Systèmes de fichiers réseau

Network Attached Storage (NAS)
une interface réseau au niveau fichier

Philippe Latu / Université Toulouse III - Paul Sabatier / www.inetdoc.net

Philippe.latu(at)inetdoc.net

- **Le système de fichiers virtuel du noyau**
  - Virtual File System (VFS)
- **Les services Internet et les échanges de fichiers**
  - FTP – SMTP – P2P – SSH – HTTP - DAV
- **Le mécanisme d'appel de procédure distant**
  - Remote Procedure Call (RPC)
- **Les caractéristiques de deux systèmes de fichiers**
  - Network File System (NFS)
  - Server Message Block (SMB) ou Common Internet File System (CIFS)

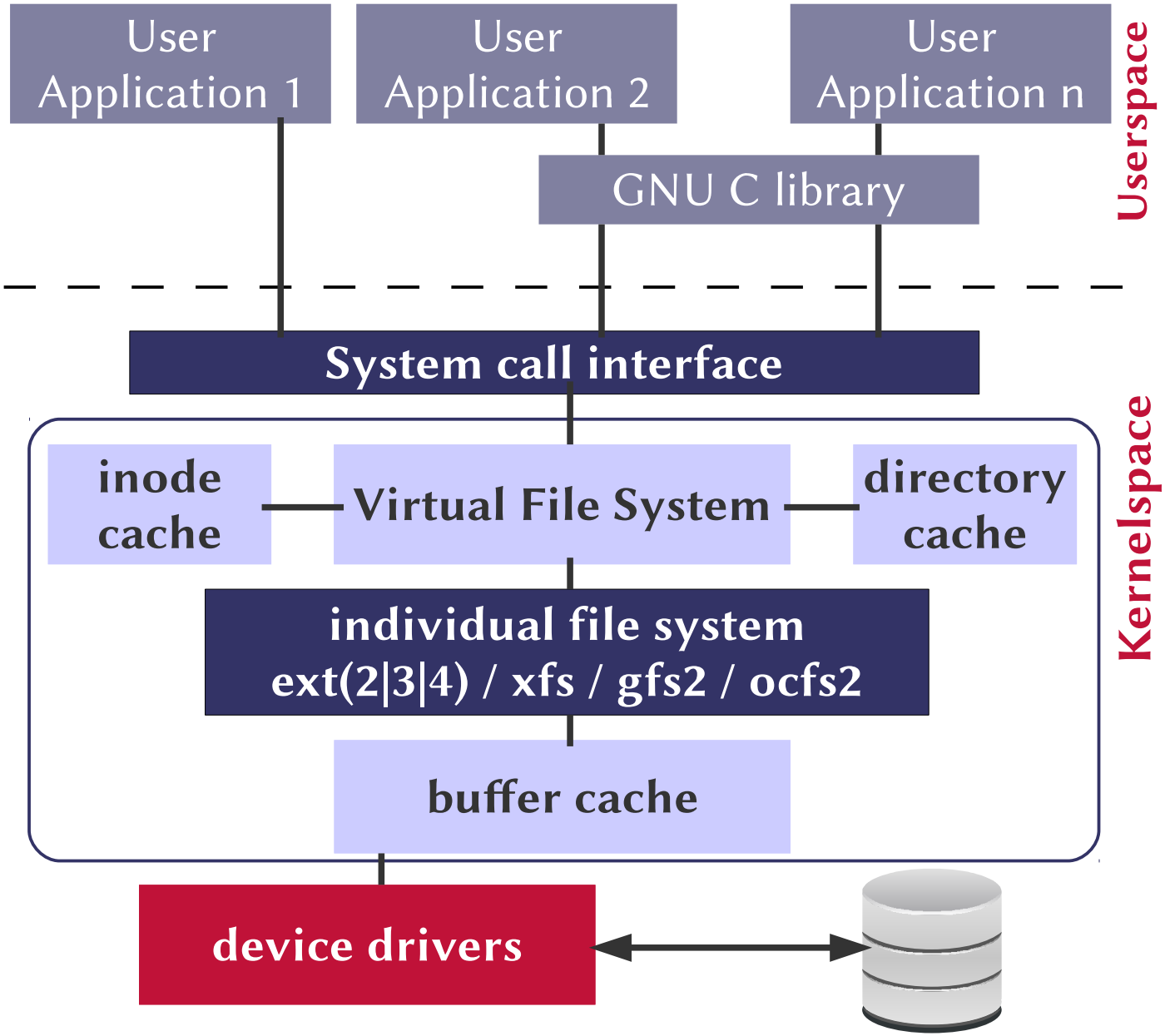
Qu'est-ce qu'un système de fichiers virtuel ?

- Une couche logicielle du noyau
- Une interface entre les processus et les dispositifs de stockage
- Une bibliothèque de programmation standard

Quels sont les objectifs ?

