



Plan

1. Structure générale des Services d'Application
2. Exemples de Services d'Application
 1. le Service d'impression LPD
 2. le Service de partage de disques NFS
 3. le Service de partage d'informations d'administration NIS
 4. Le service de nommage DHCP
3. Exemple d'application réseau
 1. MSN
 2. XMPP (Extensible Messaging and Presence Protocol)

Les Services d'Application

Un Service d'Application est conçu pour

- rendre accessibles des informations mémorisées sur un système distant
 - courrier électronique
 - transfert de fichiers
 - web
 - partage d'informations d'administration
- rendre accessible une ressource appartenant à un système distant
 - ressource de calcul
 - travail à distance
 - ressource d'impression
 - partage d'imprimantes
 - ressource en disque
 - partage de fichiers

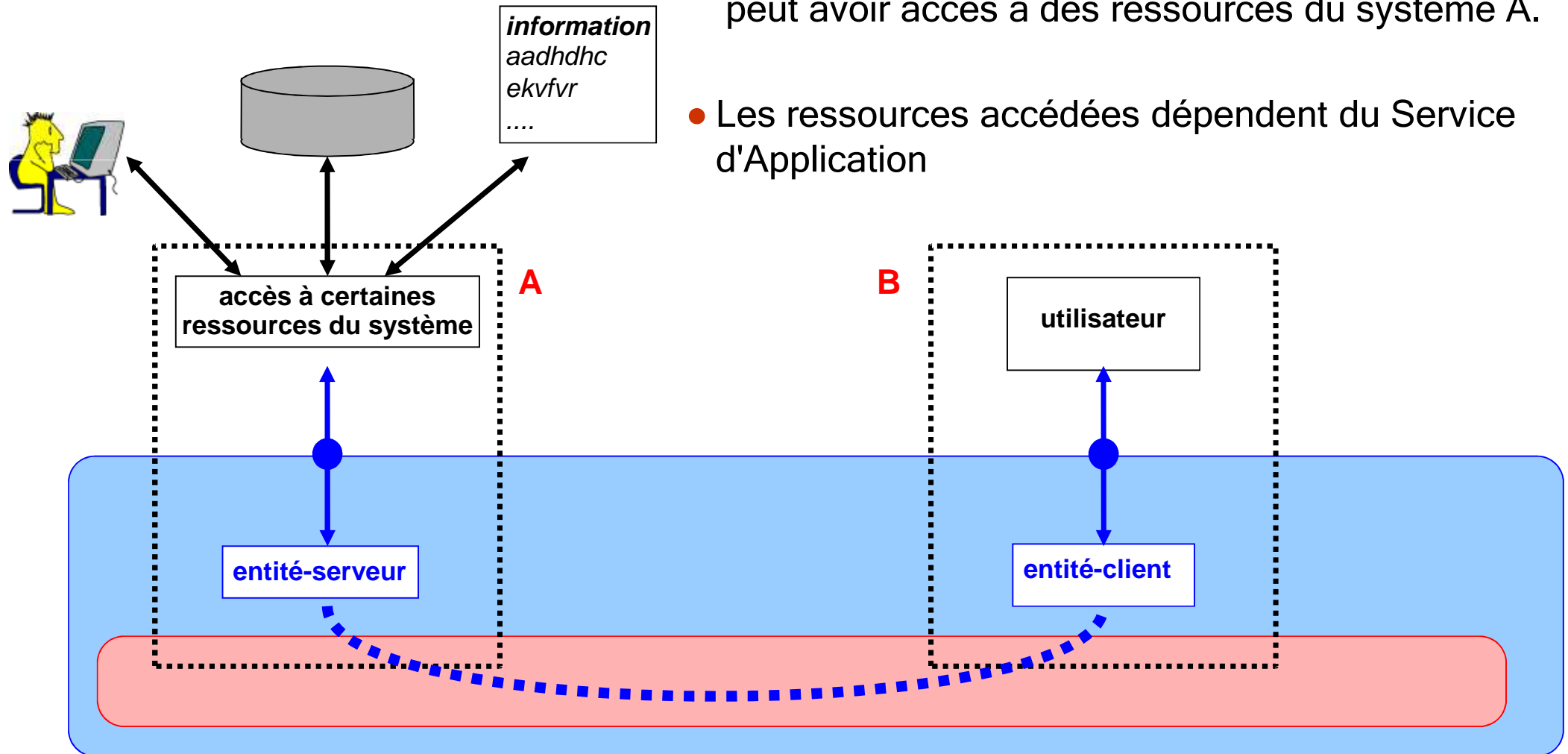
une couche Application par Service

Organisation d'un fournisseur de Service d'Application

- sur le système qui possède une information ou une ressource : une **entité-serveur**
- sur le système qui a besoin d'une information ou d'une ressource : une **entité-client**
- la transmission des PDU entre client(s) et serveurs(s) est effectuée par l'utilisation d'un Service inférieur

