

[inetdoc.LINUX]

<http://www.linux-france.org/prj/inetdoc>

Administration Système & Réseau

- Étude de 2 systèmes de fichiers réseau
 - ▶ Système de fichiers virtuel
 - ▶ Services Internet & fichiers
 - ▶ Network File System (NFS)
 - ▶ Remote Procedure Call (RPC)
 - ▶ Common Internet File System (SMB|CIFS)



Philippe Latu

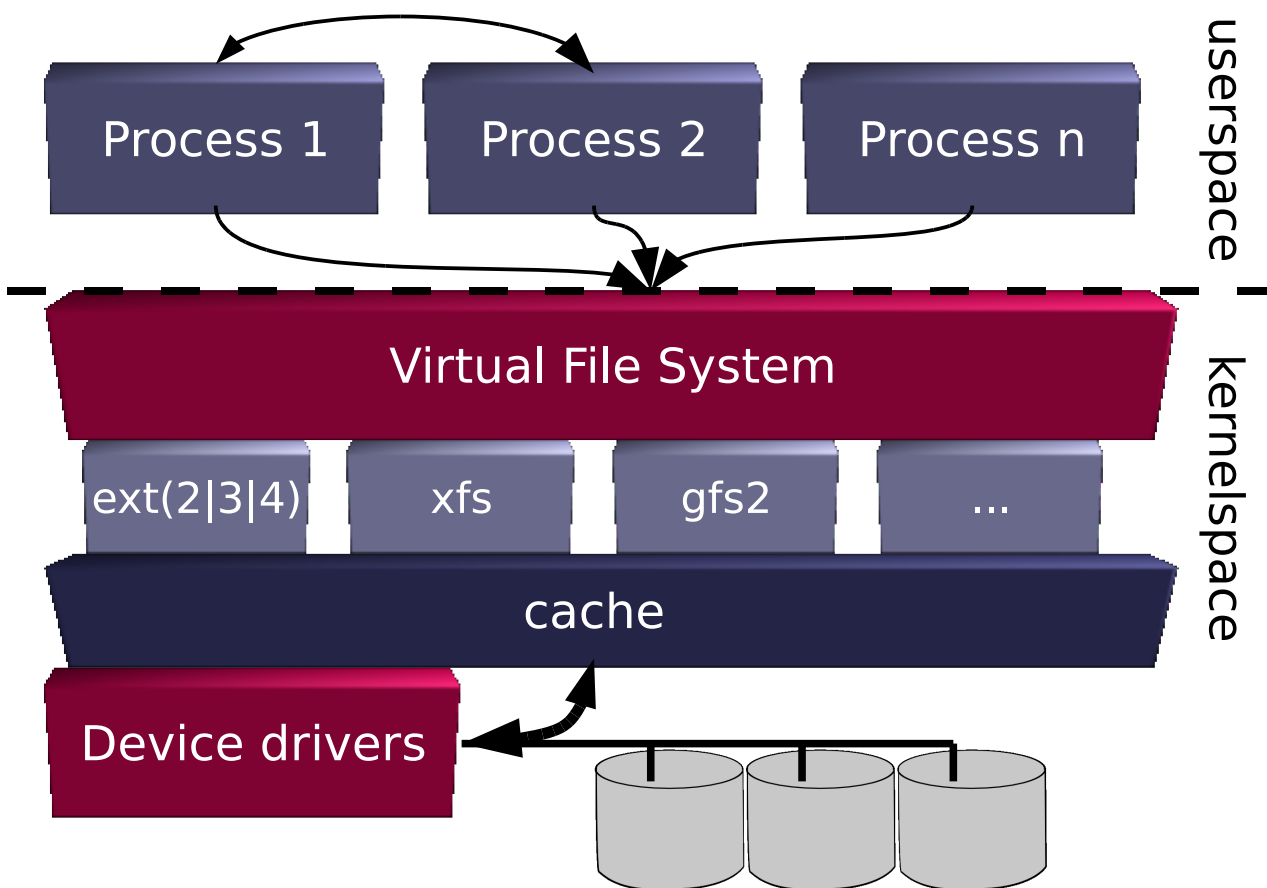
philippe.latu@linux-france.org

IUT 'A' Paul Sabatier - IUP STRI

Systeme de fichiers virtuel

- Qu'est-ce qu'un système de fichiers virtuel ?
 - Couche logicielle du noyau
 - Interface entre les processus et le système de fichiers
 - Utilisation simultanée de systèmes de fichiers différents
 - ▶ http://www.linuxdriver.co.il/kernel_map
- Quelles sont les appellations connues ?
 - Noyau Linux = VFS
 - Noyau Microsoft = IFS