- Un accès transparents vis-à-vis des utilisateurs
- Une gestion uniforme du contrôle d'accès aux fichiers et répertoires
- Un système de nommage cohérent entre stockage réseau et stockage local
- Un accès fiable, sécurisé
- Un temps d'accès le plus proche possible du stockage local

Kernel Virtual File System

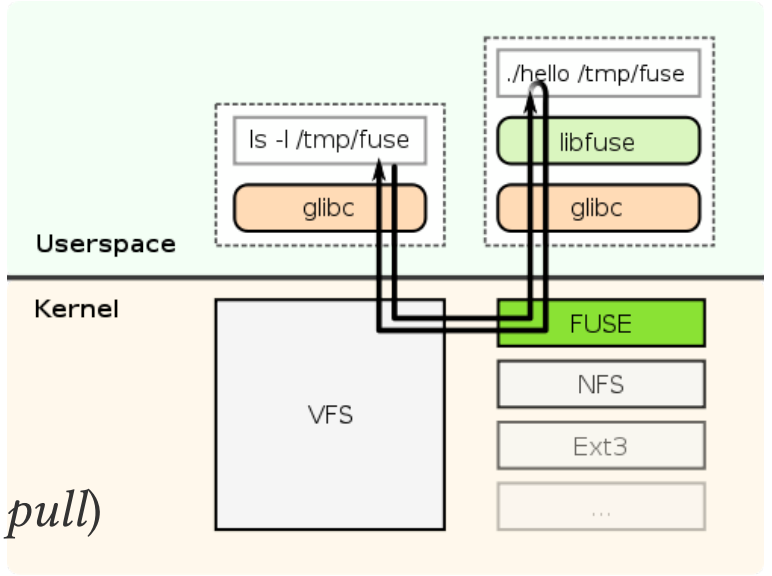


Quelles sont les contraintes ?

- Une croissance très rapide de la demande
  - Besoins en stockage très supérieurs aux besoins en puissance de calcul
- Une grande hétérogénéité
  - Disparité entre générations de systèmes de fichiers
  - Un exemple : FAT16 vs. GFS2
- Des usages très différents
  - Domestiques : VFAT, NTFS
  - Serveurs : ext4, XFS
  - Clusters : GFS2, OCFS2
  - Réseau : CIFS, NFS

Pourquoi plusieurs types d'échanges de fichiers ?

- *File Transfer Protocol* (FTP)
  - Passé de mode !
- *Simple Mail Tranfser Protocol* (SMTP)
  - Pièces jointes ou virus, spams, etc
- *Hyper Text Transfer Protocol* (HTTP)
  - Conçu pour des transferts unidirectionnels (mode *pull*)
- *Secure Shell* (SSH)
  - Pas assez à la mode !
- *Peer to Peer* (P2P)
  - Solution dévoyée par le téléchargement illégal



FUSE
Filesystem in userspace

Quels objectifs ne sont pas satisfaits ?

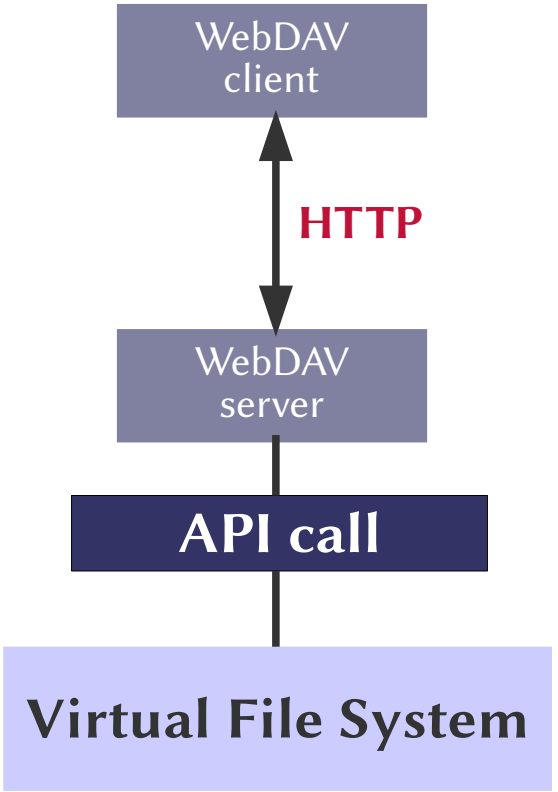
- Une gestion uniforme du contrôle d'accès aux fichiers et répertoires
- Un système de nommage cohérent entre stockage réseau et stockage local

Qu'en est-il du compromis HTTP/WebDAV ?

- Supporté par les principaux serveurs Web
  - Apache, IIS, etc
- Fonctionnement proche d'un système de fichiers

Substitution difficile

- Trop lent pour les transferts entre serveurs
- Fonctions manquantes relativement à CIFS et NFS
  - Synchronisation
  - Réplication synchrone ou asynchrone
- Non conforme POSIX
- Manque de support côté applications



Quels sont les besoins non satisfaits ?

- Balance de charge entre serveurs multiples
- Réplication automatisée
- Contrôle d'accès uniforme aux fichiers et répertoires
- Clients légers sans disque local

