes, elles sont stockées dans le répertoire `/sbin` et ne sont accessibles qu'à l'utilisateur *root*
- Le plus souvent, ces commandes modifient un ou plusieurs fichiers d'administration
- La connaissance approfondie de ces commandes permet aux administrateurs expérimentés d'automatiser leur utilisation dans des scripts

# Les scripts



- Les scripts sont des fichiers textes permettant l'enchaînement de commandes
  - Pour s'exécuter, un script doit posséder le droit d'exécution et localiser son interpréteur
- Les scripts permettent d'automatiser des tâches répétitives d'administration
- L'écriture d'un script simplifie le travail de l'administrateur en même temps qu'il sécurise le fonctionnement du système en permettant la programmation de nombreux contrôles, préalables à l'exécution d'une commande
- Langages de script couramment utilisés par les administrateurs : les *shells* Unix, Perl, Python, ...

# Les outils intégrés



- Beaucoup de distributions fournissent maintenant des outils intégrés qui permettent de réaliser les principales tâches d'administration avec une interface Homme-Machine (IHM)
- Ces outils évitent de mémoriser la syntaxe des commandes et de connaître la structure des fichiers d'administration
- Les outils peuvent différer sensiblement dans la forme selon les distributions Unix mais ils fournissent à peu près les mêmes fonctionnalités
- Certains outils fonctionnent soit en mode texte soit en mode graphique, et d'autres dans les deux modes



- Sous Linux
  - documentation gratuites (en anglais) disponibles sur <http://tldp.org>
    - sag : *The Linux System Administrator's Guide*
    - nag : *The Linux Network Administrator's Guide*
    - lasg : *Linux Administrator's Security Guide*
- Le *System Administration Cookbook*
  - un peu vieux mais traite de tous les systèmes Unix
  - plus de 400 pages, une vraie mine d'informations

# Gestion des utilisateurs et groupes



# Notions de logins et de groupes



- Pour ouvrir une session, chaque utilisateur doit posséder un **identifiant unique** - un *login name* ou *login* - protégé par un mot de passe (*password*)
- Un utilisateur appartient au moins à un groupe, dit **groupe primaire**, défini par l'administrateur à la création du compte utilisateur
- Il peut appartenir à des groupes supplémentaires auxquels il peut accéder en cours de session
- Les informations qui caractérisent tous les utilisateurs (y compris l'administrateur) sont regroupées dans le fichier `/etc/passwd` et celles des groupes dans le fichier `/etc/group`
- Les comptes d'utilisateurs et les groupes sont également utilisés pour identifier les fichiers appartenant à une application

# Les utilisateurs sous Unix



- Liste des utilisateurs dans le fichier `/etc/passwd`
- Utilisateur identifié par :
  - nom de login
  - mot de passe
  - UID : numéro unique identifiant l'utilisateur
  - GID : numéro unique du groupe auquel appartient l'utilisateur
  - nom complet
  - répertoire de travail (*home directory*)
  - shell à utiliser

# Les utilisateurs sous Unix



- Un super utilisateur
  - nom de login : *root*, UID : 0, GID : 0
  - peut outrepasser tous les droits sur tous les fichiers
  - peut effectuer tous les appels systèmes
- Utilisateur sans droit : *nobody.nobody*
- Utilisateurs spécifiques à certains services
  - *ftp* : service de ftp anonyme
  - *lp* : service d'impression
  - comptes étoilés (mot de passe chiffré = \*) donc impossible de se logger avec ces comptes

# Création des comptes utilisateurs



- Plusieurs outils sont disponibles pour créer les comptes utilisateurs
  - En mode console la commande `useradd` est généralement utilisée
- L'UID est une valeur comprise entre 0 et la valeur définie par la constante `UID_MAX` du fichier `/etc/login.defs`
  - Les valeurs  $< 100$  sont généralement réservées pour des utilisateurs associés à des services standard du système Linux
- La constante `UID_MIN` du fichier `/etc/login.defs` définit la valeur minimale des UID des utilisateurs
- L'attribution d'un UID est de la responsabilité de l'administrateur et rien ne l'oblige à les affecter séquentiellement.
  - Il peut définir sa propre stratégie
- Avec un parc de machines Linux sans une administration centralisée (NIS notamment voir plus loin), il est conseillé d'attribuer le même UID à un utilisateur qui possède un compte sur plusieurs machines du réseau