Systeme de fichiers virtuel



Systeme de fichiers virtuel

- Quels sont les objectifs ?
 - Transparence totale vis-à-vis des utilisateurs
 - Stockage réseau fiable, rapide et sécurisé

- Quelles sont les contraintes ?
 - Croissance rapide de la demande
 - ▶ Besoins en stockage réseau > Besoins en puissance de calcul
 - Coexistence problématique
 - ▶ Grande disparité de systèmes de fichiers traditionnels
 - Usages différents
 - ▶ Domestique : VFAT, NTFS
 - ▶ Serveurs : Ext(2|3|4)
 - ▶ Clusters : GFS2, OCFS2
 - ▶ Réseau : CIFS, NFS

Services Internet & fichiers

- Pourquoi plusieurs types d'échanges de fichiers ?
 - Service FTP
 - ▶ plus vraiment à la mode !
 - Service SMTP
 - ▶ virus, spams ou pièces jointes ?
 - Service P2P
 - ▶ trop à la mode pour une utilisation professionnelle !
 - Service HTTP
 - ▶ conçu pour des transferts unidirectionnels
 - Service SSH
 - ▶ pas assez à la mode !

Services Internet & fichiers

- Qu'en est-il du compromis HTTP/WebDAV ?
 - Supporté par les principaux serveurs Web
 - ▶ Apache, IIS, etc.
 - Caractéristiques voisines des systèmes de fichiers
 - Supposé se développer avec les usages de l'Internet
- Substitution difficile malgré tout
 - Trop lent pour être utilisé dans le contexte serveur
 - Manque de fonctions relativement à CIFS et NFS
 - ▶ Synchronisation
 - ▶ Réplication synchrone et asynchrone
 - Non conforme POSIX
 - Manque de support côté applications

Services Internet & fichiers

- Quels sont les besoins non satisfaits ?
 - Balance de charge entre plusieurs serveurs
 - Réplication automatisée
 - Accès transparents aux fichiers
 - Postes clients sans disque dur

- Systèmes de fichiers toujours indispensables
 - Fonction critique dans une infrastructure
 - Rapport Performances/Volume de stockage

Network File System (NFS)

- Qu'est-ce que le protocole NFS ?
 - Un ensemble de procédures avec leurs arguments
 - Une méthode d'accès transparente aux fichiers distants
 - Un standard de fait
 - ▶ Développé par Sun Microsystems au début des années 80
 - ▶ Standard ouvert aujourd'hui : RFC 1094
- Protocoles RPC/XDR
 - Remote Procedure Calls (RPC) version 2
 - ▶ Gestion d'appels de procédures entre client et serveur (RFC 1057)
 - eXternal Data Representation
 - ▶ Standard de description et d'encodage des données (RFC 1014)
- NFS est exploité depuis 20 ans
 - Performances moyennes mais fiabilité éprouvée
 - Utilisation en calcul scientifique, CAO

Network File System (NFS)

- NFS v2 - 1985

- Exportation de système de fichiers POSIX 32 bits
- Réseaux locaux avec le protocole de transport UDP
- Performances médiocres en écriture

- NFS v3 - 1994

- Exportation de système de fichiers POSIX 64 bits
- Réseaux locaux avec les protocoles UDP ou TCP
- Performances accrues en écriture

- NFS v3 = Solution très répandue

- Solutions NAS *BSD/Linux
- Réplication entre serveurs

Network File System (NFS)

- NFS v4

- Spécifications ouvertes à base de RFCs

- Adaptations nouvelles

- ▶ Adaptation aux réseaux étendus complexes
- ▶ Prise en charge du filtrage : port unique tcp/2049
- ▶ Réduction des temps de latence
- ▶ Appels RPC groupés : compound