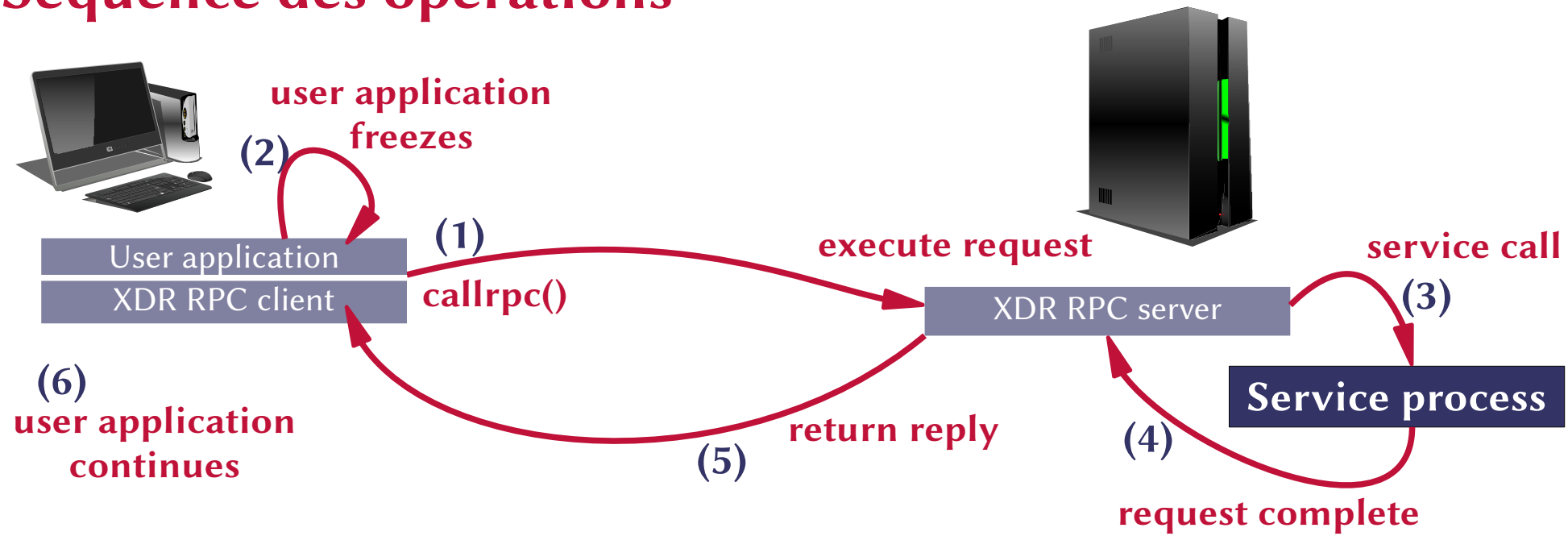
Les systèmes de fichiers sont indispensables !

- Fonction critique quel que soit l'usage
- Adaptation du rapport performances / volume de stockage
- Solutions NAS = bon rapport performances / coût

Mécanisme *Remote Procedure Call* (RPC)

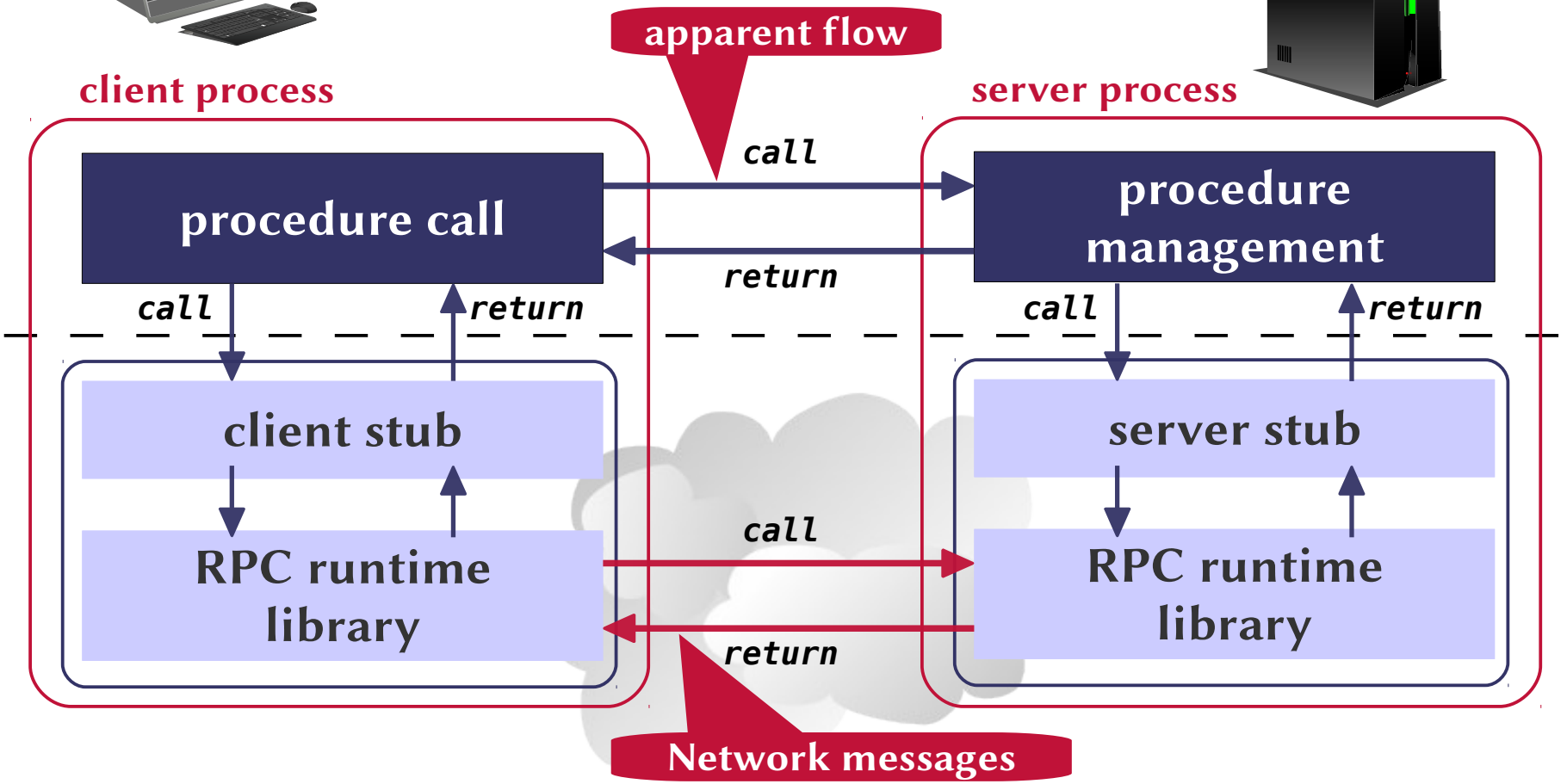
- Étendre les appels de procédures locaux
  - Contrôler et transférer des données sur un réseau
  - Réaliser des appels de sous-programmes analogues aux appels locaux
  - Exécuter le code d'un sous-programme sur une machine distante
- Adapter les traitements au modèle client - serveur

Séquence des opérations



Modélisation des flux RPC

- Communication de bout en bout
- Bibliothèques modulaires



Qu'est-ce que le protocole NFS ?