# Les groupes sous Unix



- Liste des groupes dans le fichier `/etc/group`
- Groupe identifié par :
  - nom de groupe
  - mot de passe
  - GID : numéro unique du groupe
  - liste des utilisateurs (logins séparés par des virgules)
- Possibilité d'appartenir à plusieurs groupes

```
bulfone@crystal:~$ id bulfone
uid=10145(bulfone) gid=10105(icp)
groups=10105(icp),10011(synthese),10037(www),10038(staff)
```

- Sélection du groupe principal avec la commande `newgrp`

# Authentification



- Processus qui vérifie le login et le mot de passe
  - clé de perturbation + mot de passe  $\Rightarrow$  chiffrement DES\*  
25 fois  $\Rightarrow$  mot de passe chiffré
  - comparaison avec le mot de passe chiffré stocké dans  
`/etc/passwd`
- Autres chiffrements possibles : MD5, ...
- *Shadow passwords* :
  - déporte les mots de passes chiffrés dans un fichier  
seulement accessible par *root* : `/etc/shadow`
  - empêche la récupération des mot de passes chiffrés  
pour tenter de les cracker

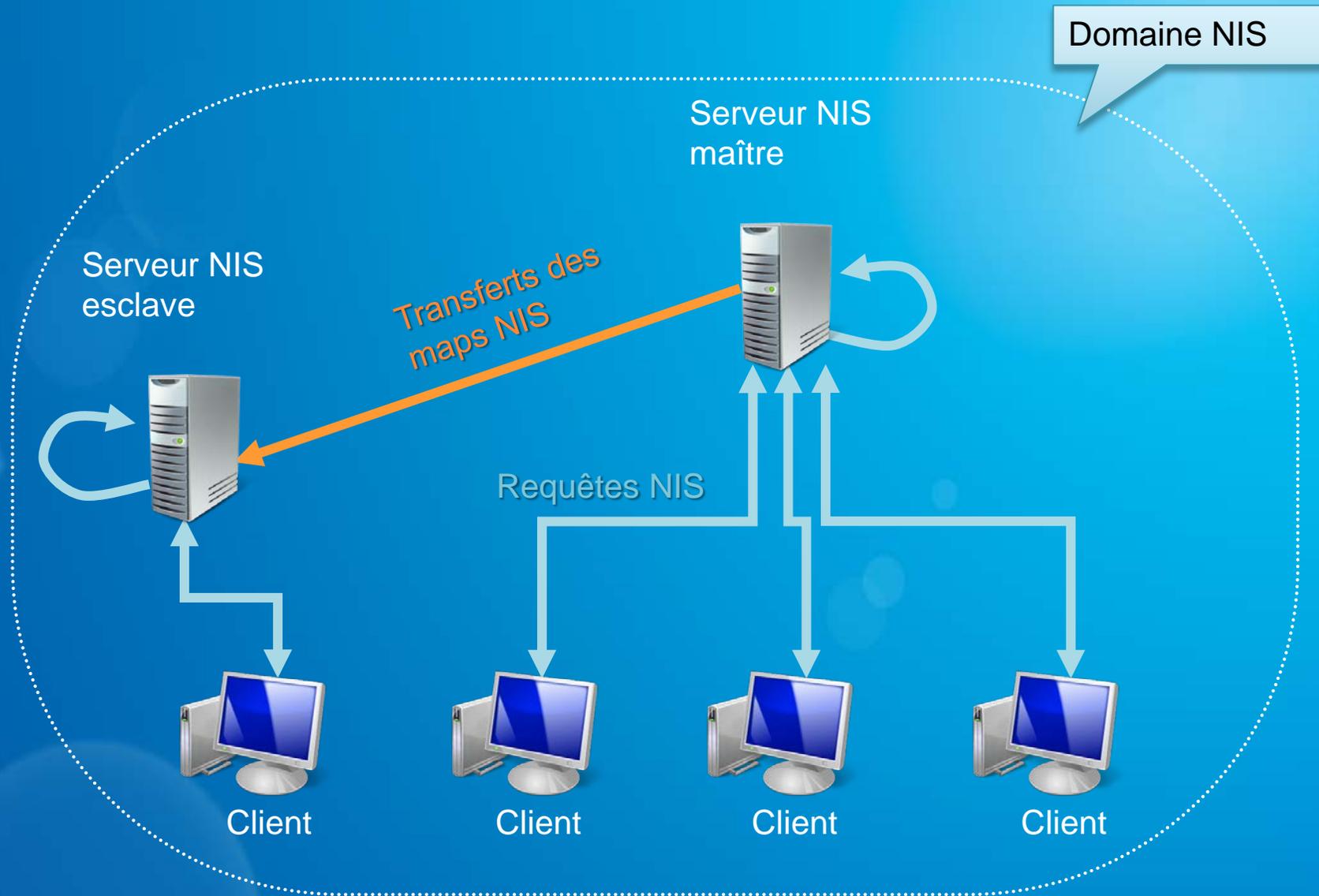
(\*) DES = Data Encryption Standard

# Le Network Information System (NIS)



- NIS ou YP (*Yellow Pages*) développé par Sun
- Gestion centralisée de fichiers communs à plusieurs machines (→ système de bases de données réparties) :
  - `/etc/passwd`, `/etc/group`, `/etc/shadow`, `/etc/hosts`, `/etc/services`, `/etc/protocols`, ...
  - les bases de données gérées par NIS s'appellent des « NIS maps »
- Un serveur maître (**master**)
  - programme `ybserv`