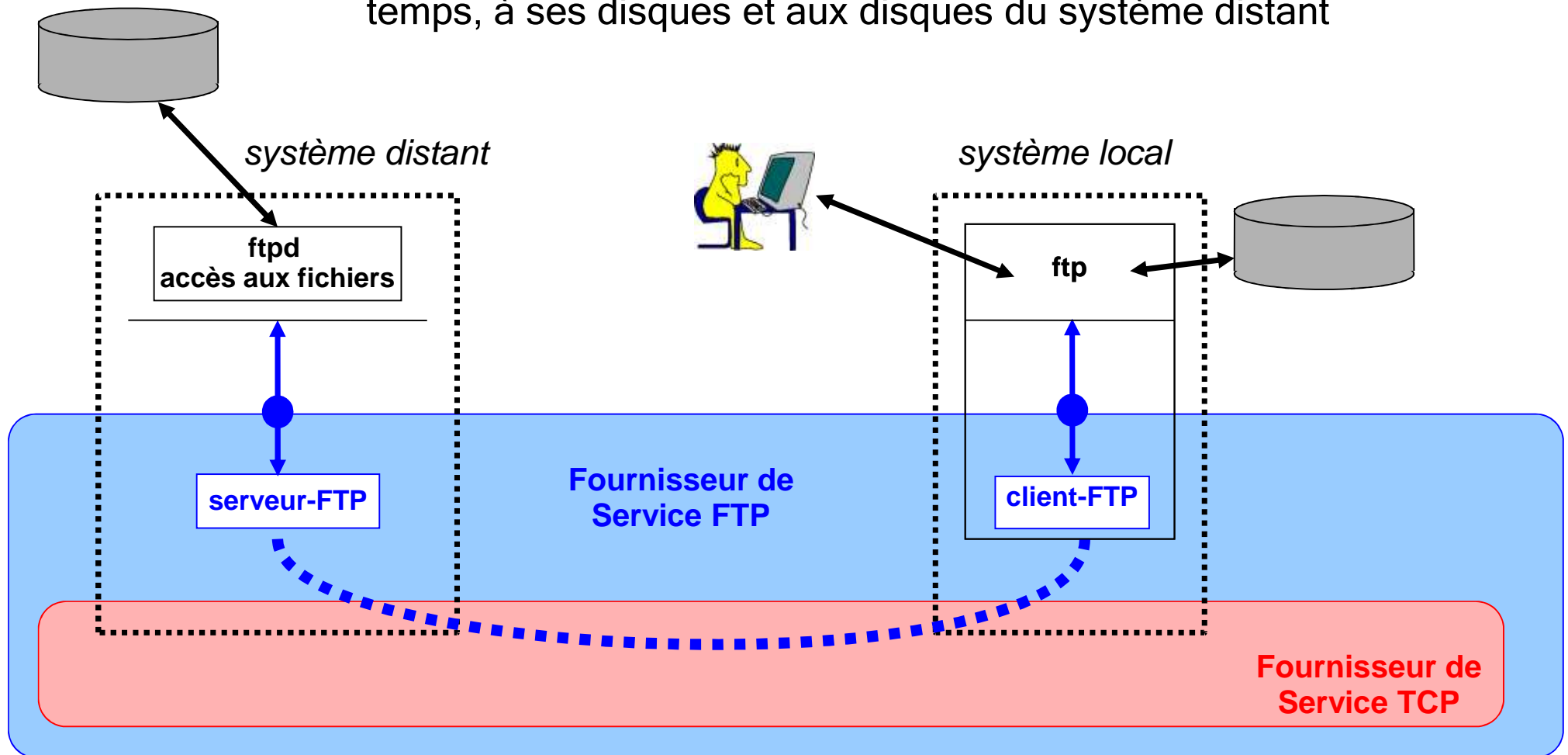
Principes

- L'entité-utilisateur installée dans le système B peut avoir accès à des ressources du système A.
- Les ressources accédées dépendent du Service d'Application



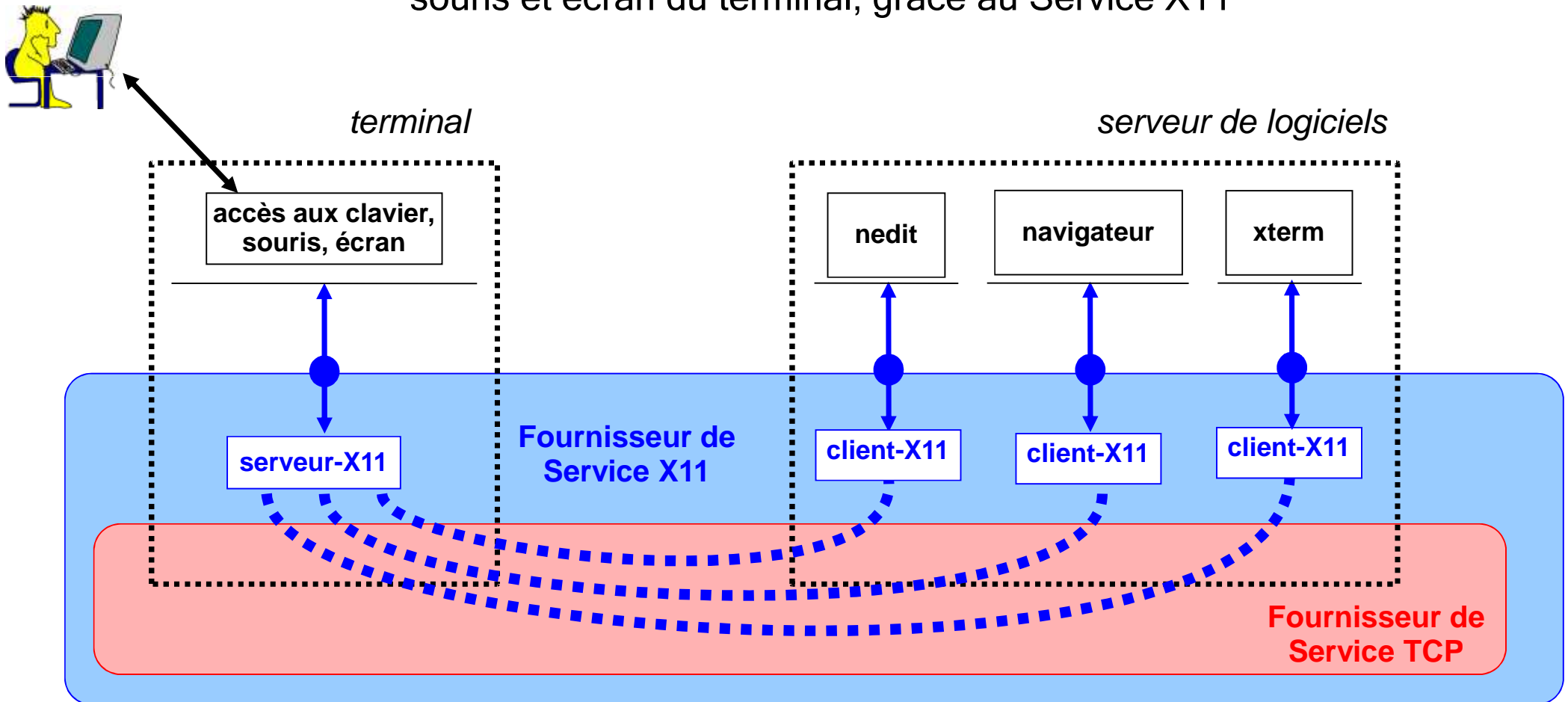
Service de Transfert de fichiers (FTP) : 1 serveur - 1 client