- Sécurité renforcée

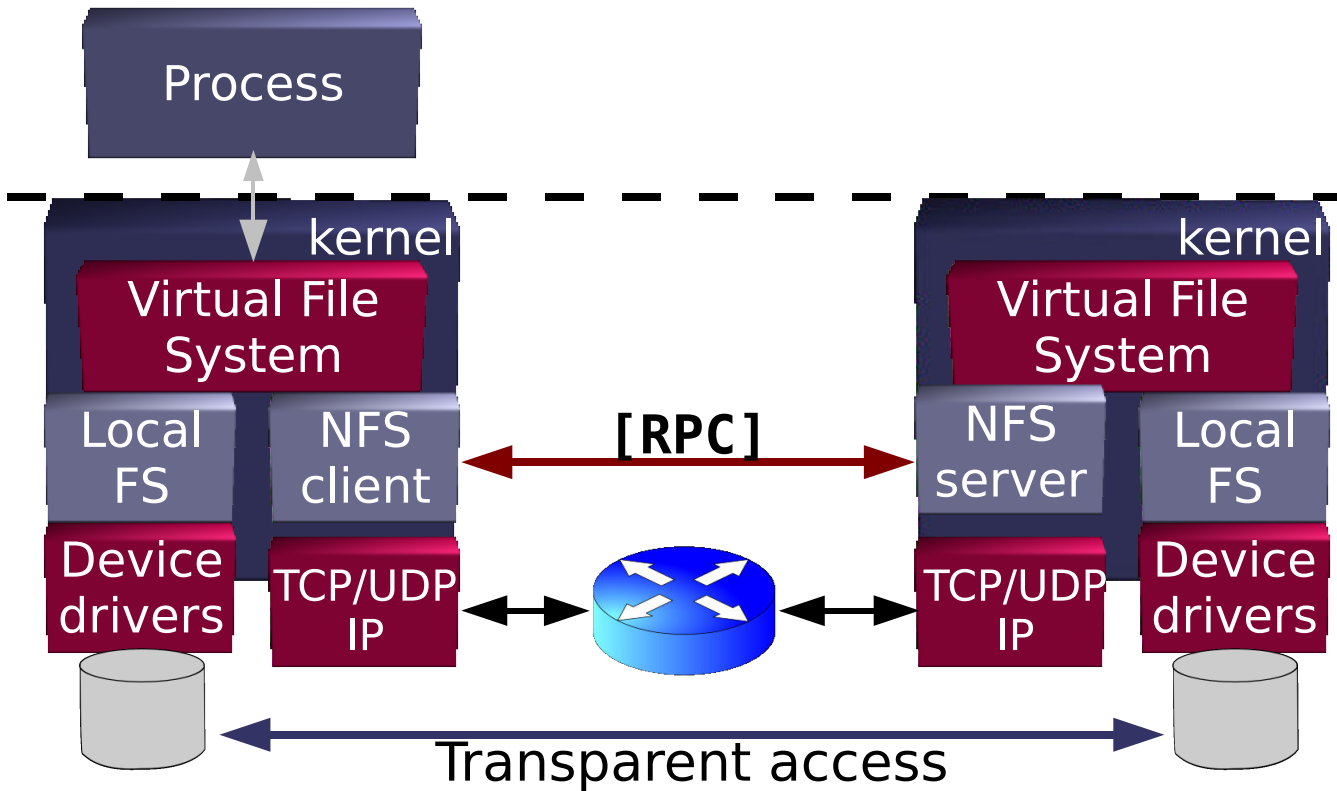
- ▶ Chiffrement des flux
- ▶ Cryptage asymétrique : clés privées/publiques
- ▶ Contrôle d'accès à granularité fine