- Eventuellement des serveurs esclaves (**slaves**) en cas de panne
- Des clients interrogeant les serveurs
  - programme `ybind`
  - les serveurs sont aussi clients

# Architecture de NIS



# Configuration des YP



- Configuration dans le fichier `/etc/yp.conf`
- Côté serveur :

```
ypserver localhost
domain nis.mondomain broadcast
```
- Côté client :

```
ypserver miashs-dc.u-ga.fr
domain nis.mondomain broadcast
```
- Configuration des services utilisant le NIS dans `/etc/nsswitch.conf`
- Compilation des YP :
  - à chaque fois qu'un fichier `/etc/...` est modifié sur le serveur
  - root doit faire : `cd /var/yp ; make`

# Commandes principales des YP



- `ypinit` : création de la base
- `ypcat` : visualisation des fichiers de la base
- `ypwhich` : nom du serveur NIS
- `domainname` : configuration du nom de domaine NIS
- `ypserv` : démon serveur de NIS
- `ypbind` : démon client de NIS
- `yppasswd` : changement de mot de passe NIS
- `yppasswdd` : démon pour le changement de mot de passe NIS

# Résolution des noms de machines



- Liste des noms/adresses IP dans le fichier `/etc/hosts`
- Résolution par les pages jaunes (YP ou NIS)
  - diffusion du fichier `/etc/hosts` du serveur NIS
- Résolution par le DNS
  - configuration dans le fichier `/etc/resolv.conf`

```
domain u-ga.fr
search u-ga.fr
nameserver 195.221.40.253
```
  - possibilité de définir plusieurs serveurs
- Configuration de l'ordre des résolutions (fichier, NIS ou DNS) dans `/etc/nsswitch.conf`