- le processus qui s'exécute sur le système local a accès, en même temps, à ses disques et aux disques du système distant



Service de Terminal distant (X11) : 1 serveur - N clients

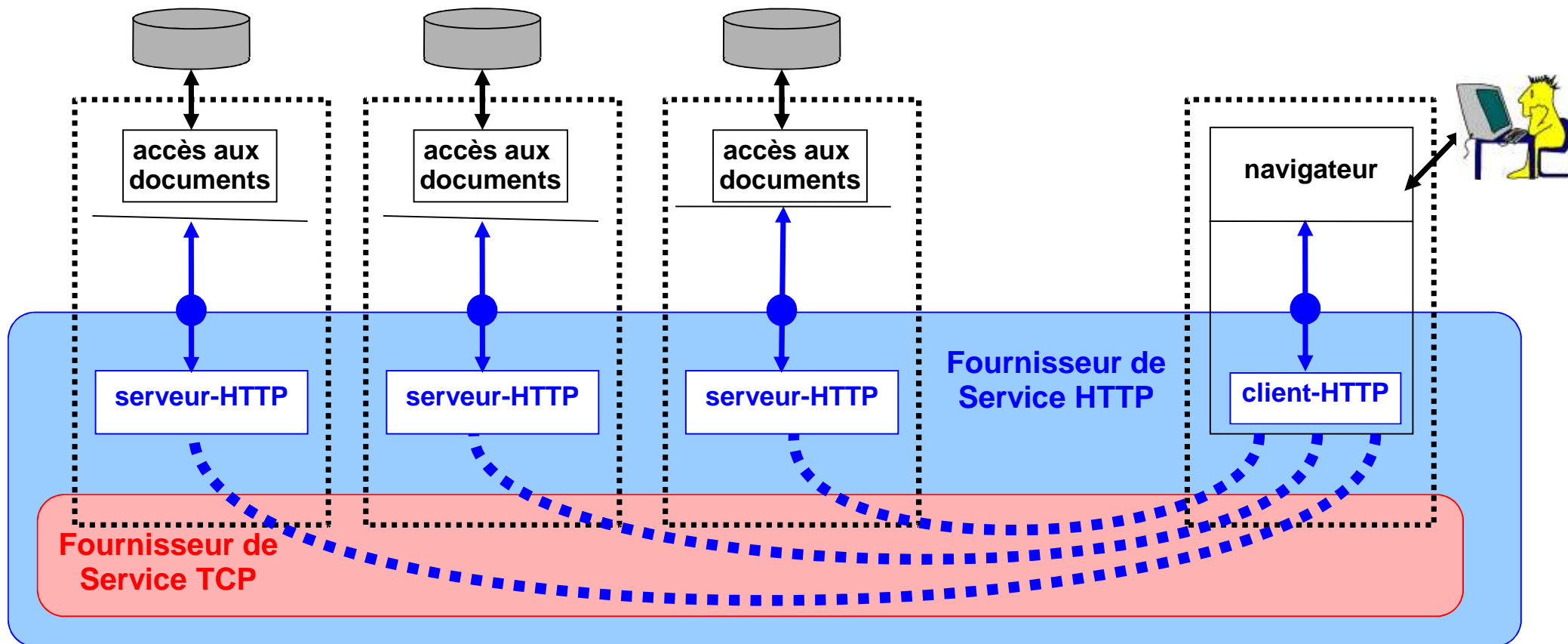
- l'éditeur de texte, le navigateur, et les autres processus qui s'exécutent sur le serveur de logiciels peuvent utiliser les claviers, souris et écran du terminal, grâce au Service X11



Service HTTP (Web) :