- Un ensemble de procédures avec leurs paramètres
- Une méthode d'accès réseau à des fichiers et des répertoires
- Un standard de référence
  - Développement initié par Sun Microsystems au début des années 80
  - Spécifications ouvertes depuis 1989 : RFC 1094

Protocoles RPC & XDR

- Remote Procedure Calls (RPC) version 2
  - RFC 1057 : gestion d'appels de procédures entre client et serveur
- eXternal Data Representation
  - RFC 1014 : Description et encodage des données

Exploité depuis plus de 20 ans

- Performances moyennes mais fiabilité largement éprouvée
- Calcul scientifique, CAO, gros volumes

NFS v2 - 1985

- Exportation de systèmes de fichiers POSIX 32 bits
- Protocole de transport UDP uniquement sur réseaux locaux
- Performances médiocres en écriture

NFS v3 - 1994

- Exportation de systèmes de fichiers POSIX 64 bits
- Protocoles de transport UDP et TCP toujours avec multiplexage de ports
- Performances améliorées en écriture
- Version la plus répandue
  - Serveurs NAS basés sur des distributions BSD et GNU/Linux
  - Réplication asynchrone entre serveurs

NFS v4.x

- Adaptation aux évolutions de l'Internet
- Exploitation sur réseaux étendus
- Réduction des temps de latence
- Communications sur un port unique : tcp/2049 (filtrage plus facile)
- Appels de procédures groupés : *compound* RPC

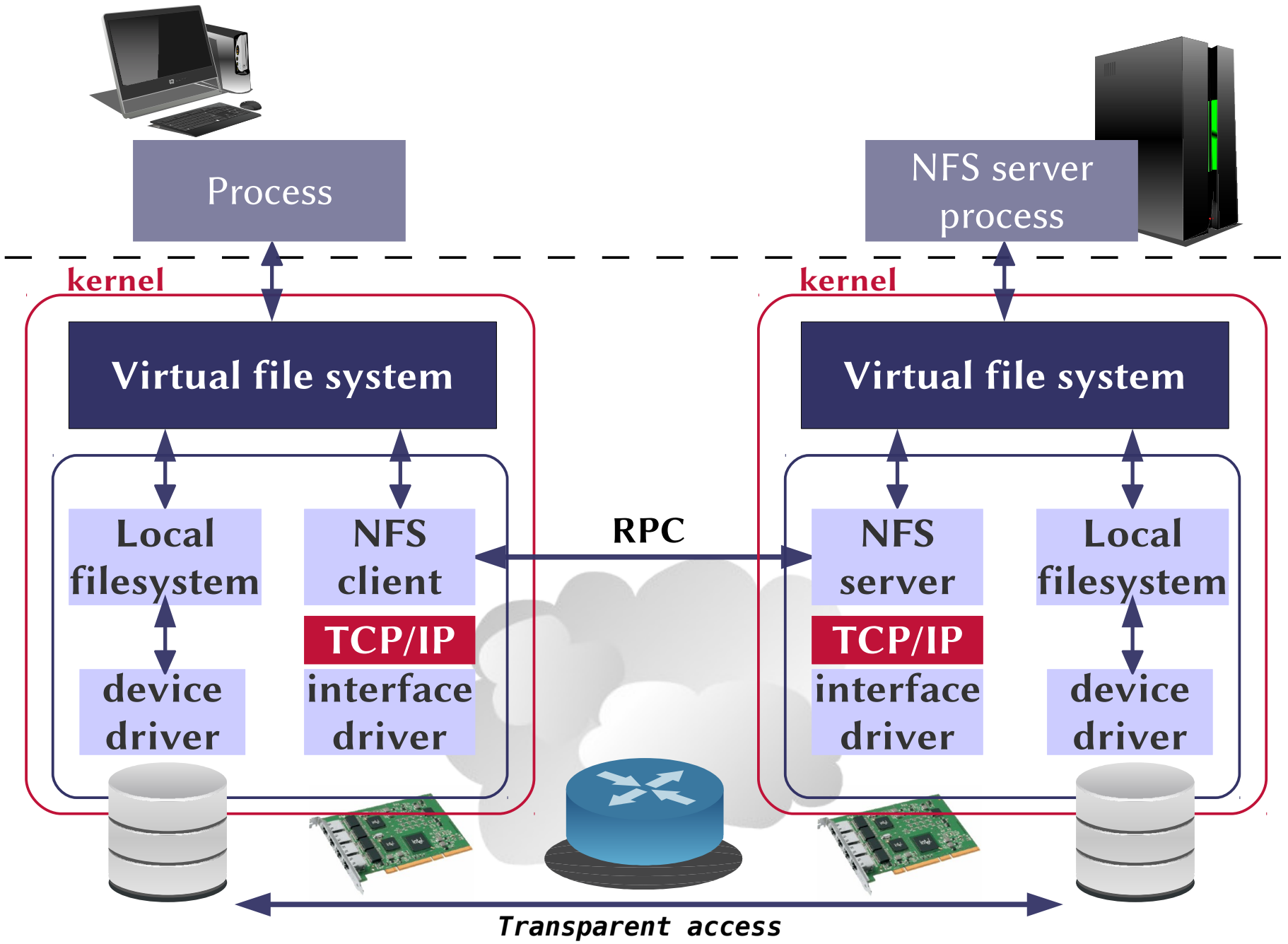
Contrôles renforcés

- Chiffrement des flux
- Cryptage asymétrique basé sur un couple de clés publique et privée
- Contrôle d'accès sur les fichiers et répertoires