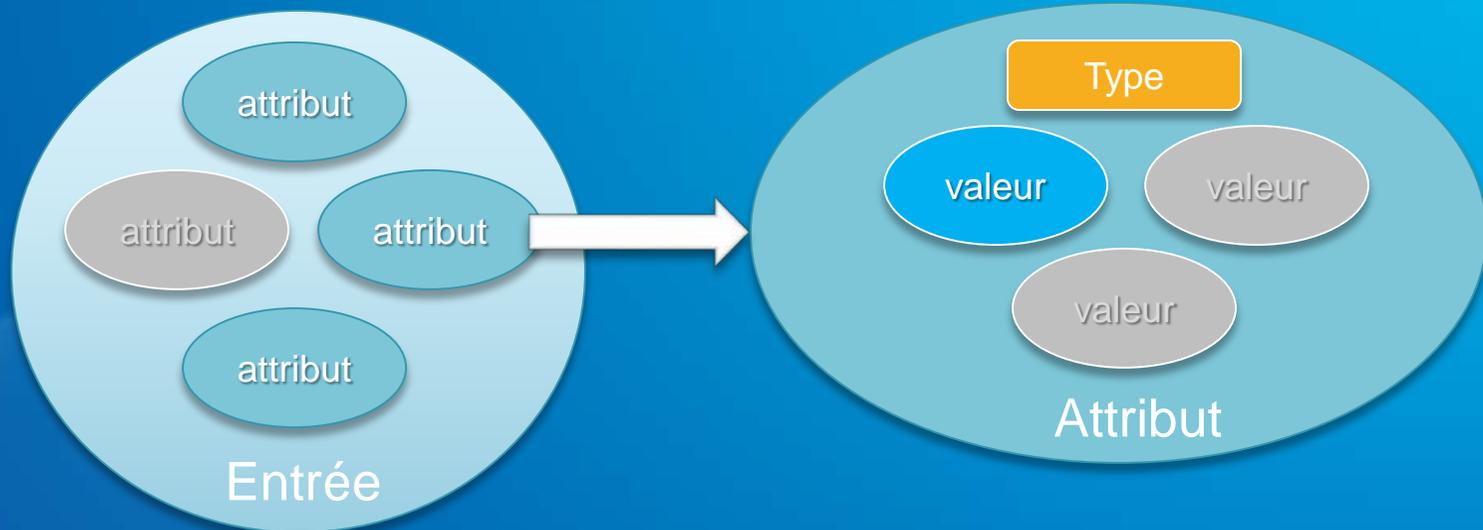


## LDAP (*Lightweight Directory Access Protocol*)

- Issu de DAP, protocole d'accès à l'annuaire X500
- Annuaire à part entière, spécifié par la version 3 du protocole (version actuelle)
- Nombreuses implémentations
  - Active Directory de Microsoft
  - OpenLDAP
  - ...
- Utilisé aussi bien par les systèmes d'exploitation que par les applications
- **Est devenu le standard des annuaires électroniques dans les systèmes d'information des entreprises**



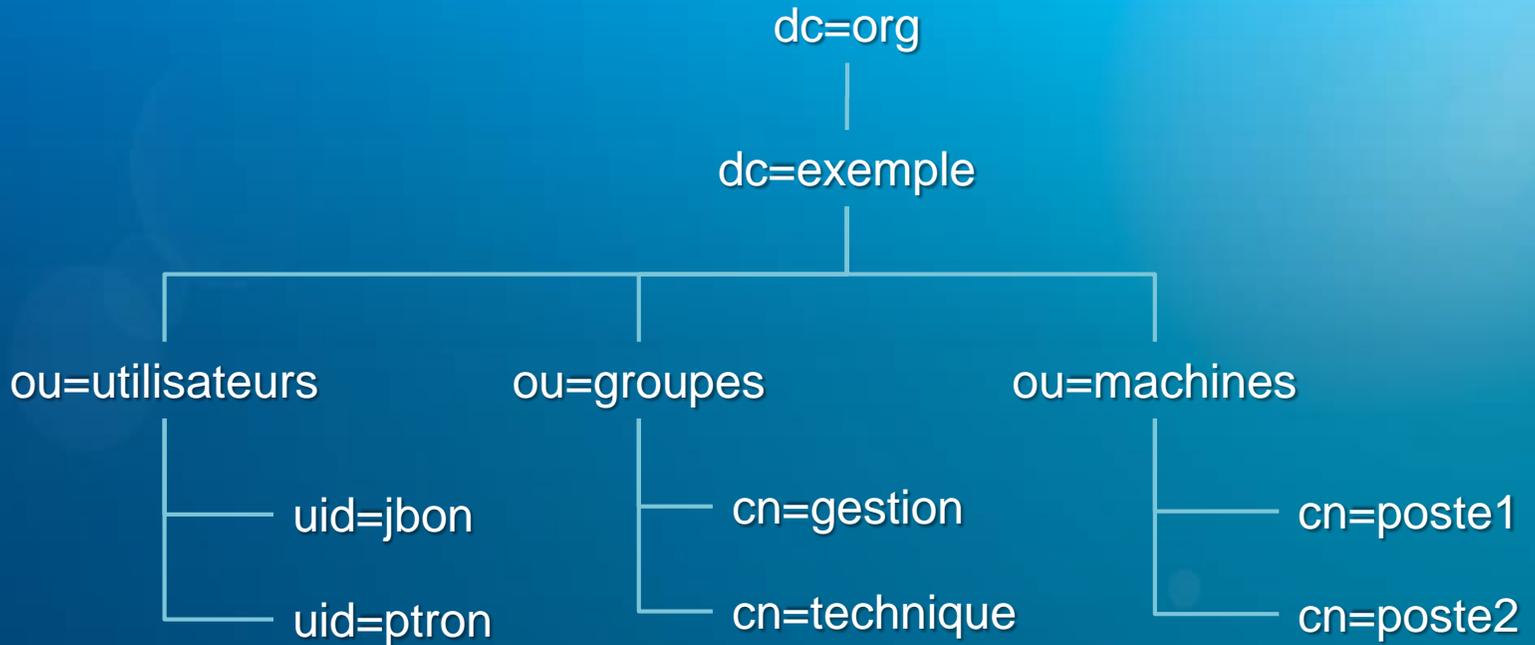
- Un annuaire est un **arbre d'entrées**
  - l'arbre reflète le modèle organisationnel, politique ou géographique de la structure représentée
- Une entrée est constituée d'un ensemble d'**attributs**
  - Un attribut possède un **nom**, un **type** et **une ou plusieurs valeurs**





