Virtual Box

- Définition, usages et classification
- VirtualBox
- Labo Jour 1 : §1 LiveCD / §2 Appliance / §3 Réseau (NAT-Bridge)
- tcpdump – WinPcap
- Labo Jour 2 : §5 Client – Server / §6 Virtual Switch
- Virtual Private Network
- Labo Jour 3 : §8 Virtual Router / §9 Virtual Firewall

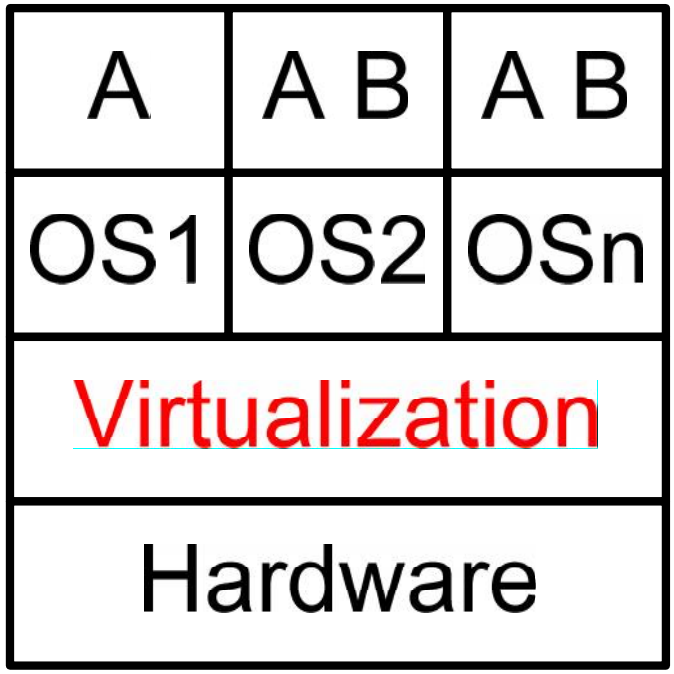
Transparent versus virtual

- ***Something is transparent when it is physically here but seems not to be***
  - → commutateur ethernet, routeur, …
  - → l'équipement est présent physiquement
  - → il est correctement configuré et devient transparent pour les utilisateurs

- ***Something is virtual when it is not physically but seems to be***
  - → disque virtuel (disque émulé en RAM)
  - → mémoire virtuelle de 2^{32} bytes (4 GByte) de Windows 32 bit
  - → émulation du BIOS (Basic Input/Output System)
  - → grâce à la virtualisation (émulation), l'utilisateur peut exécuter du code ARM sur un processeur x86

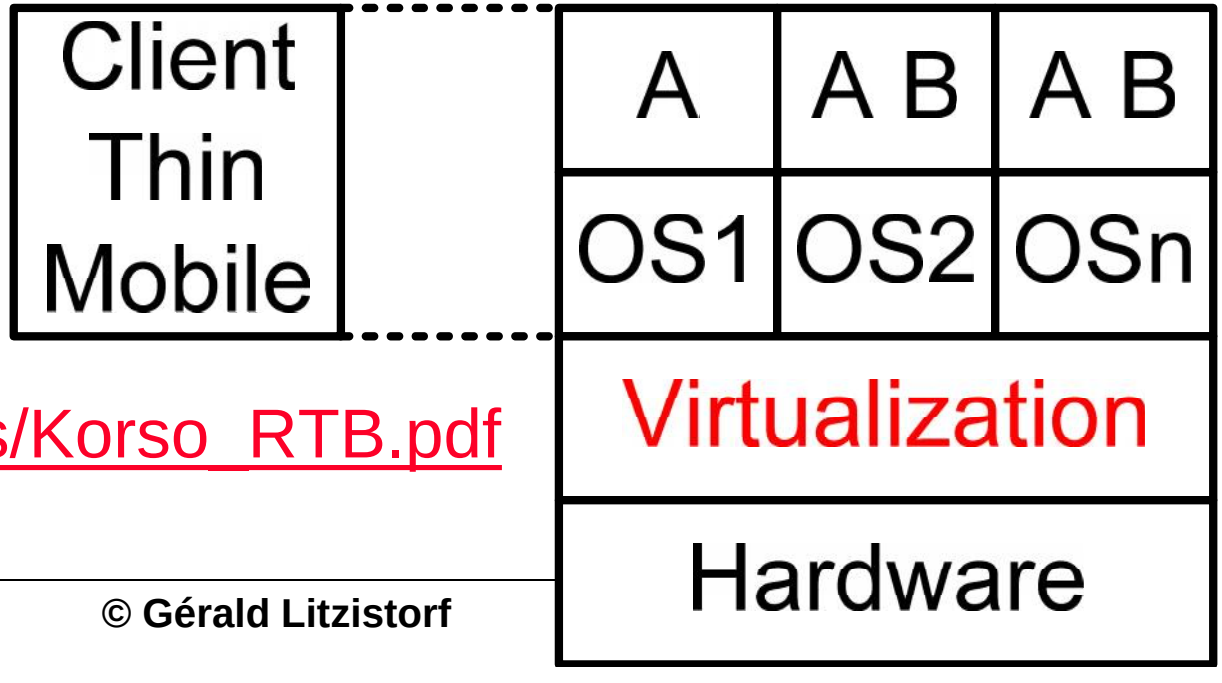
Usages de la virtualisation (1)

- Consolider des serveurs physiques
  - Des systèmes d'exploitation (OS)
  - Partage des ressources matérielles
  - Couche de virtualisation = hyperviseurs
  - VMware ESXi ou Linux KVM
  - http://www.tdeig.ch/Schema_Reseau



- Virtual Desktop Infrastructure
  - Clients légers
  - Administration centralisée

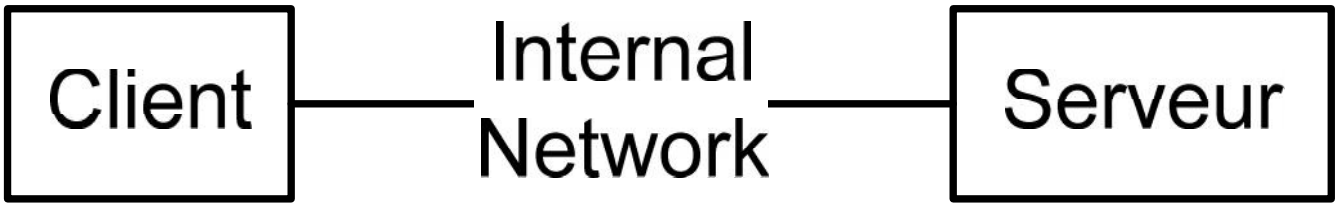
  http://www.tdeig.ch/windows/Korso_RTB.pdf



Usages de la virtualisation (2)

- Virtual Network

  Labo



- Virtualisation du stockage

  Système garantissant par exemple que chaque fichier est stocké à 3 endroits parmi les 5 possibles → VMware Virtual SAN

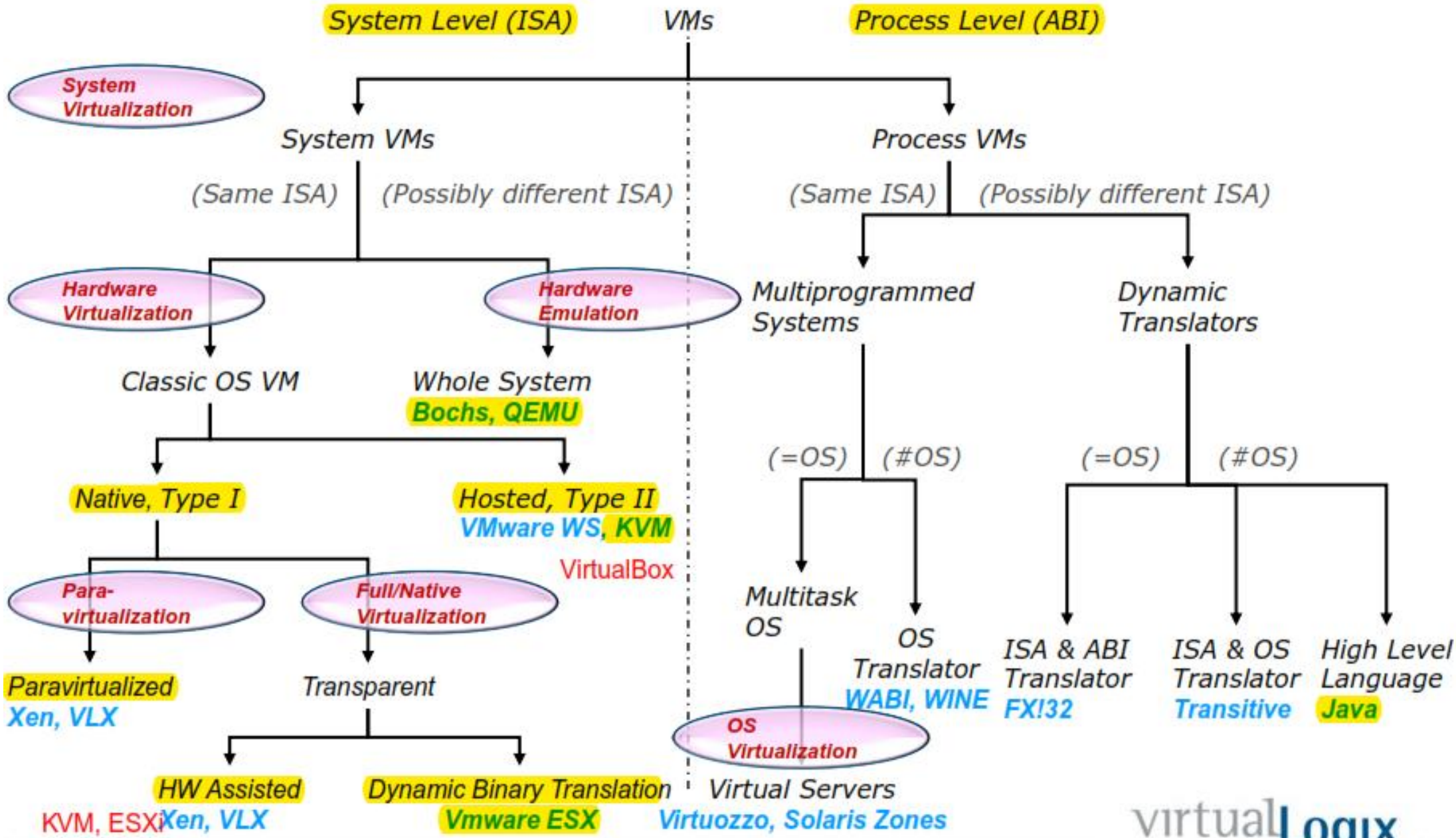
- Virtualisation de l'espace mémoire

  Chaque application croit disposer d'une espace de 2^{64} bit alors que la mémoire physique = 16 GByte