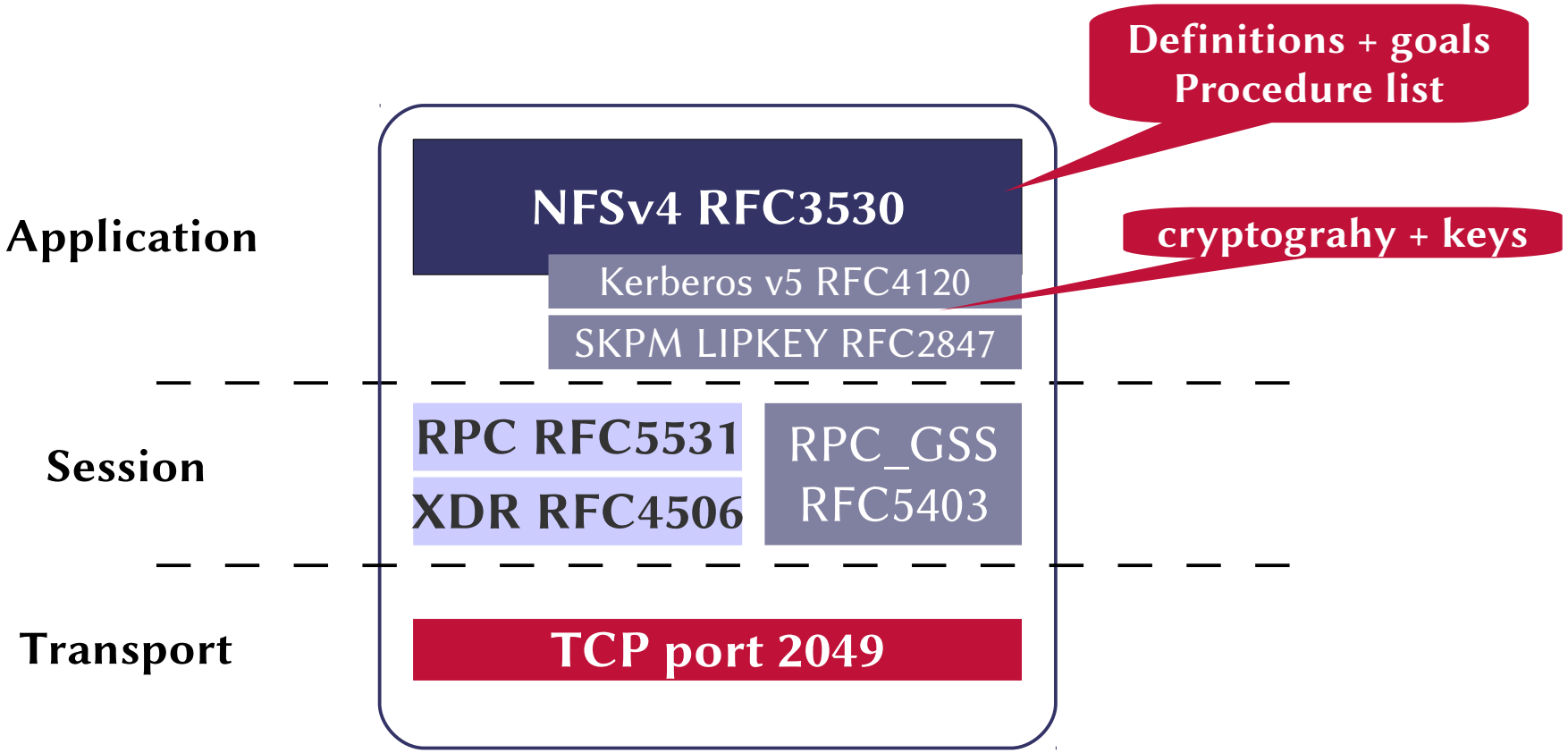
Développements en cours

- Schéma de nommage global
- Réplication synchrone
- Mobilité

- Modélisation des flux



- Pile des protocoles NFSv4.x



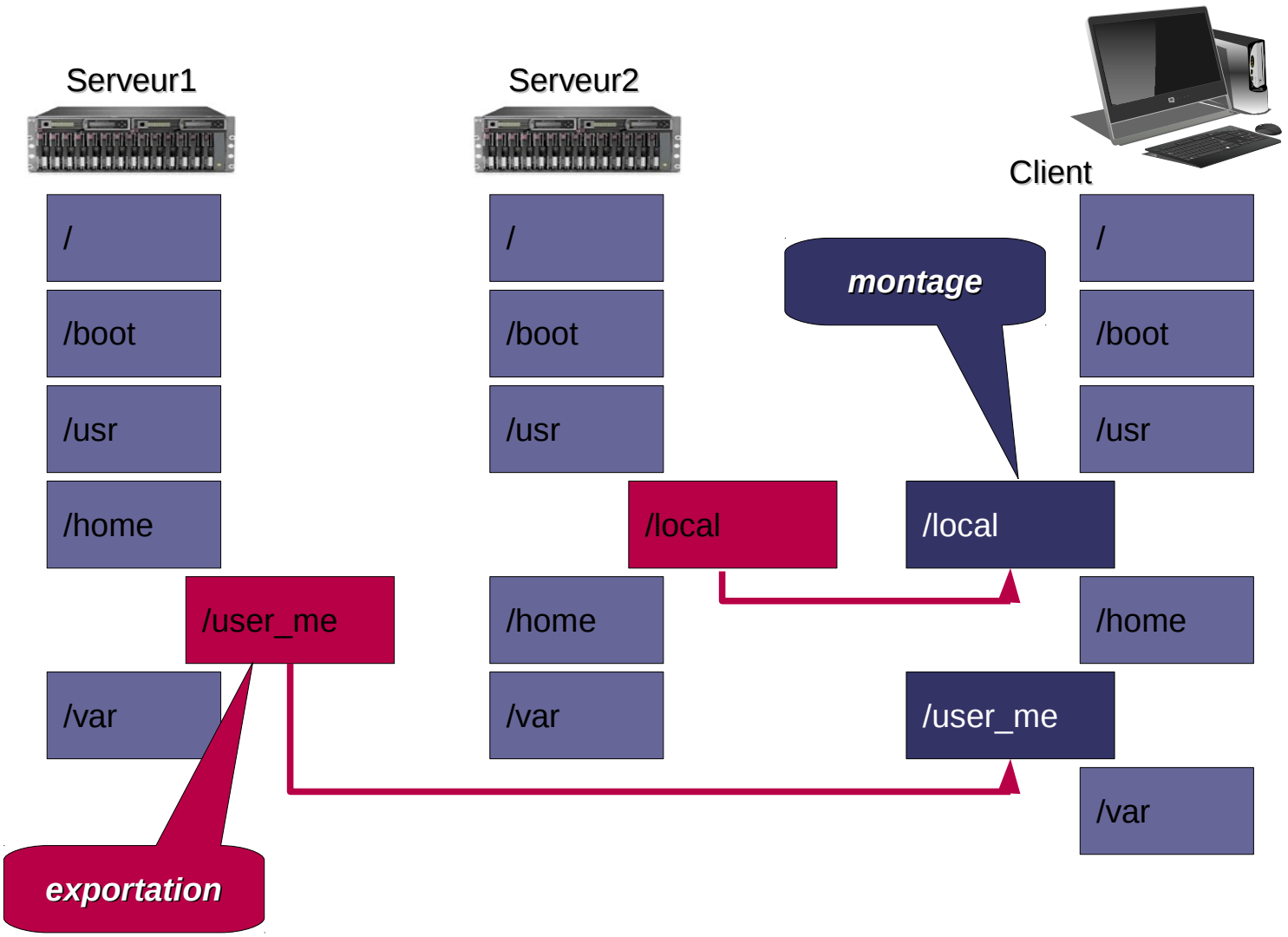
Service sans suivi de session

- Chaque appel de procédure (RPC) est indépendant
- Pas de trace conservée des appels précédents
- Amélioration du temps de reprise sur erreur
  - Aucun état à récupérer
  - Émission de nouveaux appels RPC côté client

Stockage transparent

- Arborescence partagée entre client et serveur
- Une arborescence est ***exportée*** par le serveur
- Une arborescence est ***montée*** sur un volume local par le client

- Arborescences des systèmes de fichiers



Service de montage

- Commande *mount*
- Montage d'un système de fichiers distant sur un volume local

Côté client

- Aucun processus dans l'espace utilisateur

```
$ mount | grep nfs
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
192.200.0.3:/home/etu-nfs on /ahome/etu-nfs type nfs4 (rw,addr=192.200.0.3,clientaddr=192.200.0.4)
```

```
# rpcinfo -s
   program version(s) netid(s)                         service     owner
    100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
```

Côté serveur