- Les attributs sont définis dans des **schémas**
  - Caractère multivalué = différence majeure avec les SGBD
  - Un attribut qui n'a pas de valeur est absent de l'entrée
- Chaque entrée a un identifiant unique, le *Distinguished Name* (DN)

# LDAP

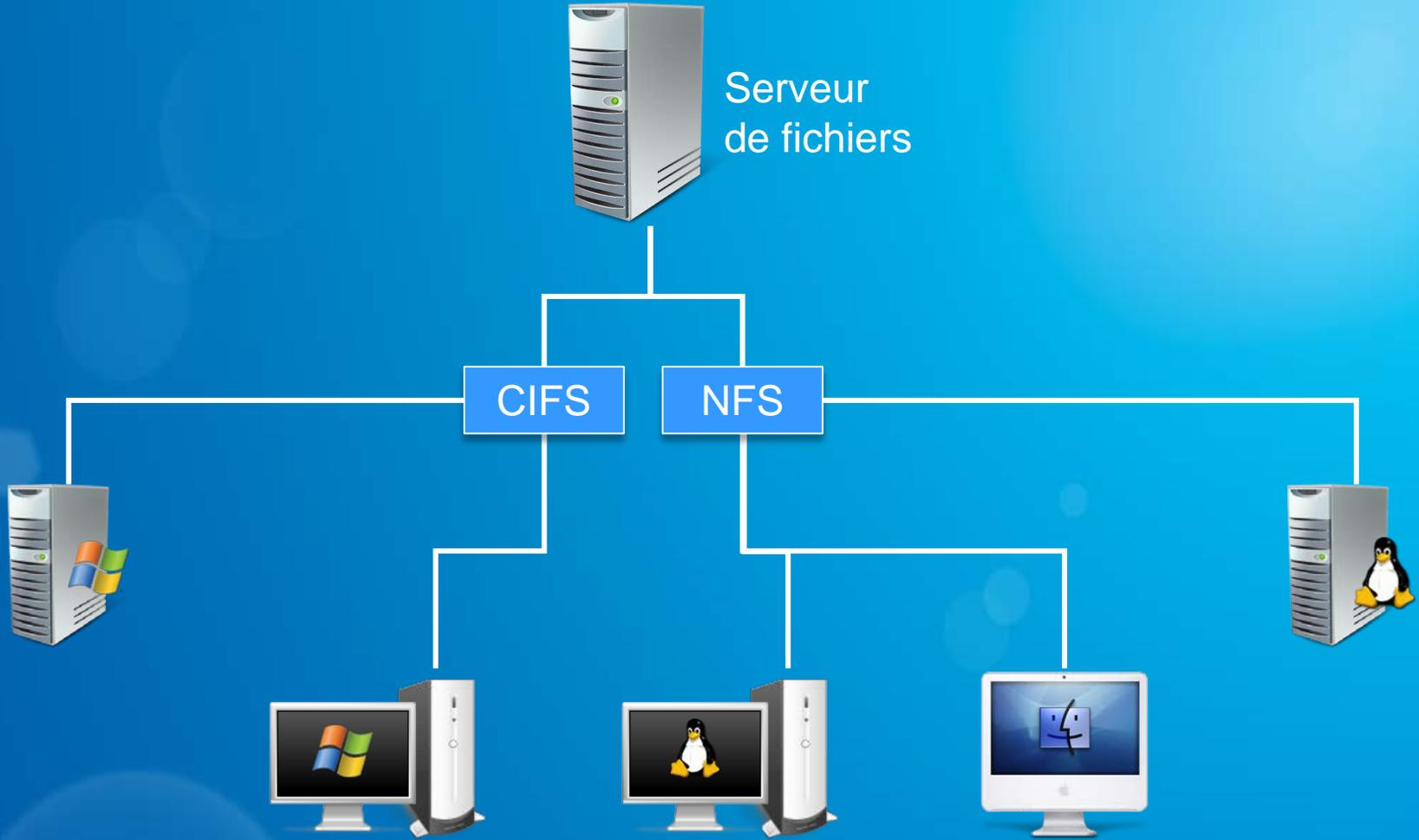


dn: uid=ptron,ou=utilisateurs,dc=exemple,dc=org  
cn: Paul Tron  
givenName: Paul  
sn: Tron  
uid: ptron  
telephoneNumber: +33 1 23 45 67 89  
mail: paul.tron@exemple.org  
objectClass: inetOrgPerson  
objectClass: organizationalPerson  
objectClass: person  
objectClass: top