- Beaucoup d'autres

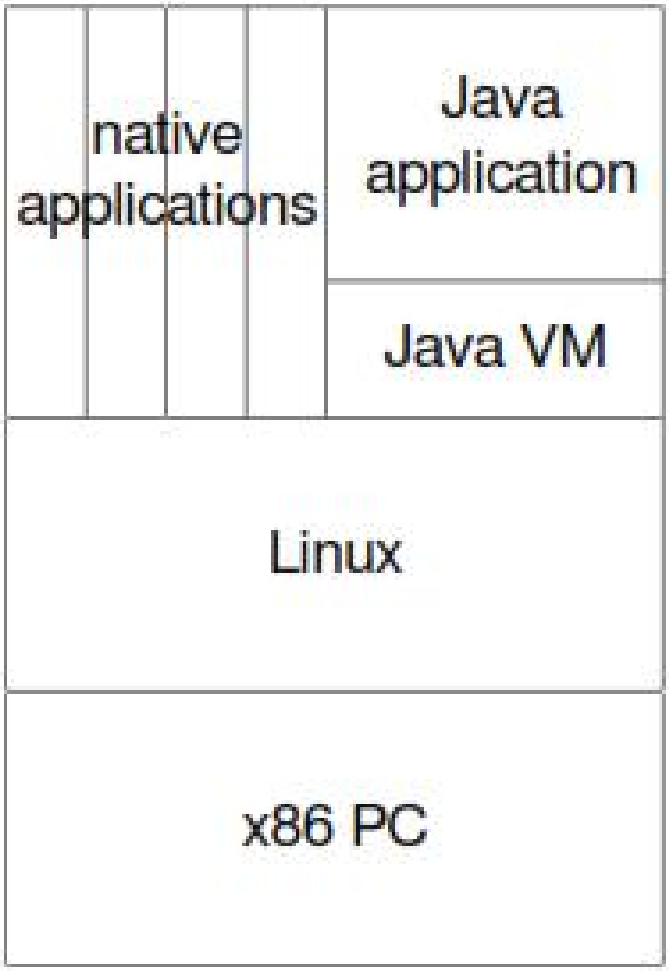
Taxonomy (derived from E. Smith & Nair)

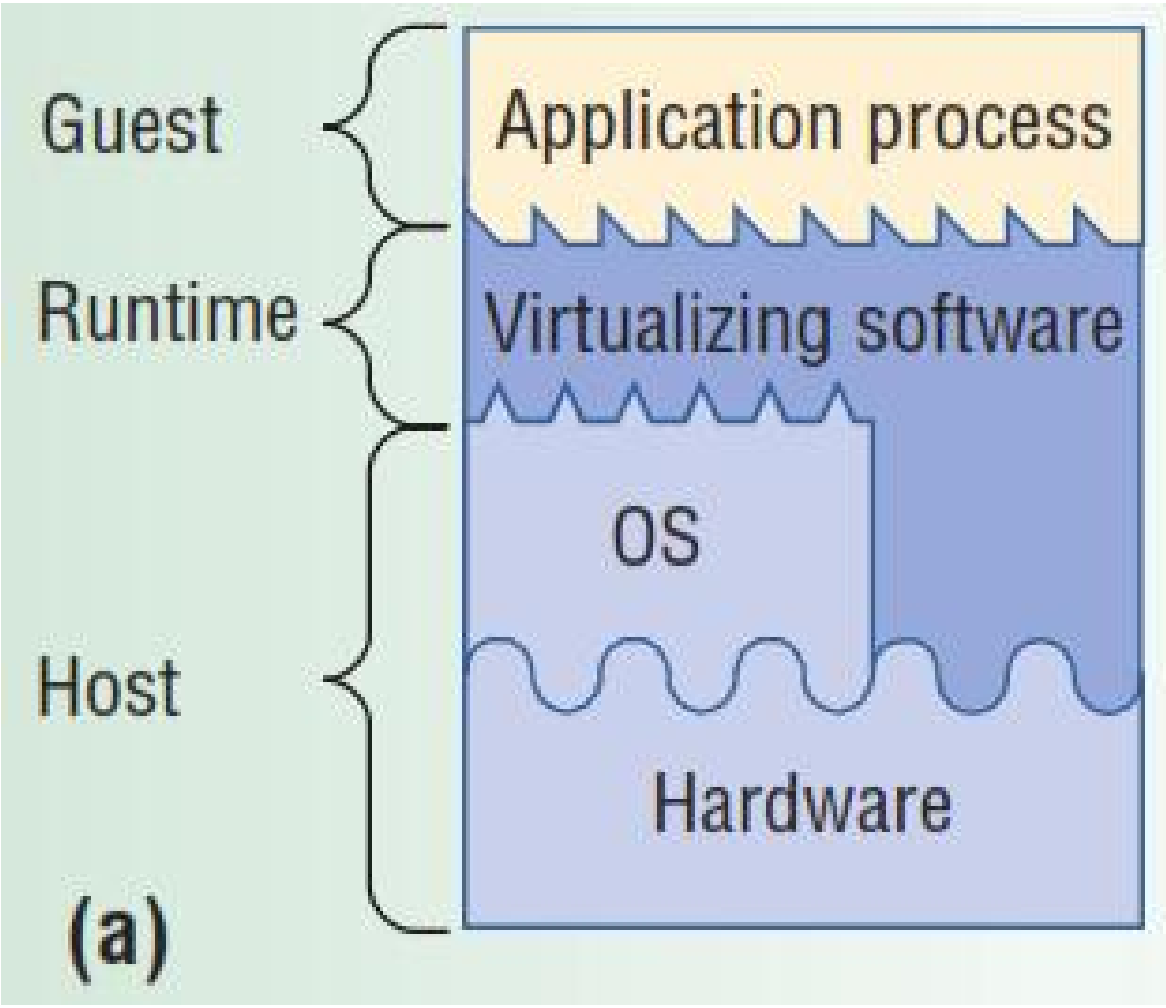
Annexes
→ tdeig



virtualLogix

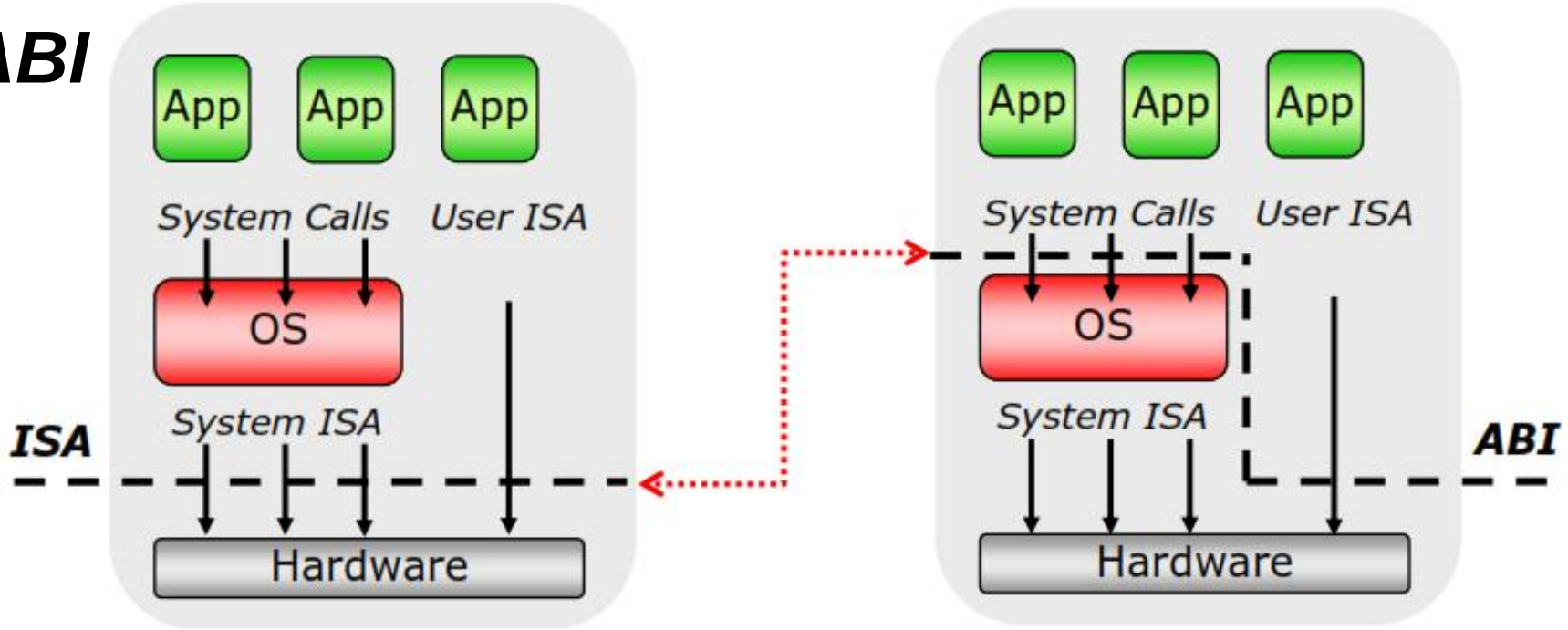
Java





- Java VM environment **interprets** Java bytecode programs and perform I/O operations through Java libraries
- This system supports native x86 applications that are **compiled** for a Linux system

ISA versus ABI

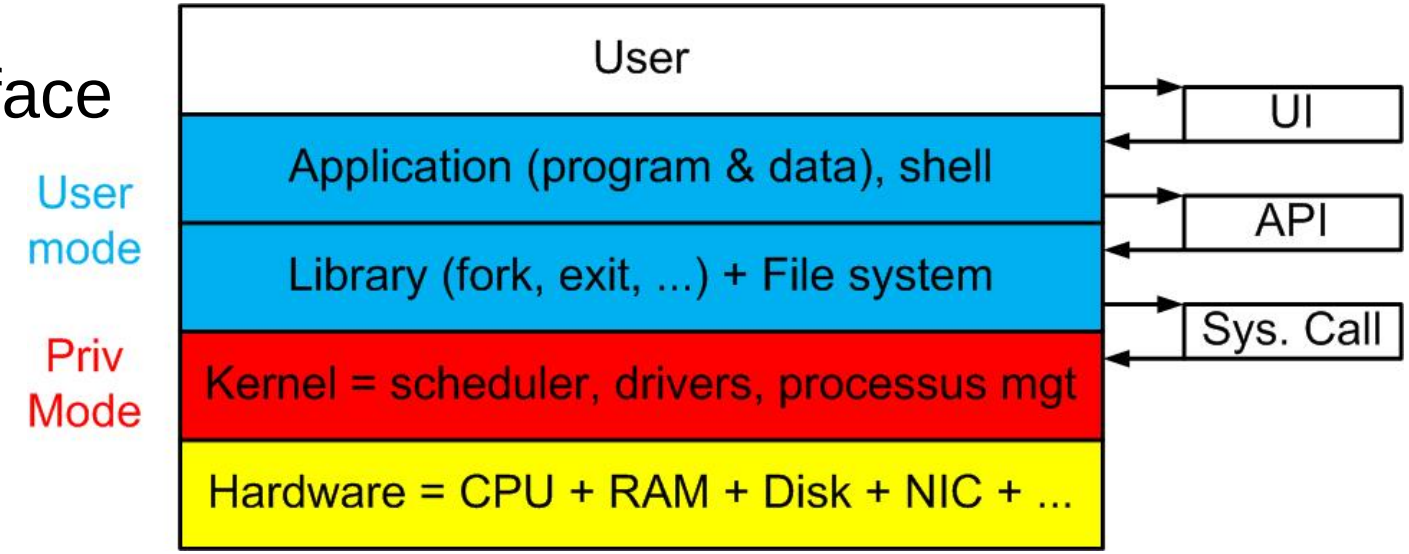


- Interface **ISA** (Instruction Set Architecture) sépare matériel du logiciel

  Exemple = ISA x86

- Application Binary Interface (**ABI**) donne un accès bas niveau (system calls) aux ressources

  Exemple = Linux



Taxonomy (suite)