- Chantiers en cours

- ▶ Mobilité
- ▶ Réplication
- ▶ Espace de nommage globalisé

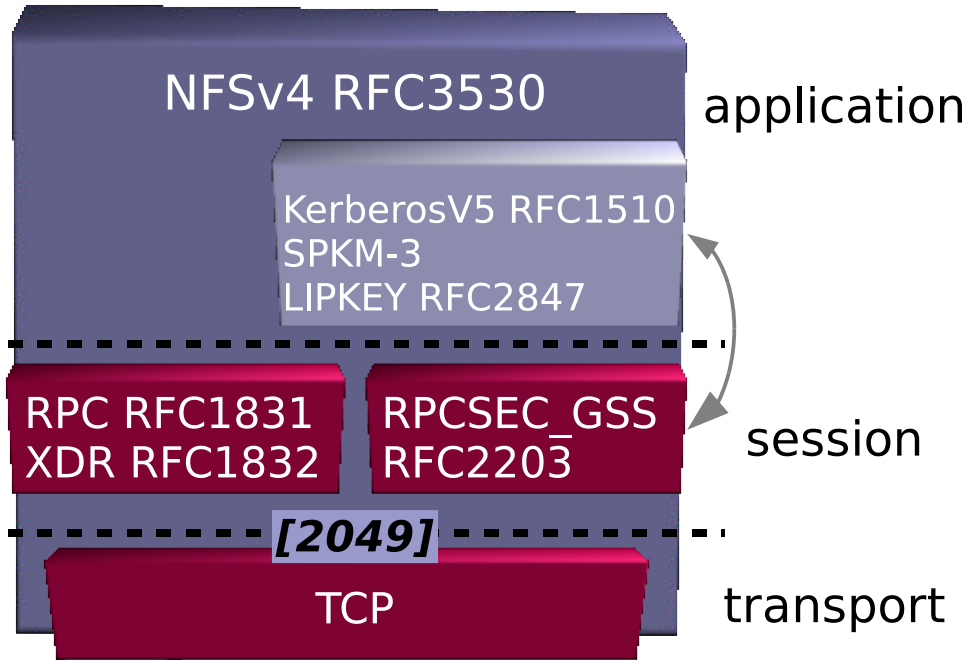
Network File System (NFS)

- Implémentation des systèmes de fichiers



Network File System (NFS)

- Pile de protocoles NFS v4

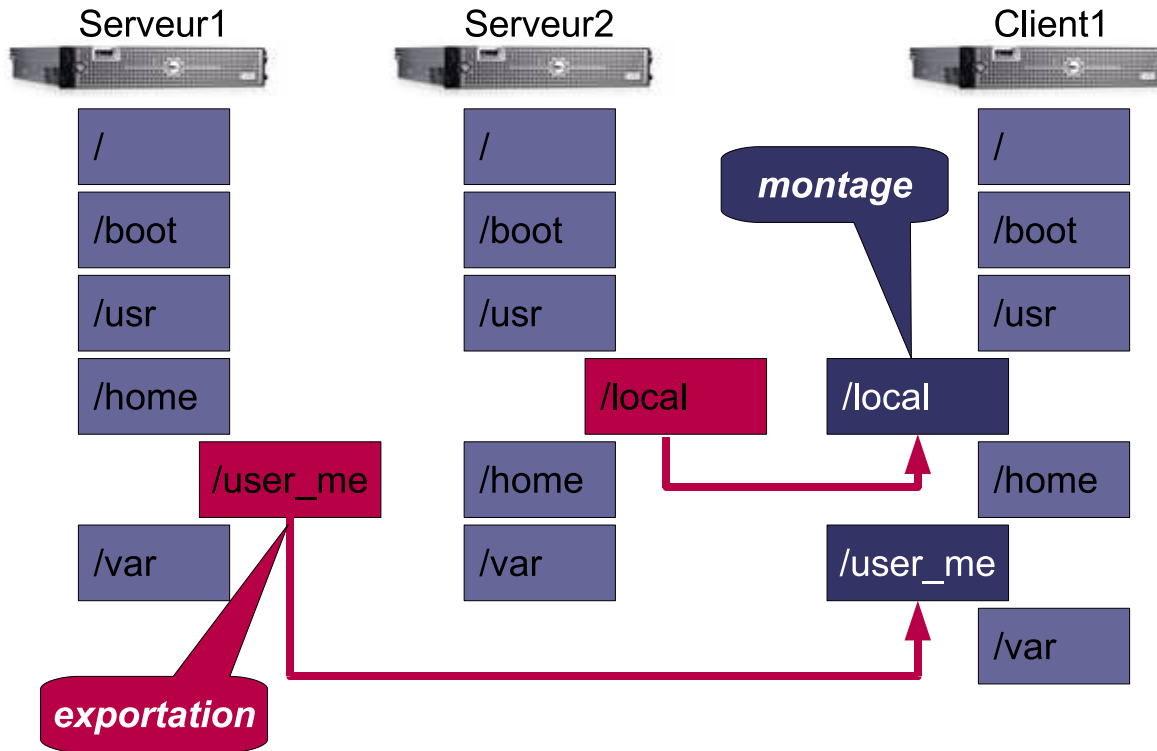


Network File System (NFS)