# Partage de fichiers



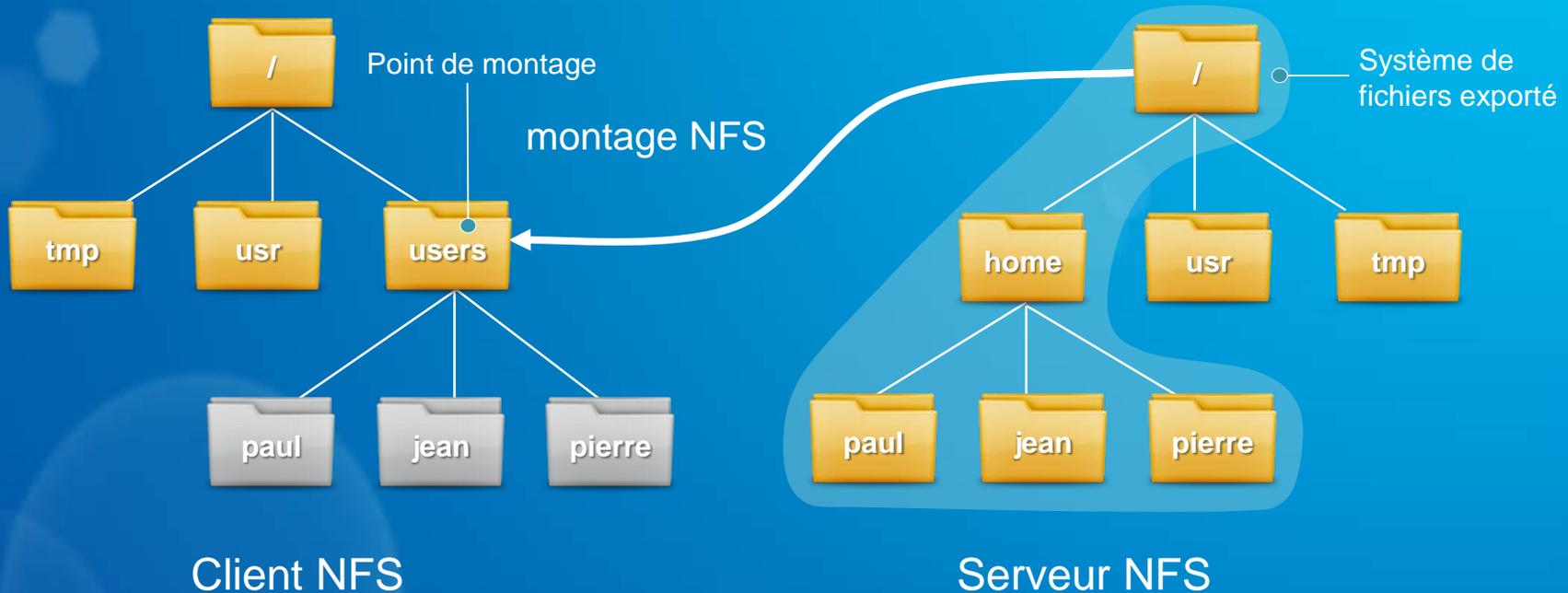
# Protocoles de partages de fichiers



# NFS



- *Network File System* développé par Sun
- But : partager un espace utilisateur entre toutes les stations
- NFS permet le montage d'arborescences appartenant à d'autres systèmes connectés au réseau
  - le serveur NFS **exporte** (ou publie) un système de fichiers auquel un ou plusieurs clients peuvent accéder
  - le client NFS **monte** un répertoire NFS exporté par le client



# RPC (*Remote Procedure Call*)



- NFS s'appuie sur RPC (*Remote Procedure Call*)
  - Également inventé par Sun
  - Décrit une méthode de type client/serveur
  - Utilise le format XDR (*eXternal Data Representation*)
  - Chaque service est représenté par un numéro
  - Le fichier `/etc/rpc` établit la correspondance entre numéro et service

# RPC (*Remote Procedure Call*)



- RPC fonctionne sur TCP/IP
- Les programmes `rpcbind/portmap` font le lien entre le numéro et le port
- Le client se connecte au serveur RPC (port 111 tcp/udp) et envoie le numéro de service
- Le serveur RPC renvoie le port du service voulu
- La commande `rpcinfo` permet de dialoguer avec le serveur RPC (visualisation des tables...)

# Configuration du serveur



- Description des partages dans le fichier `/etc/exports`

- Format

`partage hôte1(options) hôte2(options) ....`

- Options

- `rw` : droits de lecture et d'écriture sur le partage
- `ro` : droit uniquement de lecture sur le partage
- `async` : permet au serveur NFS de répondre à des requêtes avant que les modifications précédentes aient été sauvegardées
- `sync` : force la synchronisation avec le serveur (valeur par défaut)