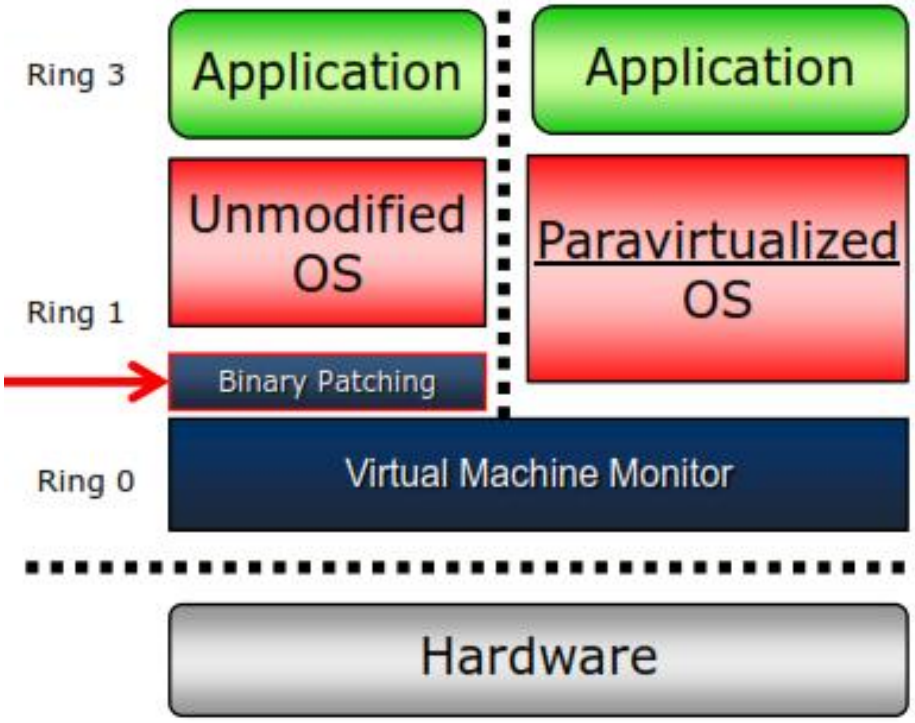
Hardware Virtualization

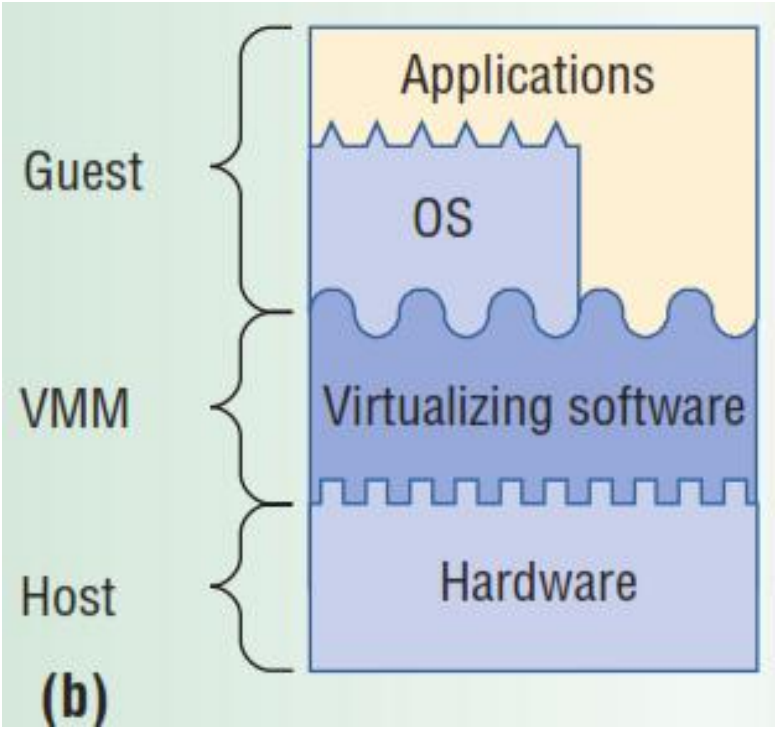
- Perfect emulation of all chips, controllers, … of the underlying physical machine
- Same operating system(s) and application software can run either on top the VM or on top the physical machine
- Example = VM370 (IBM 1972)

Hardware Emulation

- Emulation of a machine that we do not have
- Must emulate **all** instructions
- Example = QEMU can run ARM code on your x86-PC
- Example = GNS3 can run Cisco binaries

Taxonomy (suite)





Unmodified OS

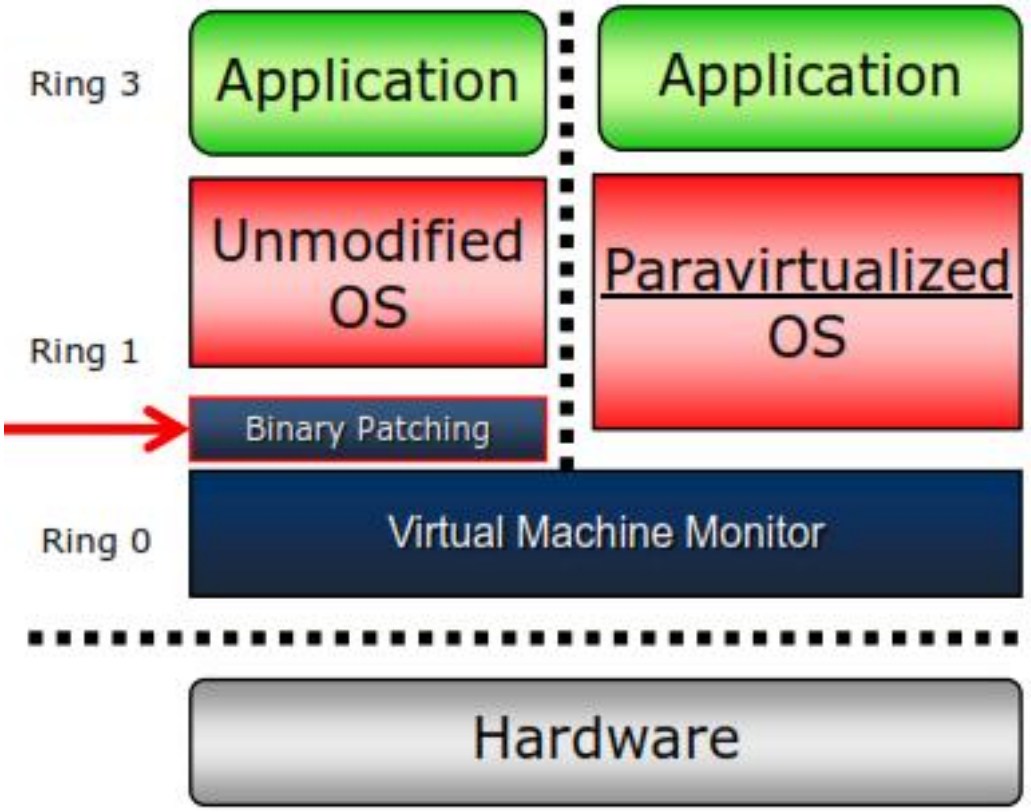
- Windows + Drivers + Software Tools → VMware

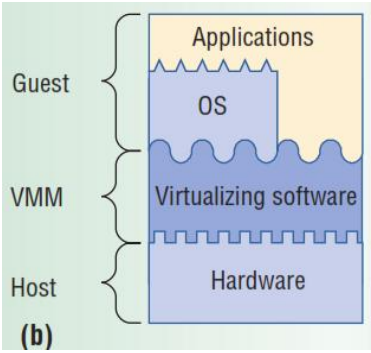
Para Virtualization

- Virtio in OS (Linux) and VirtualBox ou KVM

Taxonomy (suite)

Binary Translation



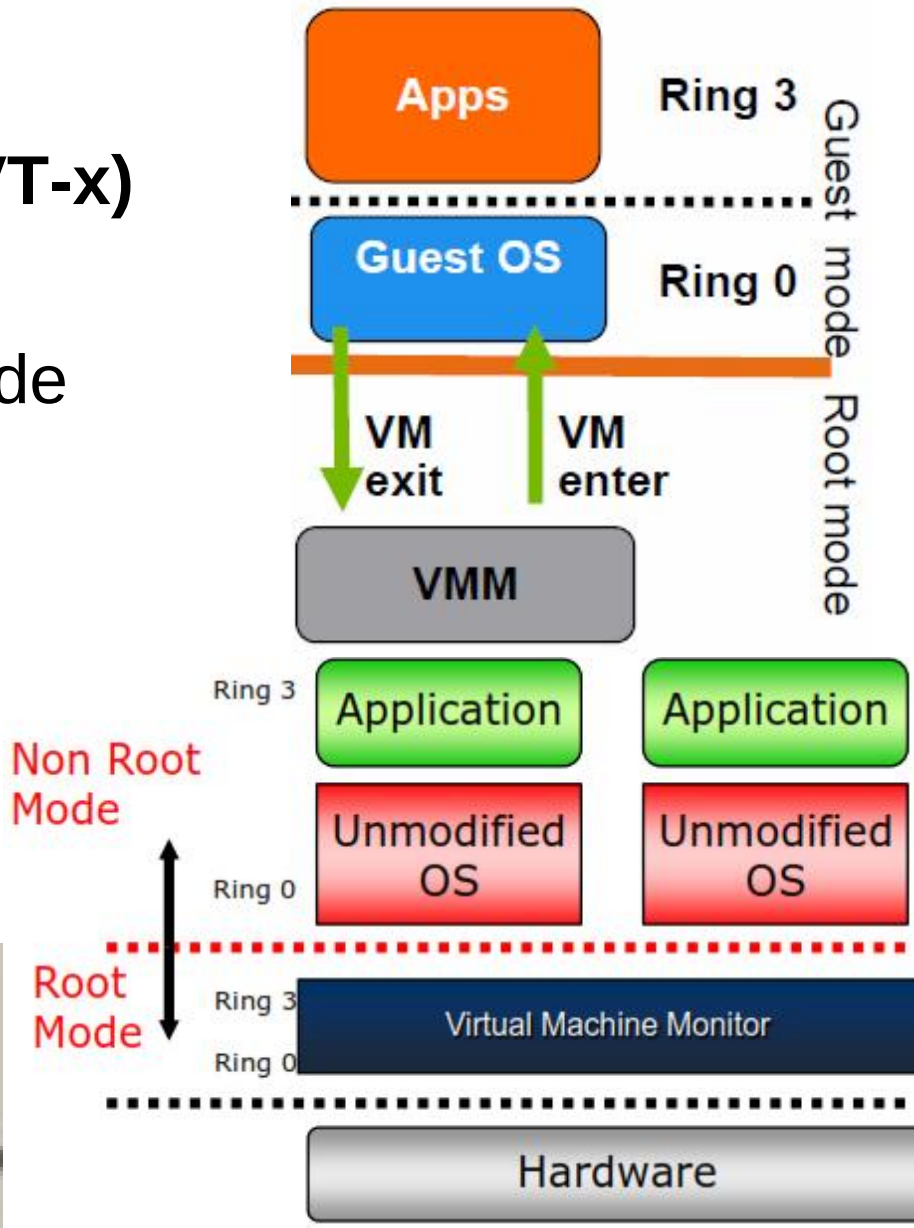


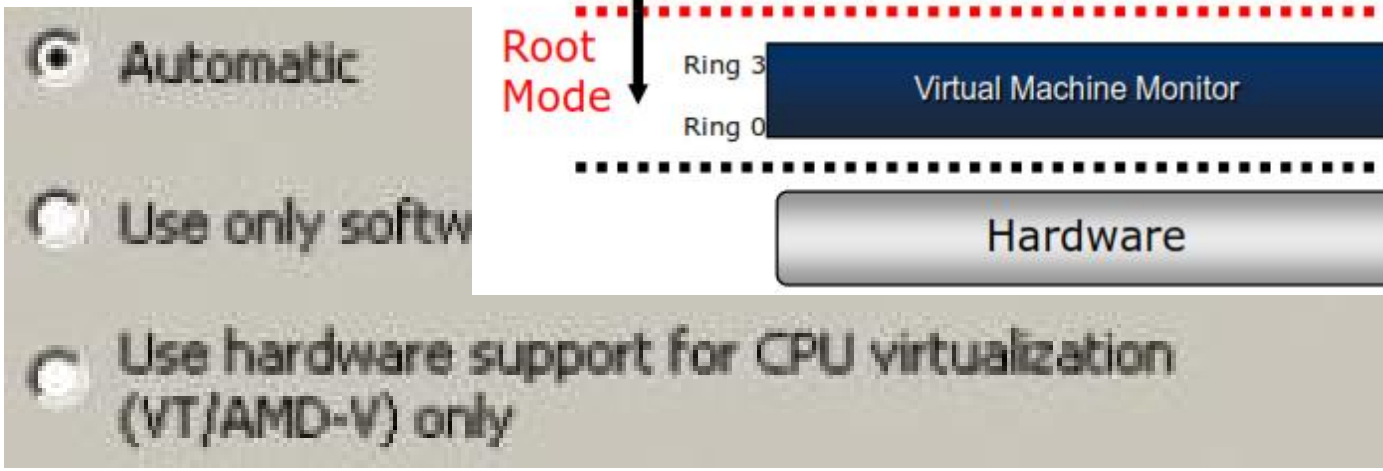
- Technique for the efficient emulation of instruction sequences
- Code Scanning, Code Rewriting, Just In Time compilation
- Most instructions are copied directly
- Non-virtualizable (dangerous) instructions are modified
- Example = VMware ESX 2001 (without hardware assist)

Taxonomy (suite)

Hardware-assisted Virtualization (Intel VT-x)

- VMM executes in Root mode
- Guest (OS, Apps) execute in Guest mode
- VM Entry instruction (VMM → Guest)
- VM Exit instruction (Guest → VMM)
- Hard-assist virtualization was added to x86 CPU (Intel VT-x or AMD-V) in 2006
- Advantages : performance & security
- ESXi can combine → BT & Hardware assist
- Linux-KVM requires Hardware assist.