- Un seul processus : le gestionnaire de montage

```
# rpcinfo -s
   program version(s) netid(s)                         service     owner
    100000  2,3,4     local,udp,tcp,udp6,tcp6          portmapper  superuser
    100003  4,3,2     udp6,tcp6,udp,tcp                nfs         unknown
    100227  3,2       udp6,tcp6,udp,tcp                -           unknown
    100021  4,3,1     tcp6,udp6,tcp,udp                nlockmgr    unknown
    100005  3,2,1     tcp6,udp6,tcp,udp                mountd      superuser
```

Exemple d'utilisation du service de montage

Côté serveur

- Exportation du répertoire /home/exports

```
# exportfs
/home/exports   192.200.0.0/27
/home/exports/home
                192.200.0.0/27
```

réseau des clients autorisés à accéder au service

répertoires exportés

- Montage local
  - Cohérence du système de nommage entre clients et serveurs
  - Indépendance entre volume physique et espace de nommage

```
# mount --bind /home/exports/home /ahome
```

vue logique

vue physique

Côté client

- Montage vers le répertoire /ahome

```
$ mount | grep nfs
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
192.200.0.3:/home/etu-nfs on /ahome/etu-nfs type nfs4 (rw,addr=192.200.0.3,clientaddr=192.200.0.4)
```