- Service sans suivi de session
 - Chaque appel RPC est indépendant
 - Pas de trace des appels précédents sur les serveurs
 - Traitement amélioré des ruptures de session
 - ▶ Aucun état à récupérer
 - ▶ Lancement de nouveaux appels côté client
- Transparence du stockage
 - Arborescence distante partagée
 - **Exportée** par le serveur
 - **Montée** par le client

Network File System (NFS)

- Arborescence du système de fichiers



Network File System (NFS)

- Service de montage

- Commande `mount`

- ▶ Montage d'un système de fichiers distant dans l'espace local du client

- Côté client

- Gestionnaire de verrous `nlockmgr`

- Côté serveur

- Gestionnaire de verrous `nlockmgr`

- Gestionnaire de montage `mountd`

- Contrôle d'accès

- ▶ Droits sur l'arborescence exportée

Network File System (NFS)

- Exemple d'utilisation du service

- Côté serveur : exportation du répertoire /var/exports

```
[192.168.1.1]:~# exportfs
```

```
/var/exports 192.168.1.4
```

^

^

| '----- Désignation du client

'----- Répertoire exporté

- Côté client : montage vers le répertoire /mnt

```
[192.168.1.4]:~# mount -t nfs -o nfsvers=3 \
```

```
192.168.1.1:/var/exports /mnt
```

^

^

| '----- Répertoire

| local client

'----- Désignation du serveur

```
[192.168.1.4]:~# ls /mnt
```

```
[192.168.1.4]:~# umount /mnt
```

Network File System (NFS)

- Analyse des questions/réponses RPC

192.168.1.4 -> 192.168.1.1 Portmap V2 DUMP Call

192.168.1.1 -> 192.168.1.4 Portmap V2 DUMP Reply (Call In 6)

192.168.1.4 -> 192.168.1.1 MOUNT V3 MNT Call <<- appel commande mount

192.168.1.1 -> 192.168.1.4 MOUNT V3 MNT Reply (Call In 14)

192.168.1.4 -> 192.168.1.1 Portmap V2 GETPORT Call

192.168.1.1 -> 192.168.1.4 Portmap V2 GETPORT Reply (Call In 17)

192.168.1.4 -> 192.168.1.1 NFS V3 FSINFO Call, FH:0xc4480c84 <<- appel commande ls

192.168.1.1 -> 192.168.1.4 NFS V3 FSINFO Reply (Call In 19)

192.168.1.4 -> 192.168.1.1 NFS V3 GETATTR Call, FH:0xc4480c84

192.168.1.1 -> 192.168.1.4 NFS V3 GETATTR Reply (Call In 21)

192.168.1.4 -> 192.168.1.1 NFS V3 ACCESS Call, FH:0xc4480c84

192.168.1.1 -> 192.168.1.4 NFS V3 ACCESS Reply (Call In 25)

192.168.1.4 -> 192.168.1.1 NFS V3 REaddirPLUS Call, FH:0xc4480c84

192.168.1.1 -> 192.168.1.4 NFS V3 REaddirPLUS Reply (Call In 27)

192.168.1.4 -> 192.168.1.1 Portmap V2 GETPORT Call

192.168.1.1 -> 192.168.1.4 Portmap V2 GETPORT Reply (Call In 29)

192.168.1.4 -> 192.168.1.1 MOUNT V1 UMNT Call <<- appel commande umount

192.168.1.1 -> 192.168.1.4 MOUNT V1 UMNT Reply (Call In 31)

Network File System (NFS)

- Quid des accès transparents ?

- Très Bien !

- ▶ Système de fichiers distant monté vu comme système local

- Bien !

- ▶ Accès unique au système de nommage pour le client

- Moins bien !

- ▶ Pas de système de nommage uniforme côté client

- Pas bien !