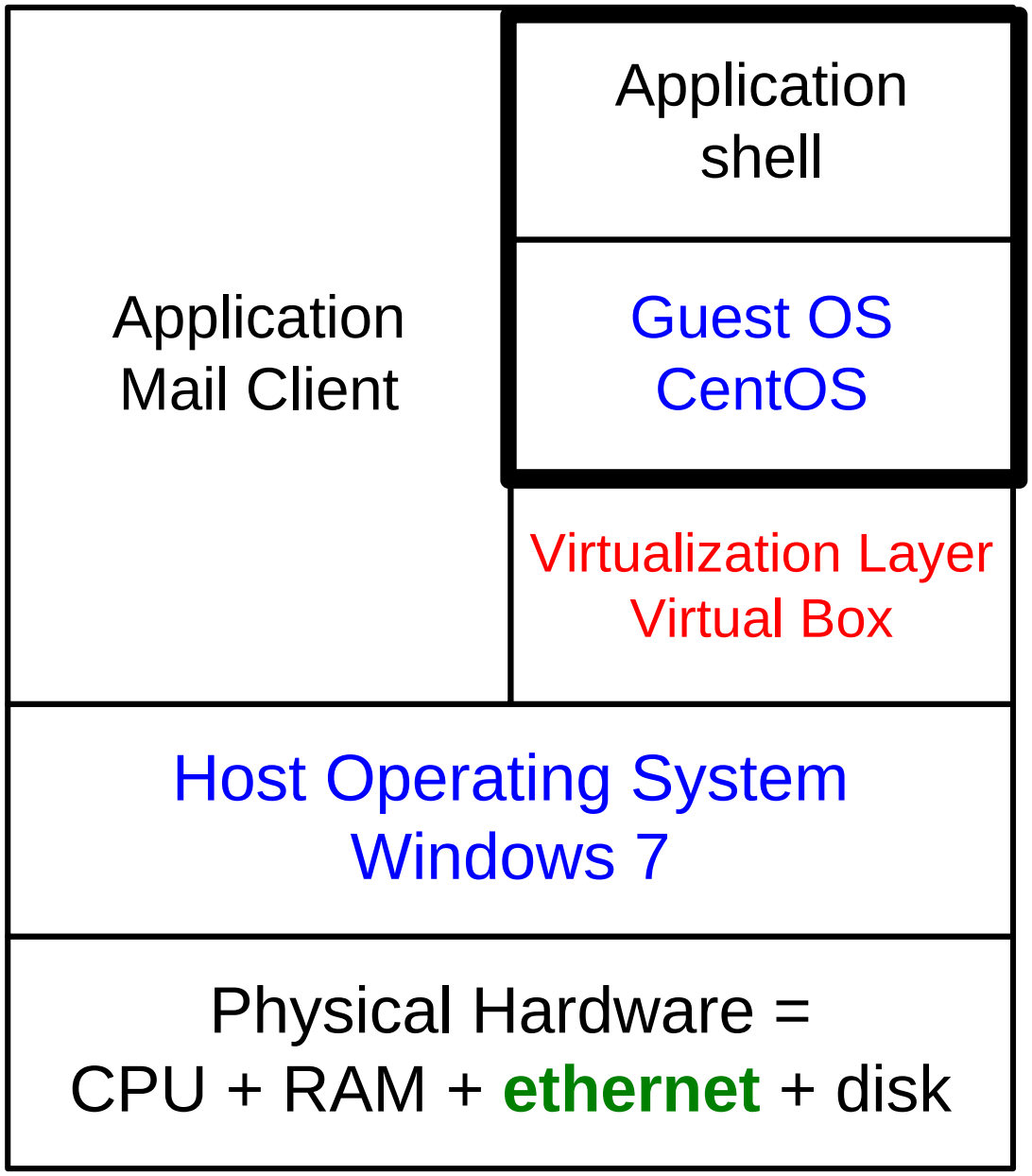




VirtualBox (Oracle)

- Vbox is a **cross-platform** virtualization application
- Vbox runs on **Windows, Linux, Mac, and Solaris hosts**
- Vbox supports a large number of **guest operating systems** including Windows (…, XP, Server 2003, Vista, Windows 7, …), Linux (2.4 and 2.6), OpenBSD, …
- https://www.virtualbox.org/
- Manuel
- http://fr.wikipedia.org/wiki/Oracle_VM_VirtualBox

VirtualBox : Architecture



- Excellent support du matériel grâce à Host OS
- Choix des Guest OS grâce à VirtualBox (Vbox)
- **VM** (machine virtuelle) contient Guest OS + applications
- On parle d'une architecture de Type 2 (sans hyperviseur)

Fonctionnement (Manual §10.4 – 10.5)

- Implementing virtualization on x86 CPUs with no hardware virtualization support is an **extraordinarily complex task** because the CPU architecture was not designed to be virtualized.
- Vbox remplace les pilotes (**drivers**) disque, réseau, … par les siens **afin d'émuler la couche physique**
- Vbox contains a Code Scanning and Analysis Manager (CSAM), which disassembles guest code, and the Patch Manager (PATM), which can replace it at runtime.
- Since 2006, **Intel VT-x** (and AMD) processors have had support for so-called "**hardware virtualization**".This means that these processors can help Vbox to intercept potentially dangerous operations that a guest operating system may be attempting and also makes it easier to present virtual hardware to a virtual machine.
- http://www.tdeig.ch/vmware/Kaegi.pdf

Labo §1 : Utiliser CentOS-6.4-x86_64-minimal.iso (1/2)

- Emulation (Manuel p14)

  … if you provide VirtualBox with the image of a CD-ROM in an ISO file, VirtualBox can present this image to a guest system as if it were a physical CD-ROM

- §1.1 Créer une nouvelle VM avec les paramètres par défaut
- §1.2 Fichiers XML créés
- §1.3 Démarrer sur le CD virtuel

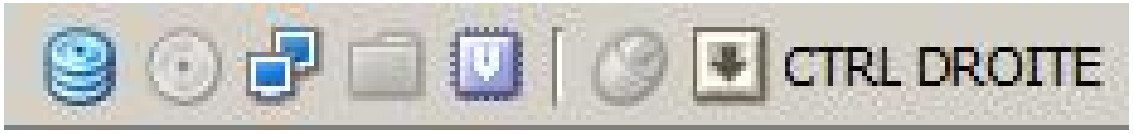
  BIOS emulation

- §1.4 Parcourir les icônes

- Fermer la fenêtre pour **stopper le chargement** (et gagner du temps)

Labo §1 : Utiliser CentOS-6.4-x86_64-minimal.iso (2/2)

- §1.4 Parcourir les icônes

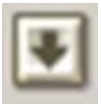



Virtual disk


CD/DVD


Network interface


Mouse


Keyboard

CTRL DROITE to switch → Important

Labo §2 : Utiliser une appliance

- Manuel p31

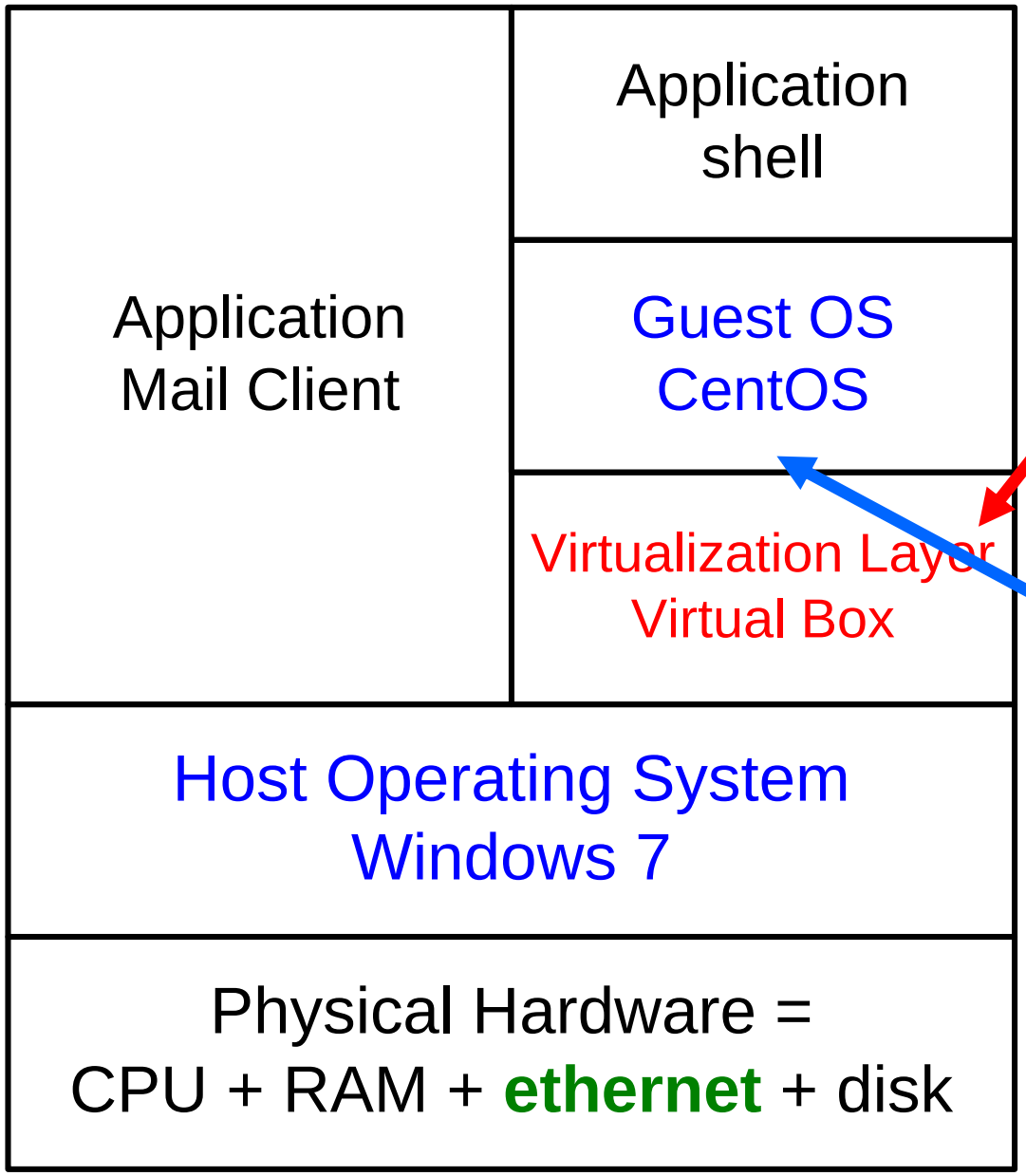
  OVF (Open Virtualization Format) is a cross-platform standard supported by many virtualization products (VirtualBox, …) which allows for creating ready-made **VMs** (virtual machines)