vue serveur

vue client

Analyse réseau des appels RPC

```
192.200.0.4 -> 192.200.0.3  TCP wpages > nfs [SYN] Seq=0 Win=14600 Len=0 MSS=1460
192.200.0.3 -> 192.200.0.4  TCP nfs > wpages [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460
192.200.0.4 -> 192.200.0.3  TCP wpages > nfs [ACK] Seq=1 Ack=1 Win=14624 Len=0
192.200.0.4 -> 192.200.0.3  NFS V4 NULL Call
192.200.0.3 -> 192.200.0.4  TCP nfs > wpages [ACK] Seq=1 Ack=45 Win=14496 Len=0
192.200.0.3 -> 192.200.0.4  NFS V4 NULL Reply (Call In 6)
192.200.0.4 -> 192.200.0.3  TCP wpages > nfs [ACK] Seq=45 Ack=29 Win=14624 Len=0
192.200.0.4 -> 192.200.0.3  NFS V4 COMPOUND Call <EMPTY> PUTROOTFH;GETFH;GETATTR
192.200.0.3 -> 192.200.0.4  NFS V4 COMPOUND Reply (Call In 10) <EMPTY> PUTROOTFH;GETFH;GETATTR
…
192.200.0.4 -> 192.200.0.3  NFS V4 COMPOUND Call <EMPTY> SETCLIENTID
192.200.0.3 -> 192.200.0.4  NFS V4 COMPOUND Reply (Call In 55) <EMPTY> SETCLIENTID
192.200.0.4 -> 192.200.0.3  NFS V4 COMPOUND Call <EMPTY> SETCLIENTID_CONFIRM;PUTROOTFH;GETATTR
192.200.0.3 -> 192.200.0.4  NFS V4 COMPOUND Reply (Call In 57) <EMPTY> SETCLIENTID_CONFIRM;PUTROOTFH;GETATTR
192.200.0.3 -> 192.200.0.4  TCP 738 > 41885 [SYN] Seq=0 Win=14600 Len=0 MSS=1460
192.200.0.4 -> 192.200.0.3  TCP 41885 > 738 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460
192.200.0.3 -> 192.200.0.4  TCP 738 > 41885 [ACK] Seq=1 Ack=1 Win=14624 Len=0
192.200.0.3 -> 192.200.0.4  NFS V1 CB_NULL Call
192.200.0.4 -> 192.200.0.3  TCP 41885 > 738 [ACK] Seq=1 Ack=69 Win=14496 Len=0
192.200.0.4 -> 192.200.0.3  NFS V4 COMPOUND Call <EMPTY> PUTFH;SAVEFH;OPEN;LOCKT;GETATTR;RESTOREFH;GETATTR
192.200.0.4 -> 192.200.0.3  NFS V1 CB_NULL Reply (Call In 62)
192.200.0.3 -> 192.200.0.4  TCP 738 > 41885 [ACK] Seq=69 Ack=29 Win=14624 Len=0
```

- Service d'automontage
  - Montage dynamique
    - Utilisation des ressources réseau à la demande
    - Utilisation de règles génériques ou *wildcards*
  - Adressage indirect
    - 1 définition d'une racine de montage
      - Fichier /etc/auto.master

```
# grep -v ^# /etc/auto.master
/ahome  /etc/auto.home
```

racine

    - 2 définition des paramètres de montage
      - Fichier /etc/auto.home

```
# cat /etc/auto.home
*       -fstype=nfs4    192.200.0.3:/home/&
```

  - Service de réplication basique
    - Extension d'un système de fichiers en ligne

Common Internet File System (CIFS)

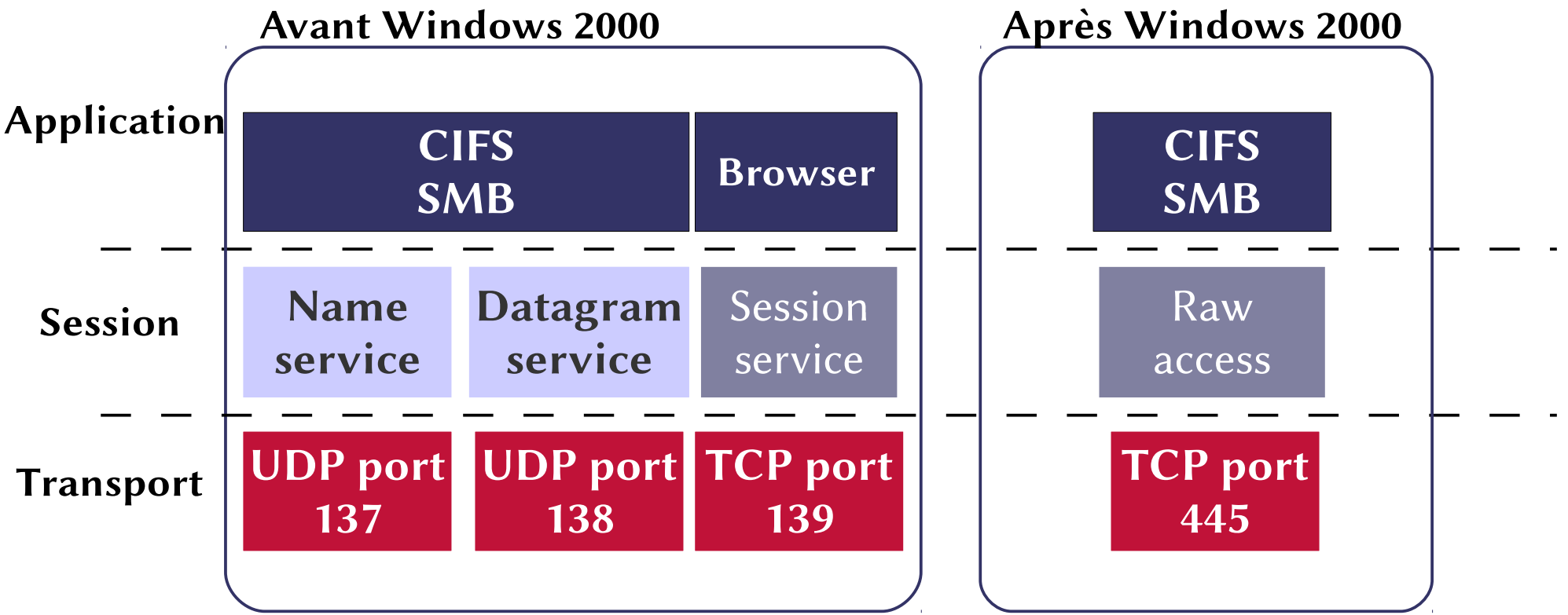
CIFS ou SMB

- Développé par IBM au début des années 80
- Conçu pour utiliser NetBIOS dans un même réseau local
- Étendu par Microsoft lors du développement de Windows 2000
  - Compatibilité NetBIOS maintenue
  - Contrôles d'accès différents des spécifications POSIX
  - Nommage global – identification unique sur le serveur et le client
  - Plus qu'un système de fichiers – gestion des imprimantes, fax, etc