- ▶ Montage «manuel» obligatoire avant tout accès distant

- Amélioration : montage automatisé

- Paquets **autofs*** et commande **automount**

- ▶ Montage dynamique à la demande des systèmes de fichiers distants
- ▶ Table de correspondance entre les points de montage et les serveurs
- ▶ Forme basique de réplication ... en lecture

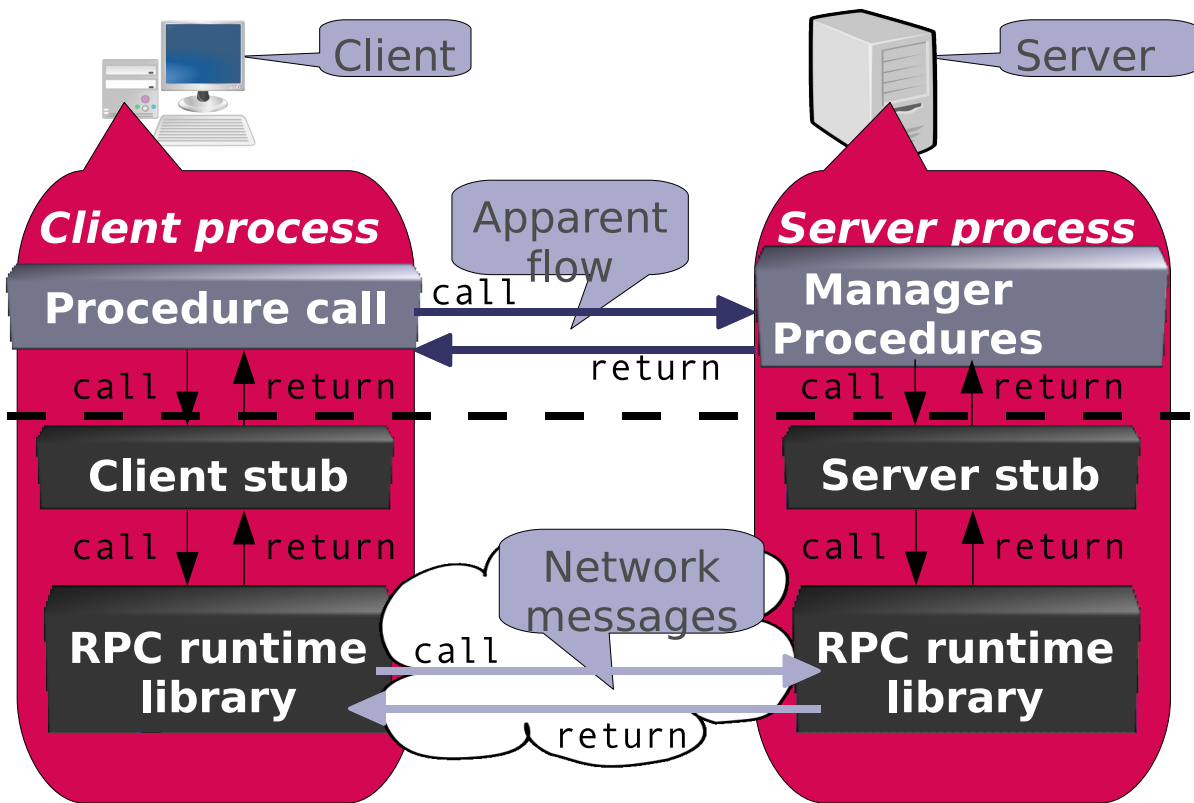
Remote Procedure Calls (RPC)

- Principe

- Étendre les appels de procédures locaux
 - ▶ Contrôler et transférer de données sur un réseau
 - ▶ Appels de procédures analogues aux appels locaux
 - ▶ Appels exécutés par un processus différent sur une machine distante
- Technique bien adaptée au modèle client-serveur
- Séquence des opérations
 - ▶ Appel RPC
 - ▶ Gel du processus appelant
 - ▶ Passage des paramètres à la machine distante
 - ▶ Exécution de la procédure sur la machine distante
 - ▶ Renvoi des paramètres à la machine appelante
 - ▶ Reprise de l'exécution du processus appelant

Remote Procedure Calls (RPC)