N serveurs - 1 client ou 1 serveur - N clients

- 1 navigateur - client HTTP - demande des documents à un ou plusieurs serveurs
- et chaque serveur-HTTP peut servir plusieurs navigateurs





Implémentation des entités d'Application (1)

Les entités client

- processus créés lorsque le besoin apparaît
 - par une personne
 - au démarrage du système
- l'entité-client doit savoir à quelle entité-serveur elle doit/peut s'adresser
 - au lancement du processus : fichier de configuration, argument
 - par saisie pendant l'exécution



Implémentation des entités d'Application (2)

Les entités serveur

- **fonctionnement permanent (démons)**
 - processus créés au démarrage du système
 - Unix : scripts dans `/etc/rc.d` ou `/etc/init.d`
 - Microsoft xp pro: Panneau de configuration → Outils d'administration → Services
- **processus lancés lorsqu'une demande arrive**
 - par un processus "écouteur"
 - Unix : `inetd`, configuré par `/etc/inetd.conf`
 - Microsoft xp pro: Panneau de configuration → Outils d'administration → Services

Méthodes de prise en charge d'une demande de Service

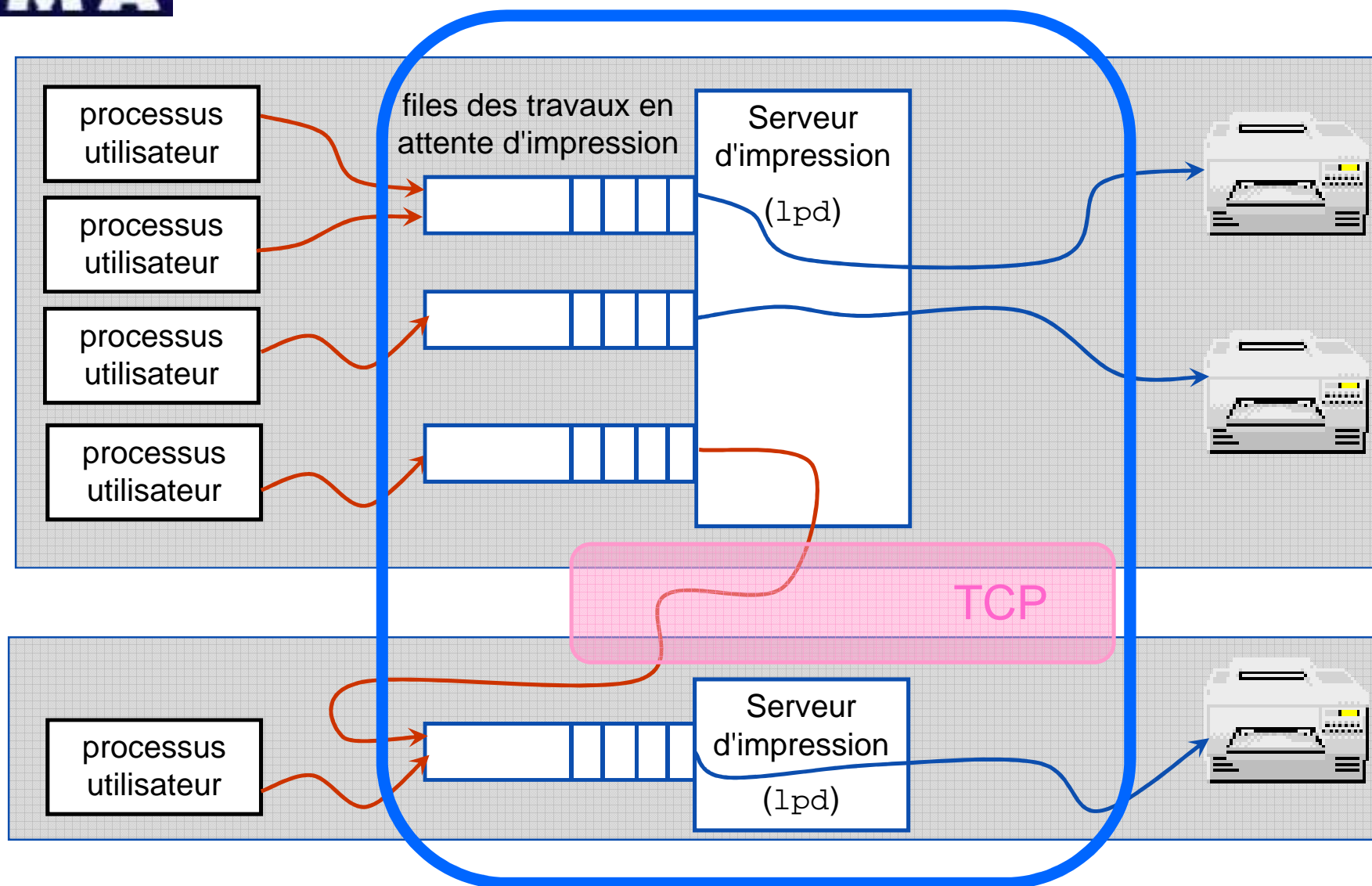
- **L'entité serveur peut traiter elle-même la demande**
 - Mise en attente des autres demandes jusqu'à la fin du traitement
- **L'entité serveur peut déléguer à un processus fils (ou un thread) le traitement de la demande**
 - Contrôle du nombre de fils (ou de thread) -> nombre maximum donné, pré-crédation ...
 - Risque de surcharge de la mémoire