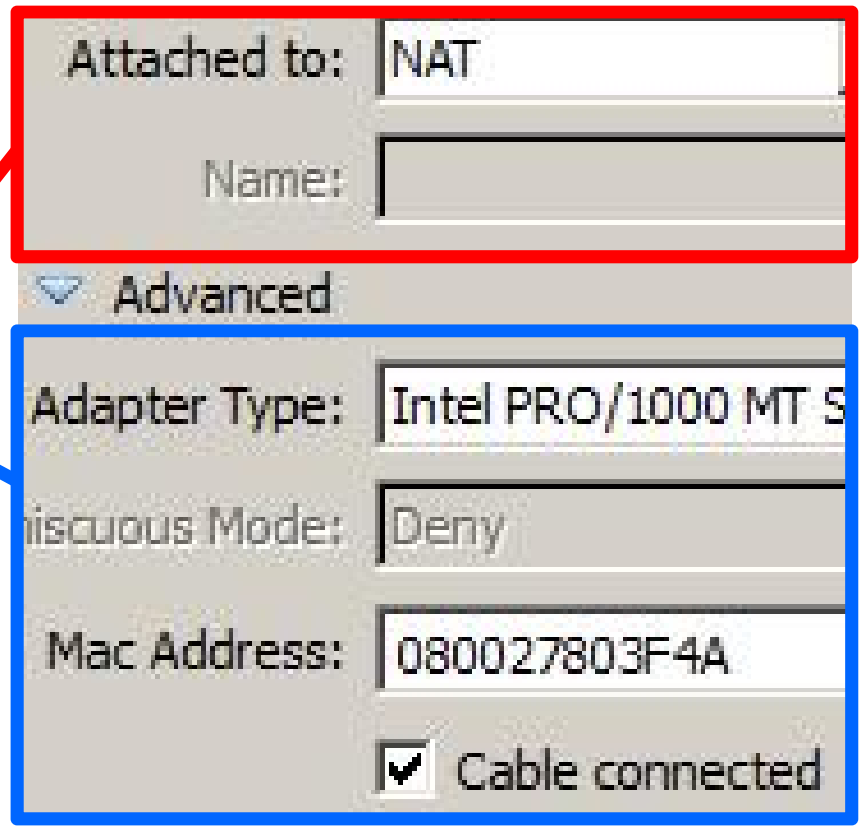
  VirtualBox makes OVF **import** and **export** easy …

  This allows for packaging so-called **virtual appliances**: disk images together with configuration settings that can be distributed easily

- §2.1 Importer une appliance (ova)
- §2.2 Retrouver les informations utiles
- §2.3 Déterminer le type de périphérique émulé

Labo §3.1 : NAT networking (default mode)

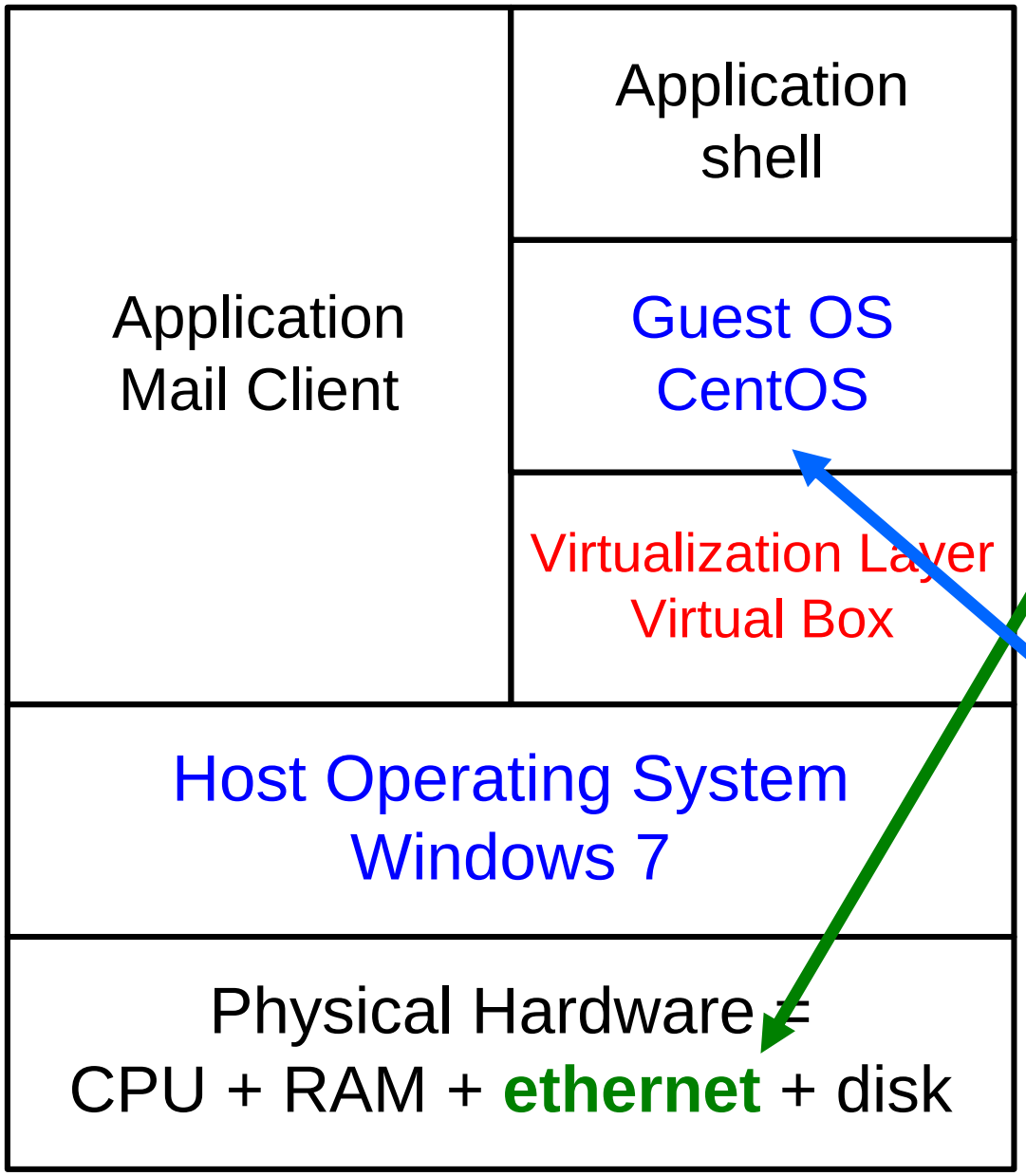


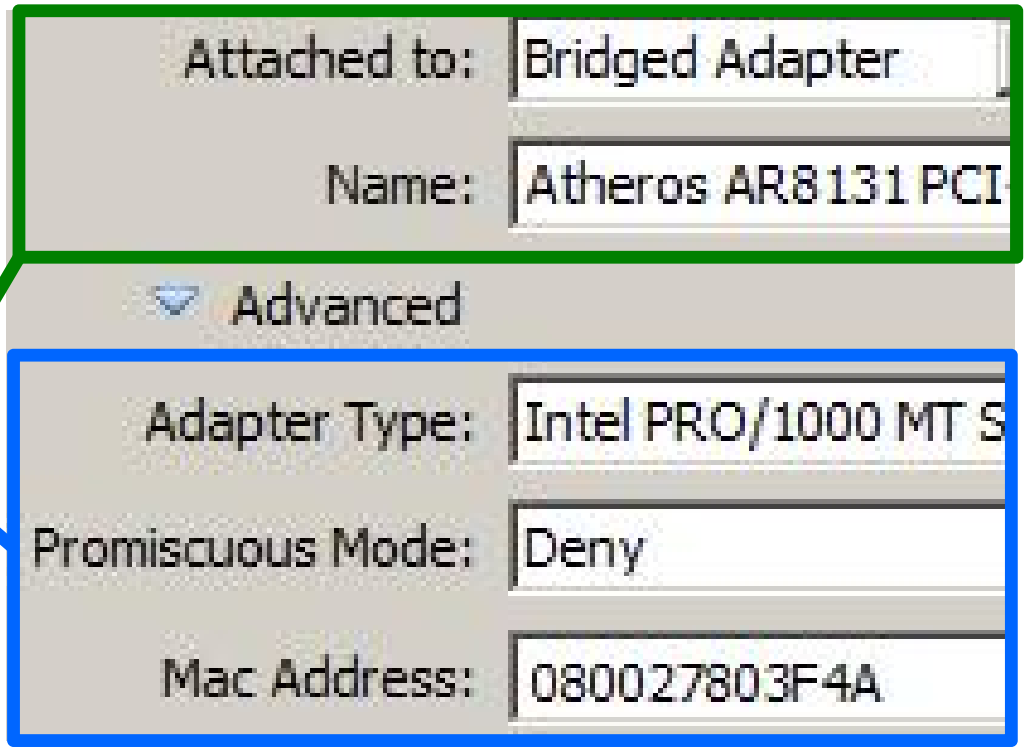


Réseau émulé par Vbox

Ex 1 : Détailler le réseau émulé par Vbox

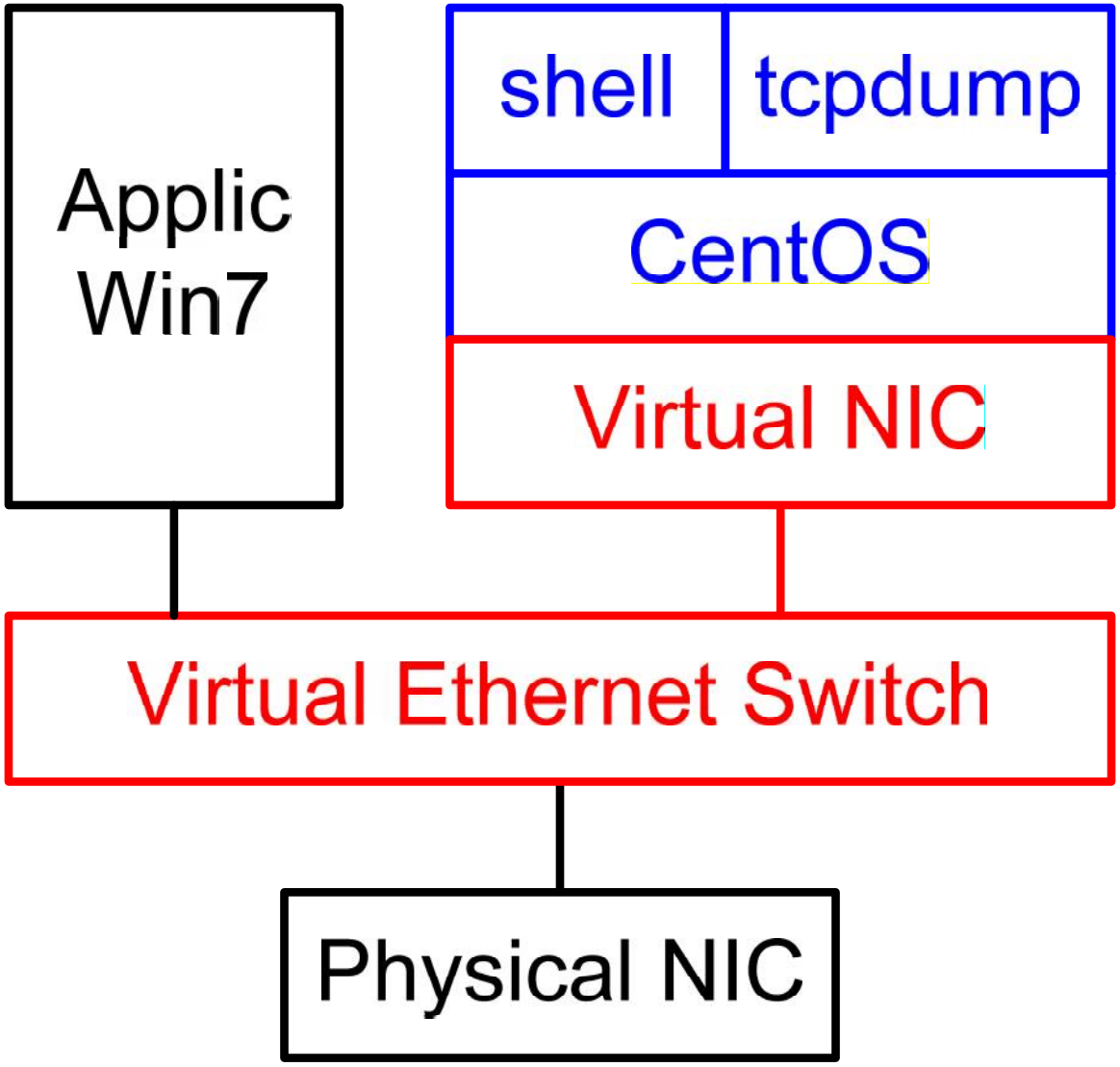
Labo §3.2 : Bridged networking





Generates a new random MAC address

Labo §3.2 : Bridged networking (modèle en couches)