- Implémentation des RPCs

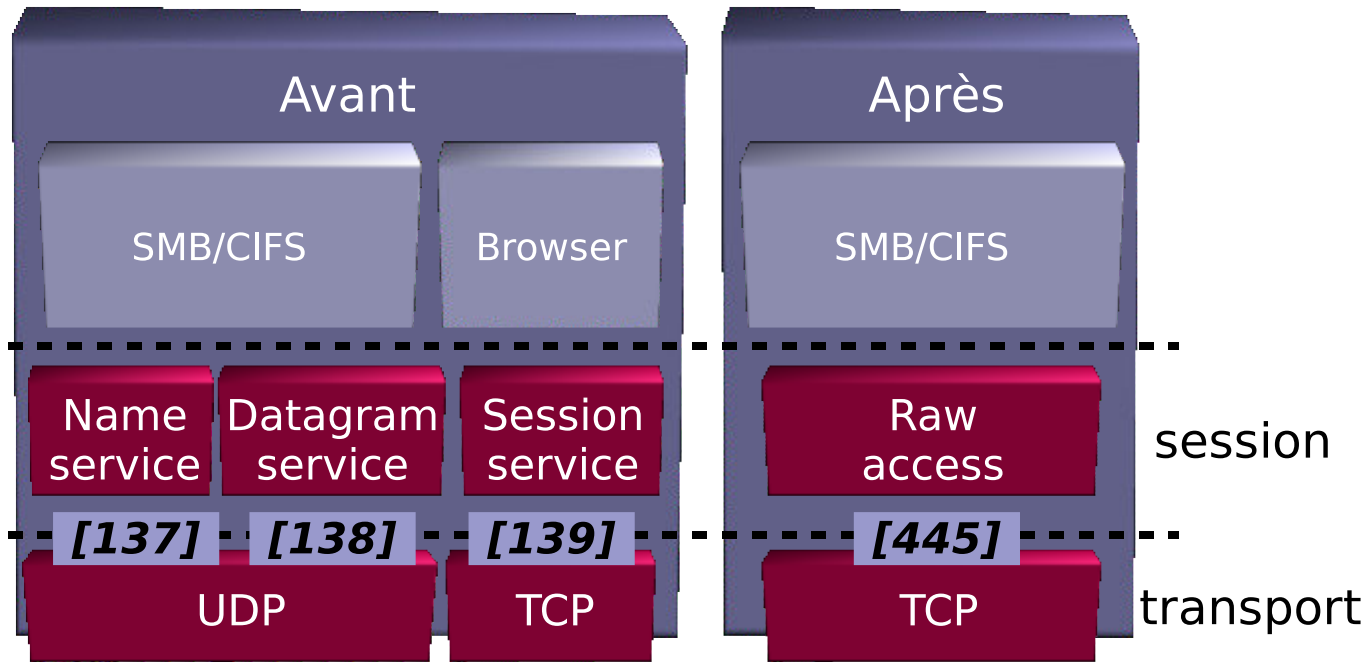


Common Internet File System (CIFS)

- CIFS = évolution de SMB
 - SMB ou Server Message Block
 - ▶ Développé au début des années 80 par IBM
 - ▶ Conçu pour utiliser l'API NetBIOS à l'intérieur d'un LAN
 - ▶ Étendu par Microsoft lors du développement de Windows 2000
 - ▶ Compatibilité NetBIOS maintenue dans Windows 2000
 - Fonctionnalités CIFS
 - ▶ Non limitées aux fichiers
 - ▶ Nommage Global : identification unique côté serveur ou client
 - ▶ Contrôles d'accès étendus ACLs
 - ACLs sur un fichier
 - ▶ Discretionary ACL : ALLOW/DENY Access Control Entries
 - ▶ Security ACL : AUDIT/ALARM Access Control Entries

Common Internet File System (CIFS)

- Pile de protocoles



Common Internet File System (CIFS)

- Exemple d'utilisation du service

```
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
```

Common Internet File System (CIFS)

- Connexion TCP et négociations

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] ... <- authentication OK
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

Common Internet File System (CIFS)

- Appel 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] ...

Common Internet File System (CIFS)

- Appel service et libération connexion TCP

```
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] ...
```

Synthèse NFS vs. CIFS

● CIFS

■ Avantages côté client

- ▶ Base installée très importante
- ▶ Granularité des ACLs
- ▶ Offre de services

■ Inconvénients côté serveur

- ▶ Protocole trop bavard pour les réseaux étendus
- ▶ Trop sensible aux ruptures de connexions

● NFS

■ Inconvénients côté client

- ▶ Développement long et lent
- ▶ Utilisation intensive de Kerberos
- ▶ Gestion des ACLs Unix != Windows

■ Avantages côté serveur

- ▶ Réplication fiable
- ▶ Adapté aux réseaux étendus