Exemple : /etc/inetd.conf

<i>#Service #Name</i>	<i>Socket Type</i>	<i>Protocol Name</i>	<i>Wait NoWait</i>	<i>User Name</i>	<i>Server Path</i>	<i>ServerArgs Args</i>
<i>ftp</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>root</i>	<i>/usr/sbin/ftpd</i>	<i>ftpd</i>
<i>telnet</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>root</i>	<i>/usr/sbin/telnetd</i>	<i>telnetd</i>
<i>shell</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>root</i>	<i>/usr/sbin/rshd</i>	<i>rshd</i>
<i>exec</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>root</i>	<i>/usr/sbin/rexecd</i>	<i>rexecd</i>
<i>ntalk</i>	<i>dgram</i>	<i>udp</i>	<i>wait</i>	<i>root</i>	<i>/usr/sbin/ntalkd</i>	<i>ntalkd</i>
<i>time</i>	<i>dgram</i>	<i>udp</i>	<i>wait</i>	<i>root</i>	<i>internal</i>	<i>time</i>
<i>finger</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>nobody</i>	<i>/usr/sbin/tcpd</i>	<i>/usr/sbin/in.fingerd</i>
<i>imap2</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>root</i>	<i>/usr/sbin/tcpd</i>	<i>/usr/sbin/imapd</i>
<i>imap3</i>	<i>stream</i>	<i>tcp</i>	<i>nowait</i>	<i>root</i>	<i>/usr/sbin/tcpd</i>	<i>/usr/sbin/imapd</i>

LPD (1) : partage d'imprimante





LPD (2) : configuration des files d'imprimante d'un serveur

imprimante locale

<code>prtlocale:\</code>	
<code>:lp=/dev/lpt1:\</code>	nom de périphérique
<code>:sd=/var/spool/lpd/prtlocale:\</code>	répertoire file d'attente
<code>:if=/usr/sbin/hplaser4ps:\</code>	filtre
<code>:lf=/var/spool/lpd/prtlocale/err:</code>	journal des erreurs

imprimante distante

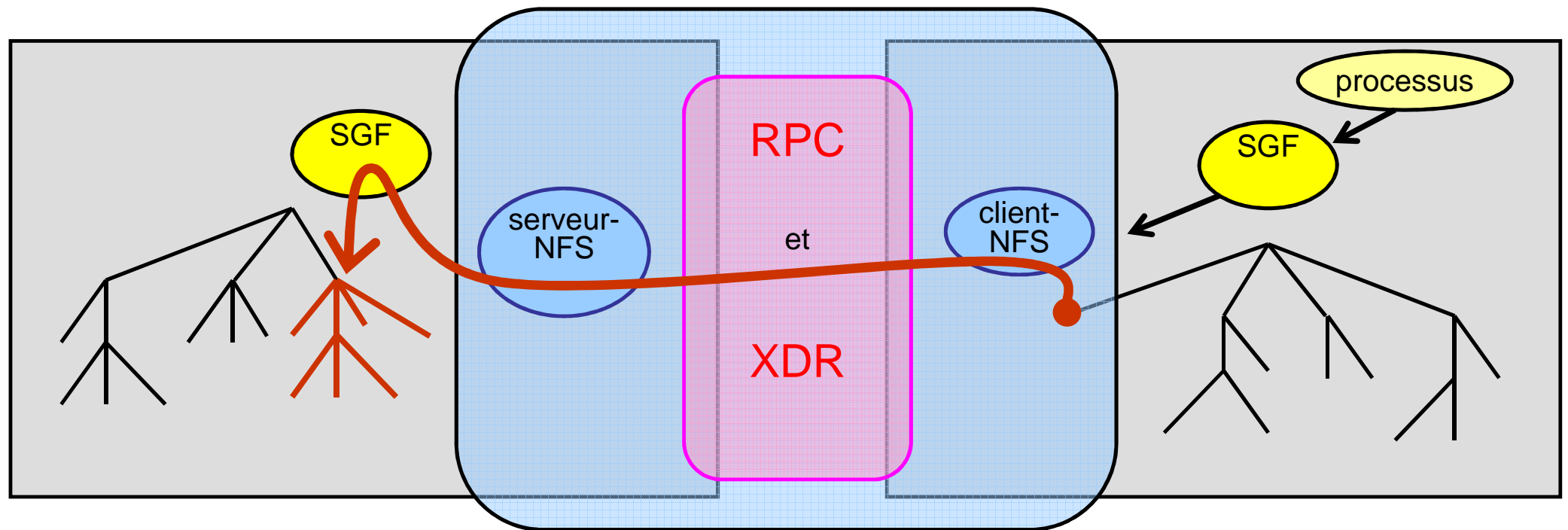
<code>prtdistante:\</code>	
<code>:lp=:\</code>	
<code>:rm=192.168.51.30:\</code>	nom du système "propriétaire"
<code>:rp=prtdocu:\</code>	nom de l'imprimante sur ce système
<code>:sd=/var/spool/lpd/prtdistante:\</code>	répertoire file d'attente
<code>:lf=/var/spool/lpd/prtdistante/err:</code>	journal des erreurs