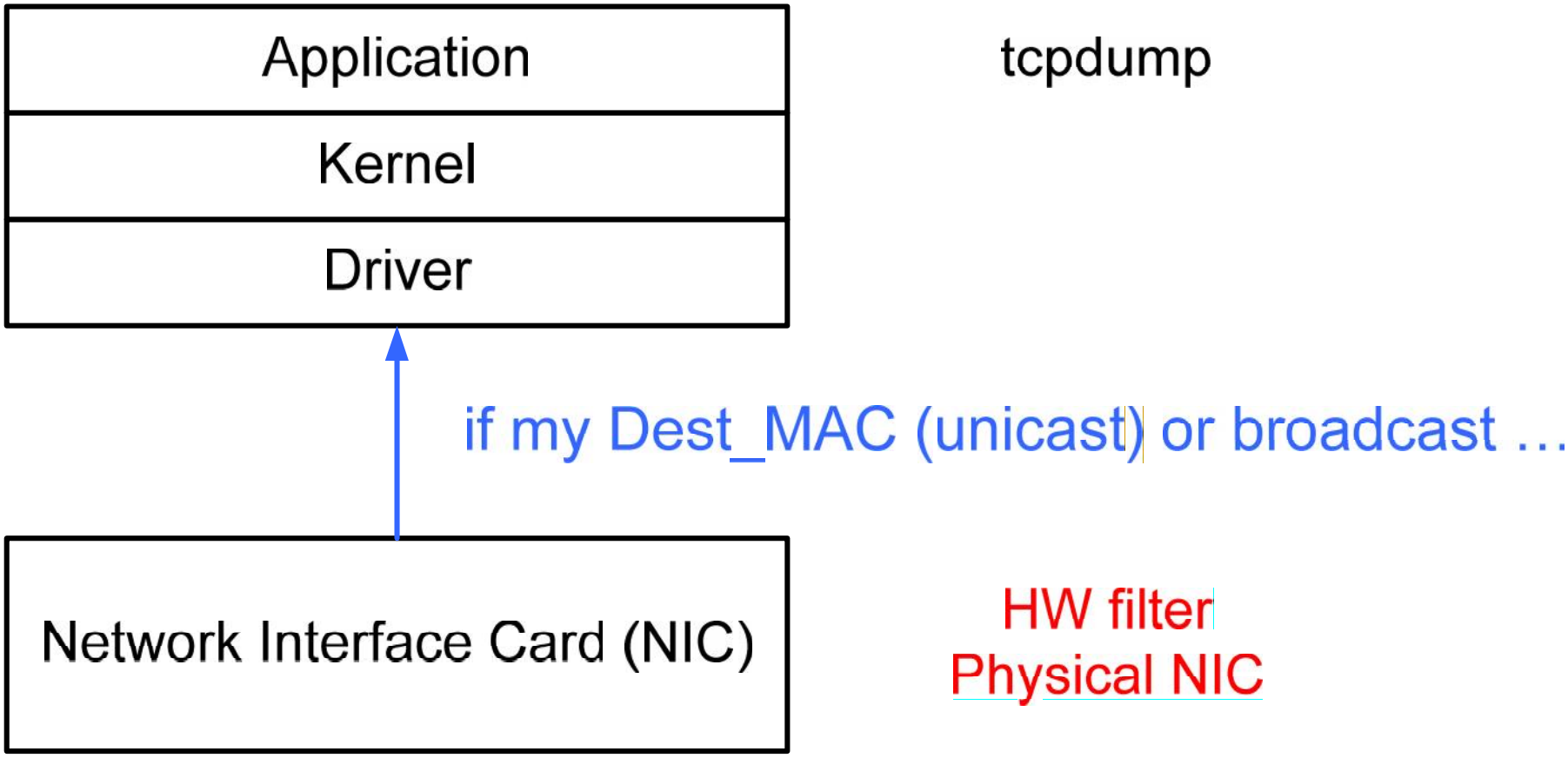
- Utilisons l'outil tcpdump (Wireshark) pour comprendre le rôle de WinPcap utilisé dans l'architecture Vbox

Network Monitoring with tcpdump (CLI)

- **`tcpdump –i eth0`** all traffic
- **`... icmp`** ICMP only
- **`... [arp/icmp/ip/udp/tcp]`** protocol only
- **`... host 192.168.1.1`** all traffic to & from this IP host
- **`... src host 192.168.1.1`** Source IP only
- **`... dst host 192.168.1.1`** Destination IP only
- **`... [src/dst] port 80`** http traffic
- **`... ((tcp) and (port 80) and ((dst host ...) or (dst host ...)))`**
- **`... ((icmp) and (ether dst host 00:01:02:03:04:05))`**
- **`... ((tcp) and (not host ... 192.168.1.200))`**

- http://www.tcpdump.org/
- http://www.wains.be/pub/networking/tcpdump_advanced_filters.txt

Hardware Receive Frame



- Ex 2 : déterminer **l'algorithme implémenté au niveau matériel** de réception d'une trame. **Pas de virtualisation.**

Synthèse du labo Vbox 1 (1/2)

- **Quels sont les principaux paramètres d'une machine virtuelle ?**

  Nb de vCPU, RAM size, Réseau (nb vNIC, NAT/Bridge), …

- **Où sont-ils mémorisés ?**

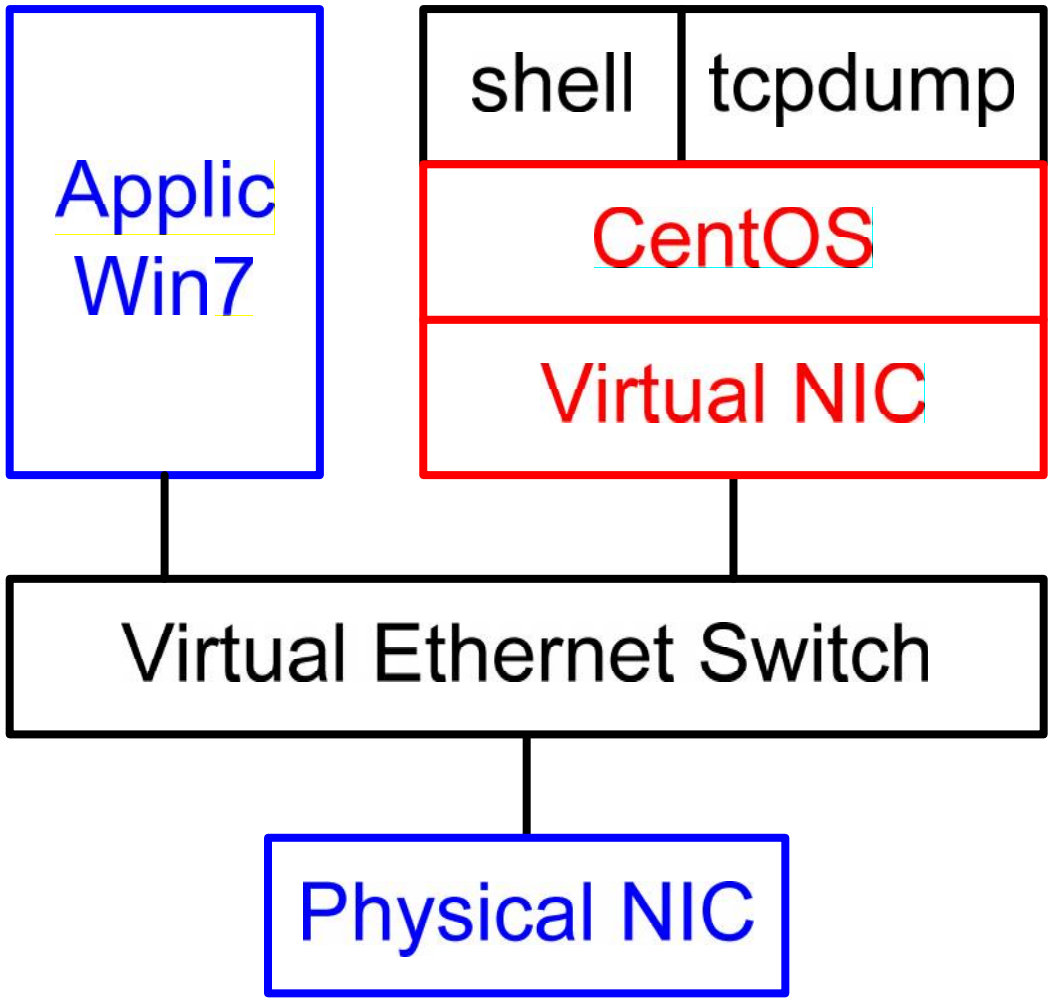
  Dans les 2 fichiers xml → §1.2

- **Que contient un fichier ova ?**

  Le fichier du disque virtuel + les fichiers xml

Synthèse du labo Vbox 1 (2/2)