SMB2

- Protocole de système de fichier par défaut depuis Windows 2008
- Performances améliorées – gestion de cache, taille de blocs, etc
  - Mêmes objectifs que pour NFS
- Protocole très utilisé dans les solutions NAS
  - Supporté sur Linux côté serveur - Samba

- Pile de protocole CIFS - SMB



Exemple d'utilisation

```
phil@[192.168.1.1]:~$ rpcclient -U phil 192.168.1.6 Password:
rpcclient $> enumprivs
found 5 privileges

SeMachineAccountPrivilege       0:6 (0x0:0x6)
SeSecurityPrivilege             0:8 (0x0:0x8)
SeTakeOwnershipPrivilege        0:9 (0x0:0x9)
SaAddUsers                      0:65281 (0x0:0xff01)
SaPrintOp                       0:65283 (0x0:0xff03)

rpcclient $> exit
```

Analyse réseau

```
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [SYN] …
192.168.1.6 -> 192.168.1.1 TCP microsoft-ds > 58296 [SYN, ACK] …
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [ACK] …
192.168.1.1 -> 192.168.1.6 SMB Negotiate Protocol Request
192.168.1.6 -> 192.168.1.1 TCP microsoft-ds > 58296 [ACK] …
192.168.1.6 -> 192.168.1.1 SMB Negotiate Protocol Response
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [ACK] …
192.168.1.1 -> 192.168.1.6 SMB Session Setup AndX Request, User: \phil
192.168.1.6 -> 192.168.1.1 TCP microsoft-ds > 58296 [ACK] ...
192.168.1.6 -> 192.168.1.1 SMB Session Setup AndX Response
192.168.1.1 -> 192.168.1.6 SMB Tree Connect AndX Request, Path: \\192.168.1.6\IPC$
192.168.1.6 -> 192.168.1.1 SMB Tree Connect AndX Response
192.168.1.1 -> 192.168.1.6 SMB NT Create AndX Request, Path: \lsarpc
192.168.1.6 -> 192.168.1.1 SMB NT Create AndX Response, FID: 0x76b2
```

Analyse réseau (suite) – appel au service

```
192.168.1.1 -> 192.168.1.6 DCERPC Bind: call_id: 1 UUID: LSA
192.168.1.6 -> 192.168.1.1 DCERPC Bind_ack: call_id:1 accept max_xmit: 4280 max_recv: 4280
192.168.1.1 -> 192.168.1.6 LSA LsarOpenPolicy request
192.168.1.6 -> 192.168.1.1 LSA LsarOpenPolicy response
192.168.1.1 -> 192.168.1.6 LSA LsarQueryInformationPolicy request,Account Domain Information
192.168.1.6 -> 192.168.1.1 LSA LsarQueryInformationPolicy response
192.168.1.1 -> 192.168.1.6 LSA LsarClose request
192.168.1.6 -> 192.168.1.1 LSA LsarClose response
192.168.1.1 -> 192.168.1.6 SMB Close Request, FID: 0x76b2
192.168.1.6 -> 192.168.1.1 SMB Close Response
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [ACK] …
192.168.1.1 -> 192.168.1.6 SMB NT Create AndX Request, Path: \lsarpc
192.168.1.6 -> 192.168.1.1 SMB NT Create AndX Response, FID: 0x76b3
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [ACK] ...
```

Analyse réseau (suite) – déconnexion

```
192.168.1.1 -> 192.168.1.6 DCERPC Bind: call_id: 5 UUID: LSA
192.168.1.6 -> 192.168.1.1 DCERPC Bind_ack: call_id:5 accept max_xmit: 4280 max_recv: 4280
192.168.1.1 -> 192.168.1.6 LSA LsarOpenPolicy request
192.168.1.6 -> 192.168.1.1 LSA LsarOpenPolicy response
192.168.1.1 -> 192.168.1.6 LSA LsarEnumeratePrivileges request
192.168.1.6 -> 192.168.1.1 LSA LsarEnumeratePrivileges response
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [ACK] …
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [FIN, ACK] …
192.168.1.6 -> 192.168.1.1 TCP microsoft-ds > 58296 [FIN, ACK] …
192.168.1.1 -> 192.168.1.6 TCP 58296 > microsoft-ds [ACK] ...
```

- Évolutions parallèles
  - Côté serveur
    - Caractéristiques de plus en plus proches
    - Gestion de cache
    - Tailles de blocs
    - Conformité POSIX attendue
  - Côté client
    - Avantage au protocole CIFS sur les systèmes Microsoft
    - Client libre NFSv4.x Windows attendu
    - Client SMB2 Linux attendu

Ressources

- Présentations
  - http://www.connectathon.org/
  - http://www.sambaxp.org/
  - http://tinyurl.com/snianfsv4
- Travaux pratiques
  - http://tinyurl.com/inetdoctpnfs