LPD (3) : les principales commandes

lpr lp	envoie un ou plusieurs fichiers dans une file d'attente d'un serveur d'impression	
lpq	affiche la liste des travaux en attente d'impression	
lprm cancel	supprime un ou plusieurs travaux en attente d'impression	
lpc	status	affiche l'état d'une file d'attente
	topq	met un travail en tête de la file d'attente
	stop	désactive l'impression
	start	ré-active l'impression
	disable	interdit la mise en file d'attente
	enable	autorise la mise en file d'attente
	down	disable + stop
	up	enable + start
	clean	vide une file d'attente

Syntaxe et arguments dans les manuels d'Unix

NFS (1) : partage de répertoires



serveur-NFS	facilités	client-NFS
mountd	montage	mount
nfsd	opérations sur les fichiers et répertoires	nfsiod

NFS (2) : caractéristiques

identifications

- du serveur, du client :
 - par l'adresse-IP des systèmes
- des utilisateurs : par UID-GID

serveur-NFS sans état

caches des clients-NFS

- attributs des fichiers et répertoires
- lectures anticipées
- écritures retardées

autorisations de montage

- liste des répertoires exportés
- liste des clients autorisés

autorisations d'accès aux fichiers

- selon UID-GID
- UID=0 (root) devient UID=65534 (nobody) sauf configuration

Service support : RPC

(Remote Procedure Call)

- liste de fonctions :

getattr	read	remove
setattr	write	rename
lookup	create	...

- paramètres :

codés par XDR (*eXternal Data Representation*)

- service support de RPC :

- UDP ou TCP

NFS (3) : configuration

Serveur

1. construire la liste des répertoires exportables `/etc/exports`
options
clients autorisés
type de montage (ro, rw)
contrôle des UID-GID
2. vérifier que le Service RPC est lancé `rpcinfo -u localhost`
3. lancer les serveurs `/etc/rc.d/init.d/nfs start`
 - de montage
 - d'opérations sur les fichiers
 - de verrouillage des fichiers
4. vérifier l'exportation

Client

- montage
 - à partir d'un fichier de configuration
 - manuellement
 - options de montage habituelles ou spécifiques à NFS (voir manuel `mount(8)`)
- opérations sur les fichiers
 - par les fonctions habituelles

Informations d'administration

informations nécessaires pour le fonctionnement des systèmes

- identification des utilisateurs comptes, mots de passe, adresses-mail, ...
- identifications des systèmes noms, adresses-MAC, adresses-IP,...
- identification des entités du réseau appellations des protocoles, des applications ...

domaine d'administration

ensemble des systèmes qui ont besoin des mêmes informations d'administration

- des bases d'informations d'administration
 - base des comptes base des adresses-mail
 - base des systèmes ...
- un ou plusieurs serveurs d'informations d'administration
- des utilisateurs d'informations d'administration

partage d'informations d'administration

- primitives d'accès aux informations d'administration
- réglementation des accès

les solutions les plus connues		
NIS	Network Information System	SUN
DCE	Distributing Computing Environment	ISO
SMB	System Message Block	Microsoft
LDAP	Lightweight Directory Access Protocol	IETF
AD	Active Directory	Microsoft



NIS (1) : définitions

domaine-NIS

défini par un nom (de domaine)
chaque système doit savoir à quel domaine-NIS, il appartient

map-NIS

un ensemble d'informations d'administration de même type
une map est construite à partir d'une source (fichier texte, ou autre forme)
exemple : la map `passwd.byname` est construite à partir de la source : `/etc/passwd`
copie maître, copies esclaves

serveur-NIS

un serveur-NIS maître par domaine, il possède la copie maître des maps
un ou plusieurs serveur-NIS esclaves par domaine, ils possèdent une copie esclave des maps
installé sur un système qui possède une copie des maps
il répond aux demandes d'accès aux maps