- **Quelles sont les adresses ethernet utilisées ?**

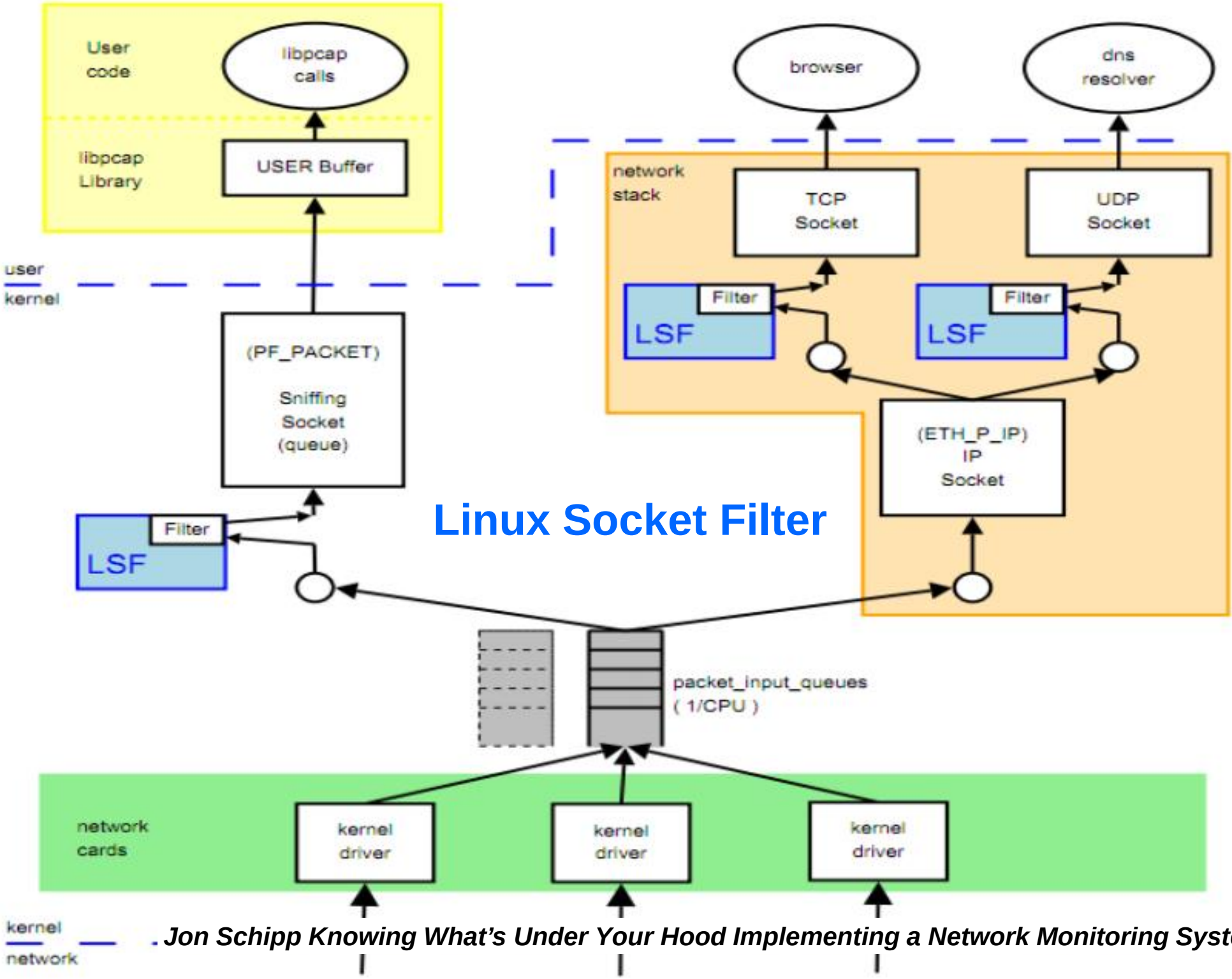


Win7 → adr. Eth physique

VM → adr Eth virtuelle

Vbox configure donc l'interface Eth physique en mode Promiscuous

Vbox (WinPcap) s'occupe du filtrage Ethernet


User code
libpcap calls
libpcap Library
USER Buffer
user
kernel
(PF_PACKET)
Sniffing Socket (queue)
LSF
Filter
browser
dns resolver
network stack
TCP Socket
UDP Socket
LSF
Filter
LSF
Filter
(ETH_P_IP)
IP
Socket
Linux Socket Filter
packet_input_queues
(1/CPU)
network cards
kernel driver
kernel driver
kernel driver
kernel
network

Linux Socket Filter (LSF)

- La partie droite représente un navigateur (tcp) et un client DNS (udp)

  `ETH_P_IP` supporte all of the IP-suite protocols : TCP, UDP, ICMP

- La partie gauche correspond à un sniffer (Wireshark - tcpdump)

  `pf_packet` supporte un mode RAW

- La partie inférieure montre 3 interfaces réseau (physiques)

  Tous les paquets admis au niveau Ethernet (Ex 2) sont diffusés

- Linux Socket Filter

  Aucun filtrage pour la partie gauche → mode Promiscuous

  Filtrage Eth – IP – TCP/UDP pour la partie droite

- From http://www.linuxjournal.com/article/4659

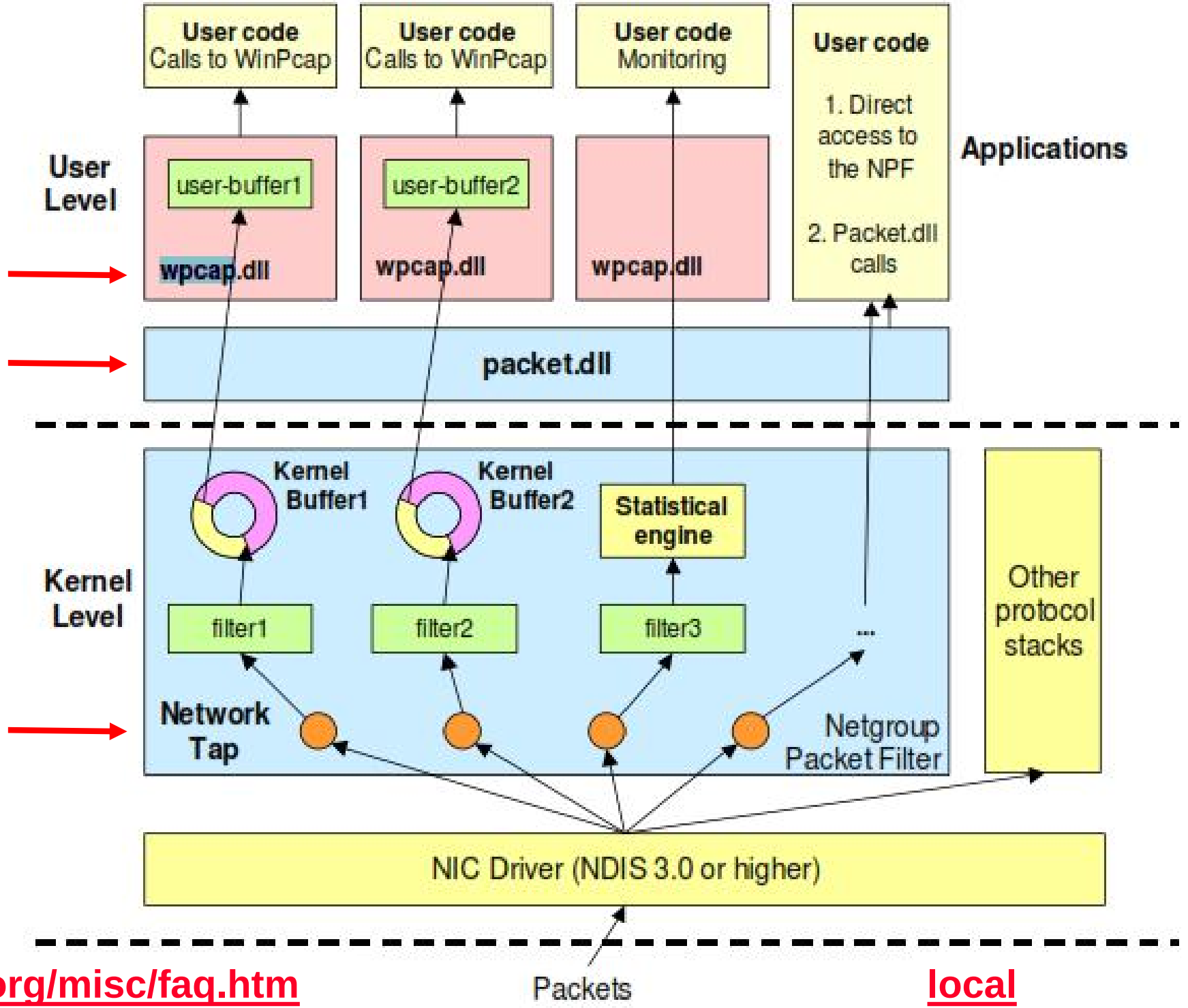
WinPcap

3 fichiers

wpcap.dll

packet.dll

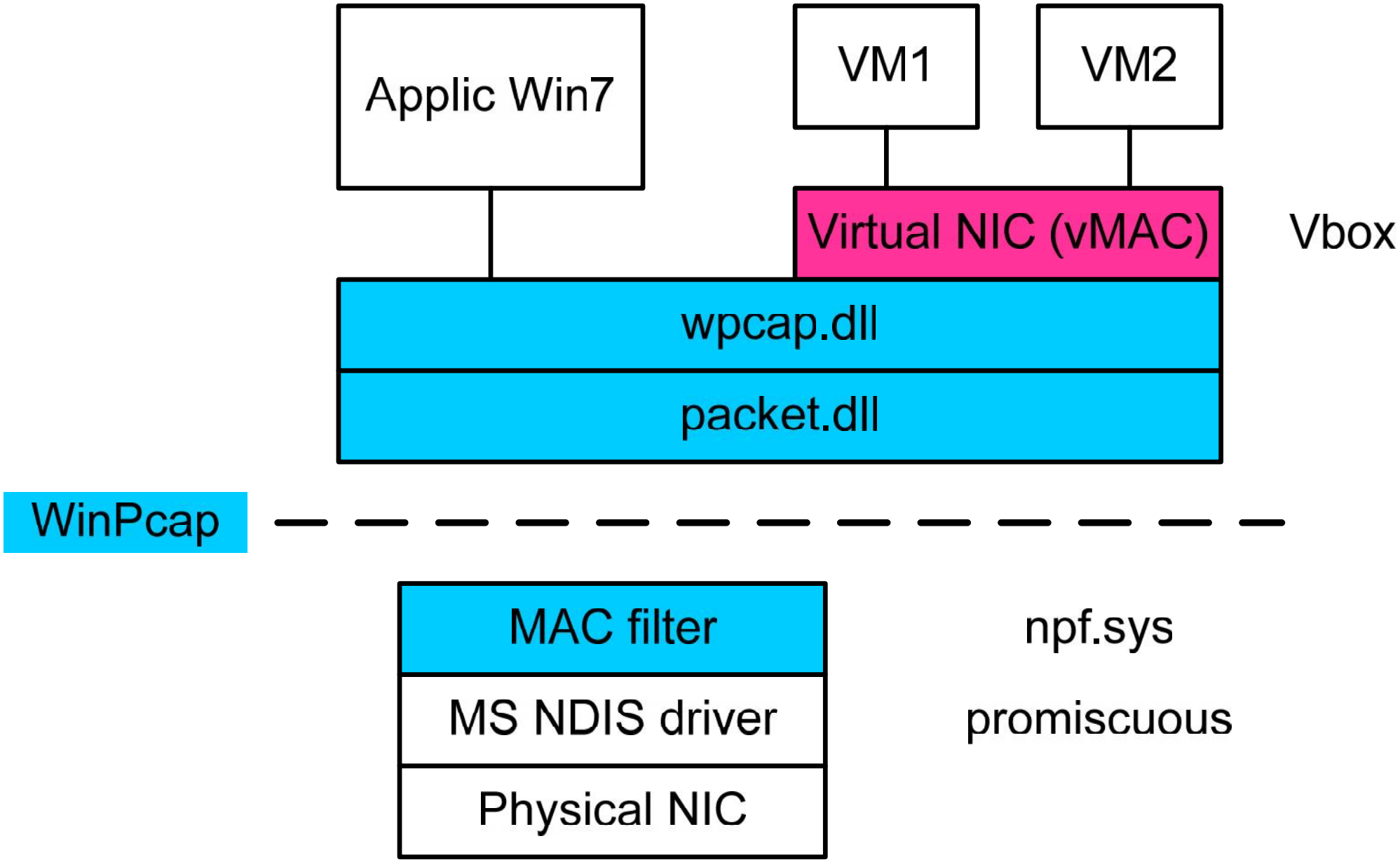
npf.sys



www.winpcap.org/misc/faq.htm

local

Vbox : Bridged networking mode (détail)



Vbox : Bridged networking mode (détail)

- Le port physique doit être mis en mode promiscuous

- Le filtrage est réalisé par npf.sys
  Il est basé sur Linux Socket Filter

- Vbox émule des interfaces Ethernet virtuelles

- Vbox + WinPcap implémente un switch virtuel

§5.1 Hard disk controllers: IDE, SATA (AHCI), SCSI, SAS

- In a **real PC**, hard disks and CD/DVD drives are connected to a device called **hard disk controller** which drives hard disk operation and data transfers

- Vbox can **emulate** the four most common types of hard disk controllers typically found in today's PCs: IDE, **SATA**, SCSI and SAS

- Emulation de périphérique

  Vbox fait croire à l'utilisateur qu'il possède ce type de périphérique

  Ce périphérique n'est pas physique mais virtuel

Stockage sur disques (suite dans chapitre Storage)