- `root_squash/no_root_squash` : transforme ou non les requêtes d'UID/GID 0 en UID/GID anonyme (`root_squash` par défaut)
- `all_squash/no_all_squash` : transforme ou non tous les UID/GID en utilisateur anonyme (`no_all_squash` par défaut)
- `anonuid/anongid` : définit explicitement l'UID/GID des utilisateurs anonymes
- Autres options disponibles voir `man exports`

# Commandes d'administration



- `/etc/init.d/nfs {start|stop|restart}`
  - Lancer/Stopper/Relancer le serveur & RPC
- `exportfs`
  - Permet de manipuler les partages
  - `exportfs -a`  
exporte tout les partages de `/etc/exports`
  - `exportfs -u`  
désactive un partage (`-ua` désactive tous les partages de `/etc/exports`)
  - `exportfs -o rw,async host1:/data`  
exporte le répertoire `/data` du serveur vers `host1`
- `showmount`
  - Visualise les partages actuellement exportés
- `nfsstat`
  - Affiche des statistiques concernant les RPC et NFS

# Options de la commande mount



- `fg|bg`
  - Si le montage échoue, les tentatives suivantes se font en avant ou arrière plan (`fg` par défaut)
- `rw|ro`
  - Le partage est monté en lecture/écriture ou lecture seule (`rw` par défaut)
- `intr|nointr`
  - Autorise ou non l'interruption grâce à CTRL+C (`intr` par défaut)
- `suid|nosuid`
  - Autorise ou non l'exécution de programmes SUID (`suid` par défaut)
- `hard|soft`
  - Recommence l'opération jusqu'à sa réussite ou se termine en cas d'échec (`hard` par défaut)

# Automounter



- Système de « montage » automatique de partition ou de systèmes de fichier (FS) au moment de l'accès côté client
- Les FS sont automatiquement démontés après une période d'inactivité
  - Evite de consommer des ressources inutilement
  - Permet aux clients de s'affranchir des connexions persistantes et des phénomènes de blocage