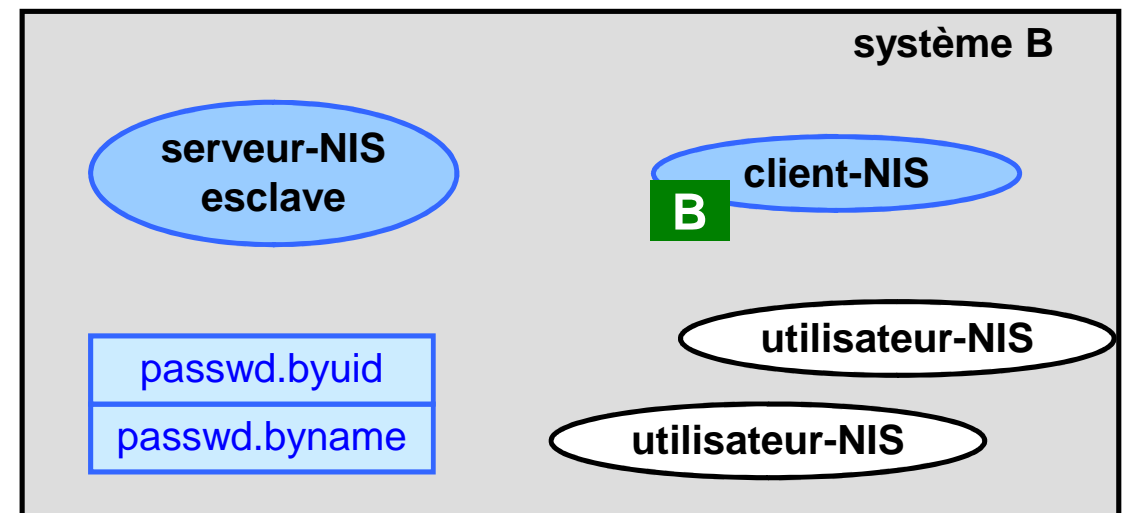
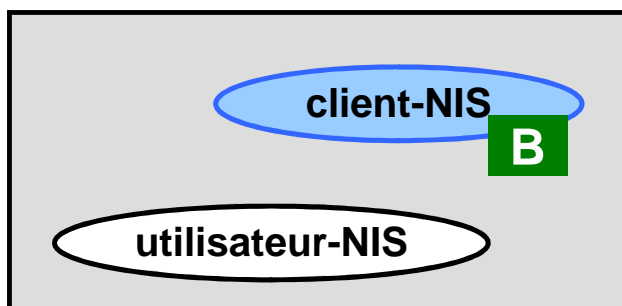
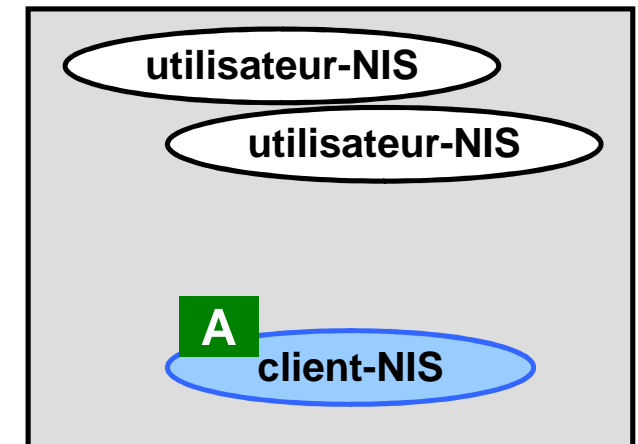
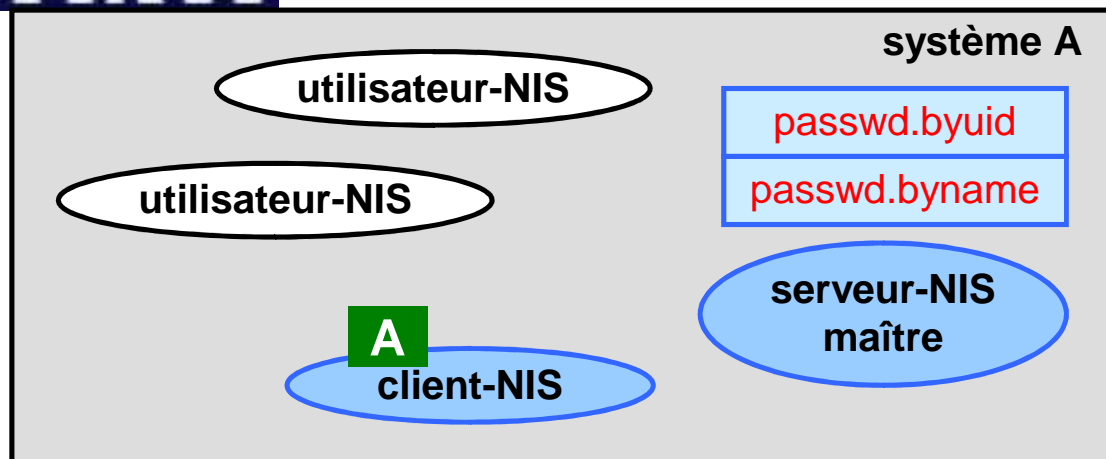
client-NIS

installé dans tout système du domaine qui utilise des informations d'administration fournies par le Service-NIS

utilisateur-NIS

entité qui a besoin d'accéder à des informations d'administration

NIS (2) : fonctionnement





NIS (3) : configuration

le nom de domaine-NIS

- dans un fichier de configuration, lu au démarrage du système
 - Linux : `/etc/sysconfig/network`

la copie maître des maps-NIS

- choix des maps : dans `/var/yp/Makefile`
- source des informations : fichiers texte, LDAP, BDR..
- initialisation par un script (`ypinit`)
- mise à jour par un script (`/var/yp/Makefile`)

le serveur-NIS maître
un serveur-NIS esclave
un client-NIS

<code>ypserv -m</code>	lancés au
<code>ypserv -s</code>	démarrage
<code>ypbind</code>	du système

un utilisateur-NIS

- ordre de consultation des sources d'informations d'administration
 - configuration dans `/etc/nsswitch.conf`
 - codes des sources `files yp dns`

DHCP : définitions

DHCP : Dynamic Host Configuration Protocol -> RFC 2131
(successeur de BOOTP)
utilisation de la couche transport UDP : port 67 client -> serveur
port 68 serveur -> client

Serveur fournit (en général) :

- Adresse IP
- Masque de réseau
- Adresse de la passerelle
- Adresse du DNS, nom du domaine

Un serveur a un « pool » d'adresses qu'il peut distribuer
➡ utilisation d'un **bail** sur la durée de location d'une adresse

DHCP : fonctionnement

1. Emission d'un broadcast pour la demande d'une adresse IP
-> *DHCPDISCOVER*
2. Réponse en unicast d'un serveur vers le client en lui spécifiant les différents paramètres à utiliser
(auparavant le serveur fait 2 pings pour savoir si l'adresse qu'il propose n'est pas déjà utilisée...)
-> *DHCP OFFER*
3. Client donne son accord sur cette adresse en faisant un broadcast
-> *DHCP REQUEST*
4. Le serveur acquitte cette accord en répondant en unicast
-> *DHCP ACK*

Pour libérer une adresse IP, utilisation de *DHCP RELEASE*

Protocole non routable



MSN (1)

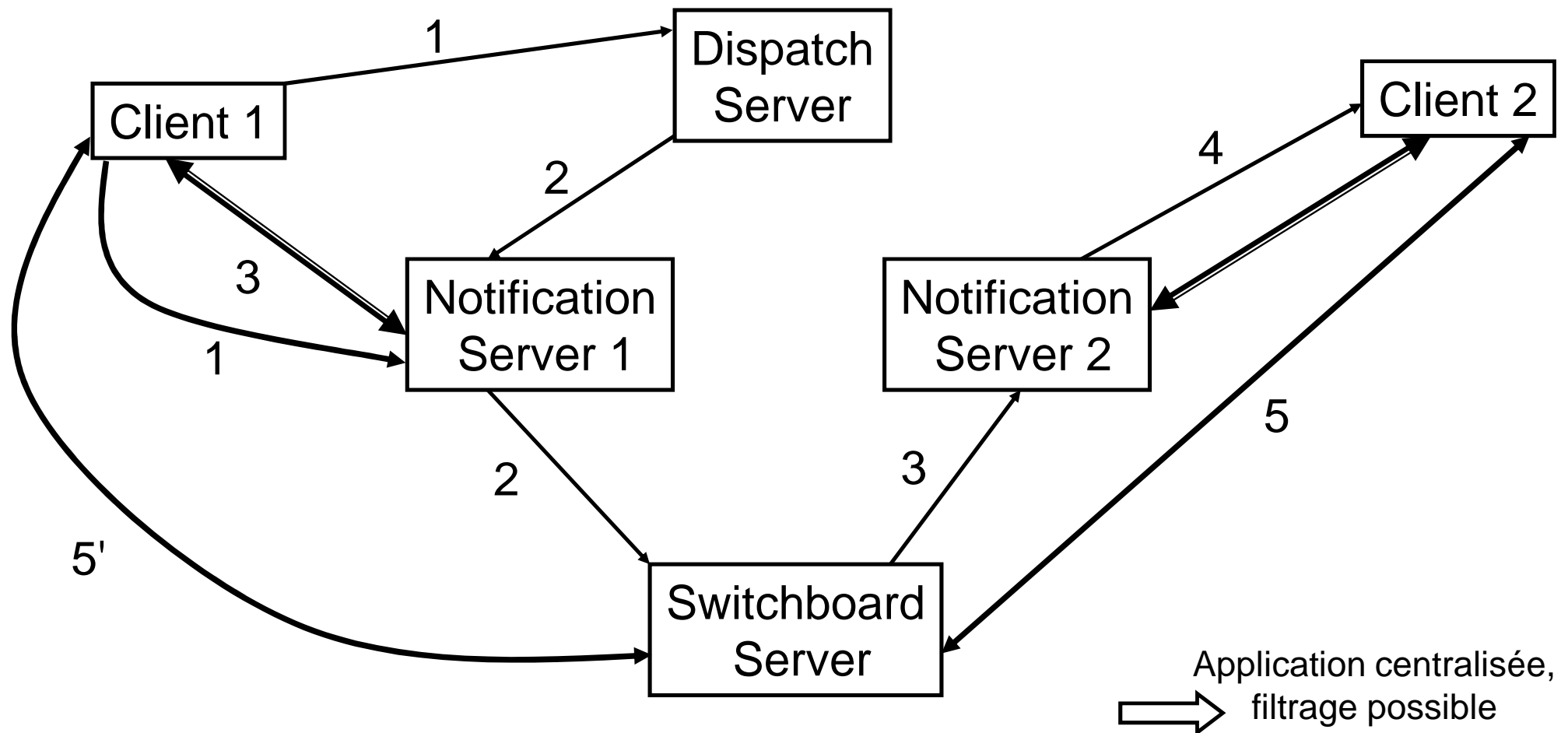
MSN : Microsoft Network

→ actuellement : Windows Live Manager

- Logiciel propriétaire
 - Messagerie instantanée ou non avec gestion des "contacts"
 - VoIP
 - Dossier partagée
 - Tableau blanc
 - Calendrier partagé
- Gestion des utilisateurs via un pseudonyme
 - 4 états possibles : Disponible, Occupé, Absent, Hors ligne
- Protocole Utilisé → Microsoft Notification Protocol (MSNP)
 - au-dessus de TCP
 - Protocole propriétaire (donc non divulgué), version 16 actuellement

MSNP, version 2

MSNP : sur TCP, port 1863 → version 2 est la seule version publiée



XMPP (1)

XMPP : Extensible Messaging and Presence Protocol
→ autre nom possible : Jabber

- **Spécification**
 - Messagerie instantanée point à point ou point à multipoint
 - Utilisation de XML pour décrire les messages
 - Définit dans les RFC 3920 - 3923, donc libre de droit
- **Utilisation**
 - Gestion des états de l'utilisateur
 - Utilisé par : google, facebook, ...
 - Gestion de la sécurité si nécessaire
 - Connexion permanente sur le serveur xmpp

XMPP (2)

- Utilisation d'une adresse unique pour l'utilisateur : nom@machine
(JID -> Jabber Identifier)
- Possibilité de rejoindre la communauté Jabber (mais non effectuée en général)