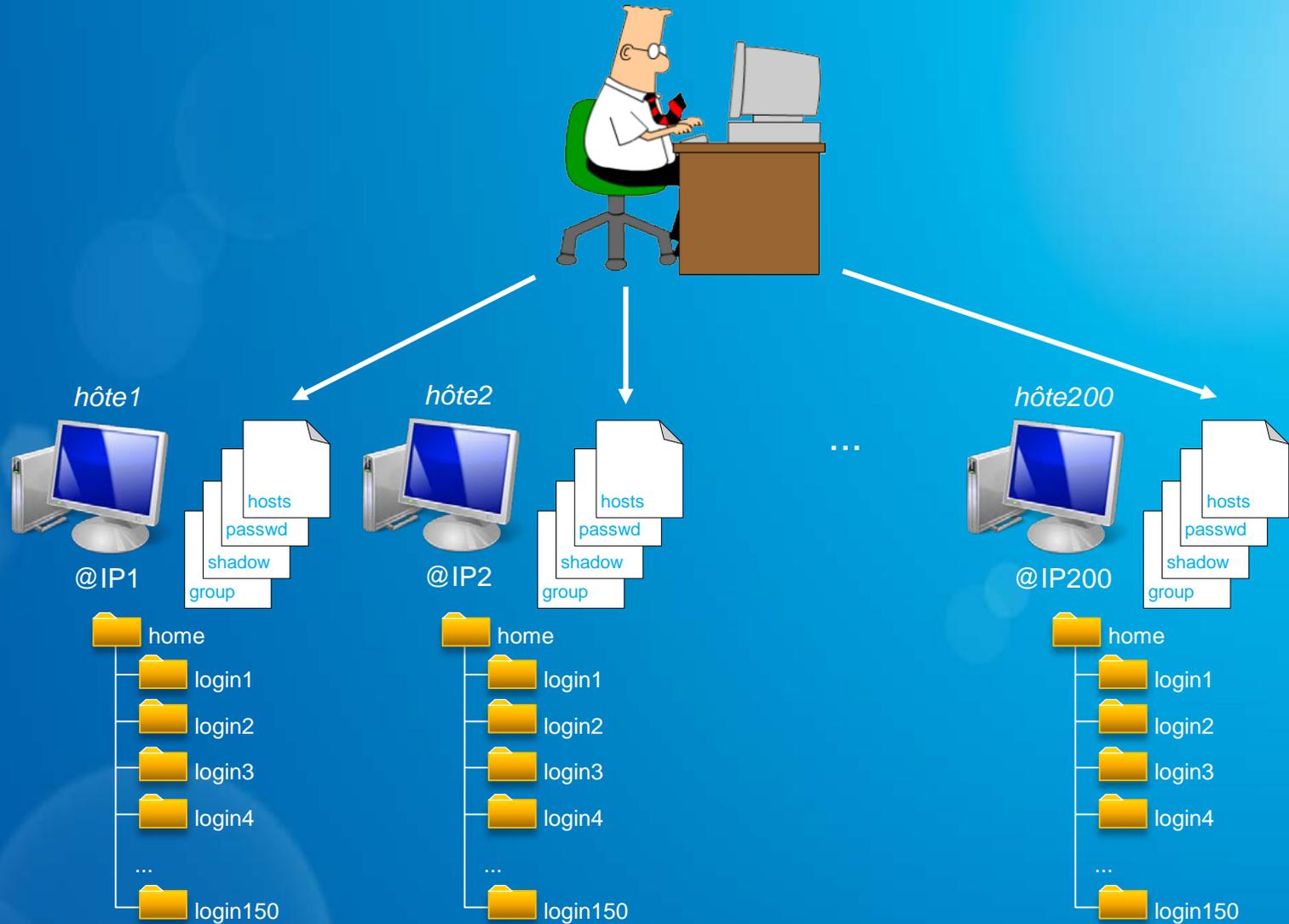
# Pourquoi utiliser NIS et NFS ?



- Les données du problème
  - 1 réseau local
  - 200 postes de travail
  - 150 utilisateurs
  - 1 seul administrateur : VOUS !



# Solution sans NIS ni NFS



# Solution avec NIS + NFS