- **DAS** *(Direct Attached Storage)*

  Disque(s) relié(s) physiquement au PC ou au *mainframe*

  http://en.wikipedia.org/wiki/Direct-attached_storage

- **Principaux paramètres des disques durs *(Hard Disk)***

  Capacité (MByte), MTBF *(Mean Time Before Failure)*

  Interfaces = ATA/IDE, SCSI *(Small Computer System Interface)*, SATA *(Serial ATA)*, *SAS (Serial Attached SCSI)*

  Performance = F *(blocksize*, *read/write*, *sequential/random*, *cache)*

  IOPS *(I/O per s)*, *thoughput (Mbyte/s)*, *response time*

  http://www.passmark.com/products/pt_advdisk.htm

- Technologie = …, SSD (Solid State Drive), …

§5.2 Disk image files : VDI, VMDK, VHD, HDD (1/3)

- Disk image files **reside** on the host system (Win7) and are seen by the guest systems (Linux) as hard disks of a certain geometry

- When a guest OS (Linux) reads from or writes to a hard disk, Vbox **redirects** the request to the image file

- Like a physical disk, a virtual disk has a **size (capacity),** which must be specified when the image file is created

- As opposed to a physical disk however, Vbox allows you to **expand** an image file after creation

§5.2 Disk image files : VDI, VMDK, VHD, HDD (2/3)

VirtualBox supports 3 variants of disk image files :

- **Normally**, Vbox uses its own container format for guest hard disks – Virtual Disk Image (**VDI**) files

  This format will be used when you **create** a new VM with a new disk

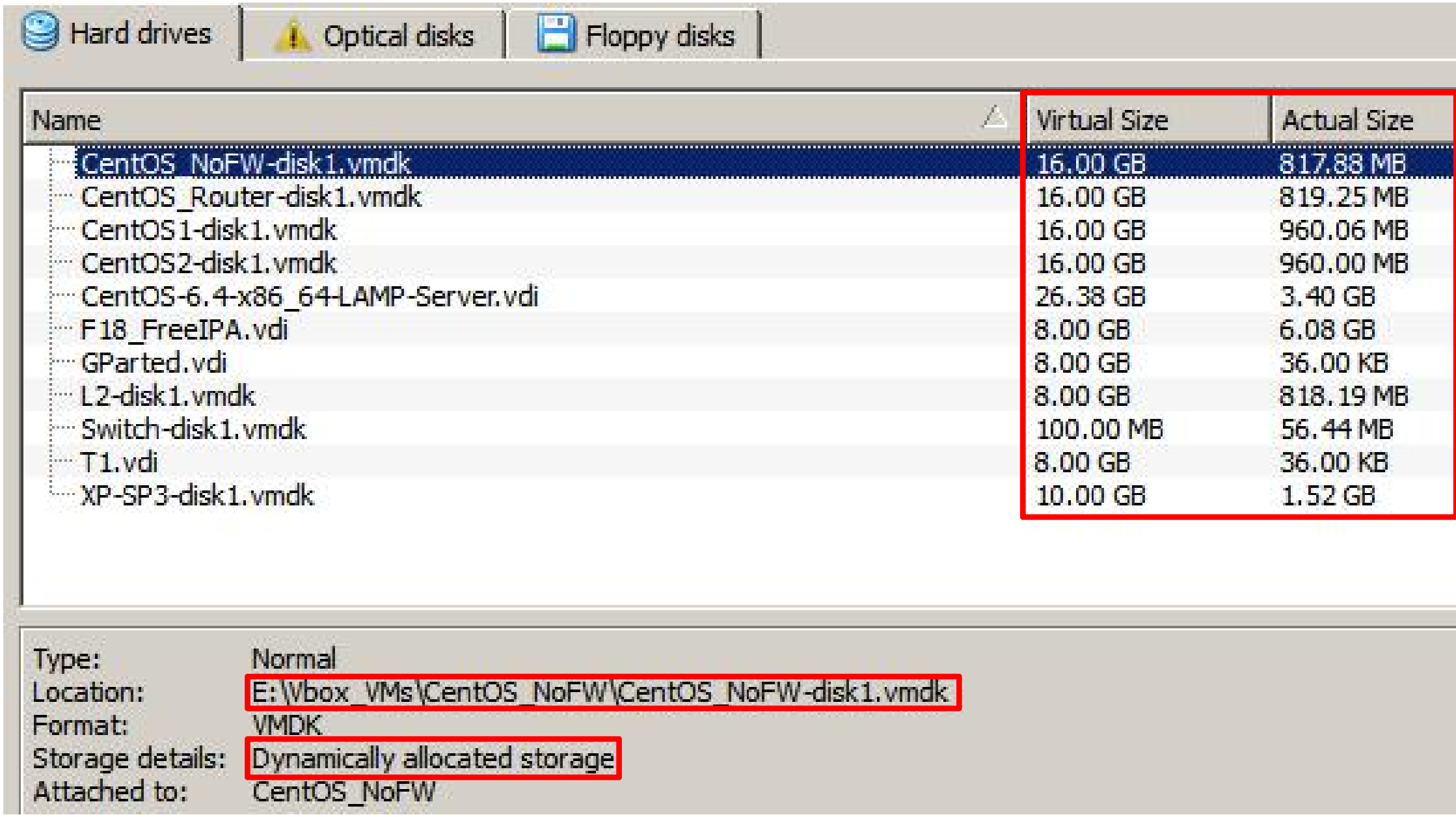
- Vbox supports the popular and open **VMDK** format (VMware)

- Vbox supports the **VHD** format (Microsoft)

§5.2 Disk image files : VDI, VMDK, VHD, HDD (3/3)

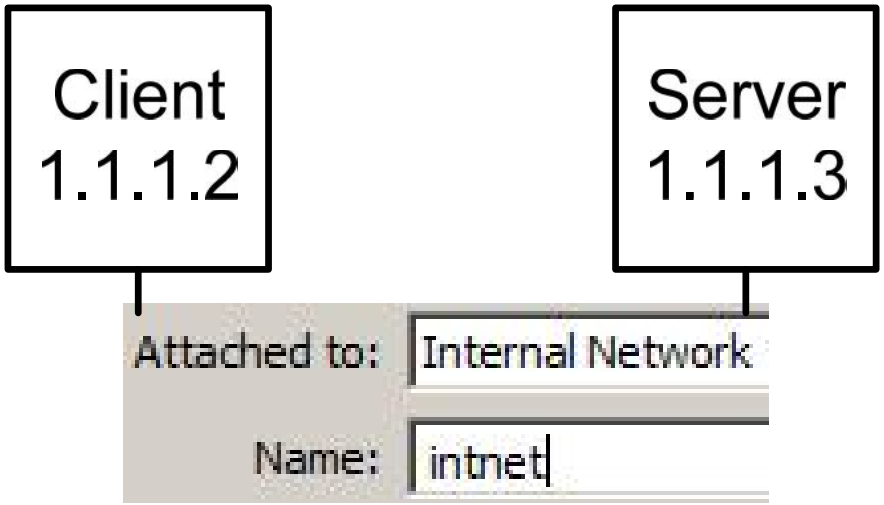
- If you **create a fixed-size image**, an image file will be created on your host system which has roughly the same size as the virtual disk's capacity. So, for a 10G disk, you will have a 10G file.
- Note that the creation of a fixed-size image can take a **long time**
- For more flexible storage management, use a **dynamically allocated image**
- This will initially be very small and **not occupy any space for unused virtual disk sectors,** but will grow every time a disk sector is written to for the first time, until the drive reaches the maximum capacity chosen when the drive was created
- Labo Vbox précédent : le disque virtuel occupe 817 MByte et émule un disque de 16 GB

§5.3 Virtual Media Manager (File ou Ctrl D)



§6.5 Internal Networking

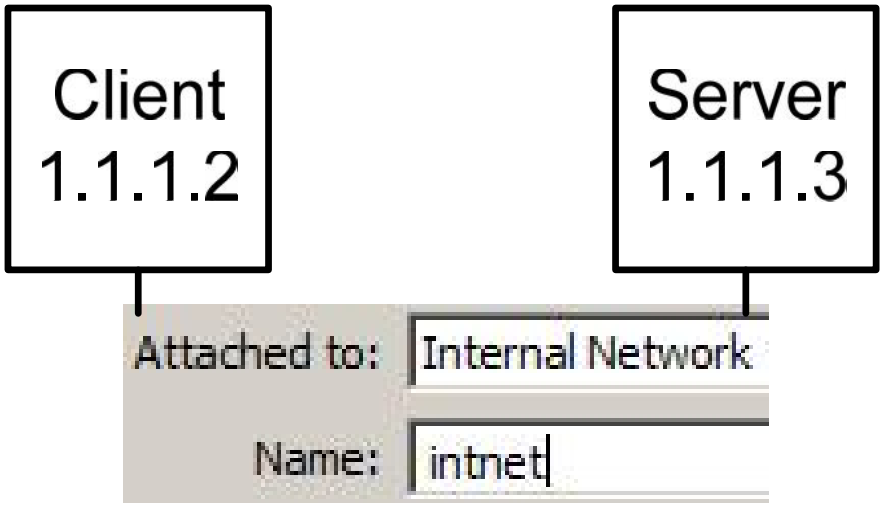
- Internal Networking is similar to bridged networking **without physical access**



- Internal networks are created automatically

- The Vbox driver implements a complete Ethernet switch and supports both broadcast/multicast frames and promiscuous mode

Labo §5 : Client – Server basés sur CentOS-6.4 (20 min)



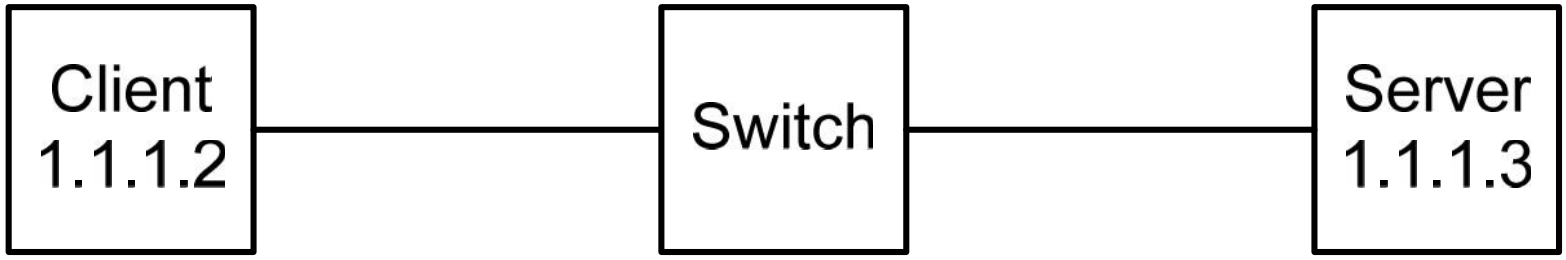
- CentOS_C.ova à disposition
- Clone pour produire CentOS_S
- **`nano /etc/sysconfig/network-scripts/ifcfg-eth0`**
- Tests avec ping

/etc/sysconfig/network-scripts/ifcfg-eth0

```
DEVICE=eth0
TYPE=Ethernet
#UUID                          used by Network Manager
ONBOOT=yes                     Device activated at boot-time
NM_CONTROLLED=no               No Network Manager
BOOTPROTO=none                 no boot protocol
HWADDR=08:00:27:b5:a2:eb
IPADDR=192.168.1.10
NETMASK=255.255.255.0
GATEWAY=192.168.1.1
DNS1=192.168.1.1
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth0"
```

Labo §6 VirtualSwitch : Commutateur basé sur Linux

- 3 appliances à disposition



  Client & Server du §5

  Switch basé sur Linux Microcore 4.0 (version minimale de 69 MB)

- Configurer le switch (next slide)

- Tester depuis Client avec ping 1.1.1.3 → Pas de connexion !

- Méthodologie de dépannage (troubleshooting) avec tcpdump

Linux Virtual Switch (brctl)

- Créer un switch virtuel à partir d'un système Linux

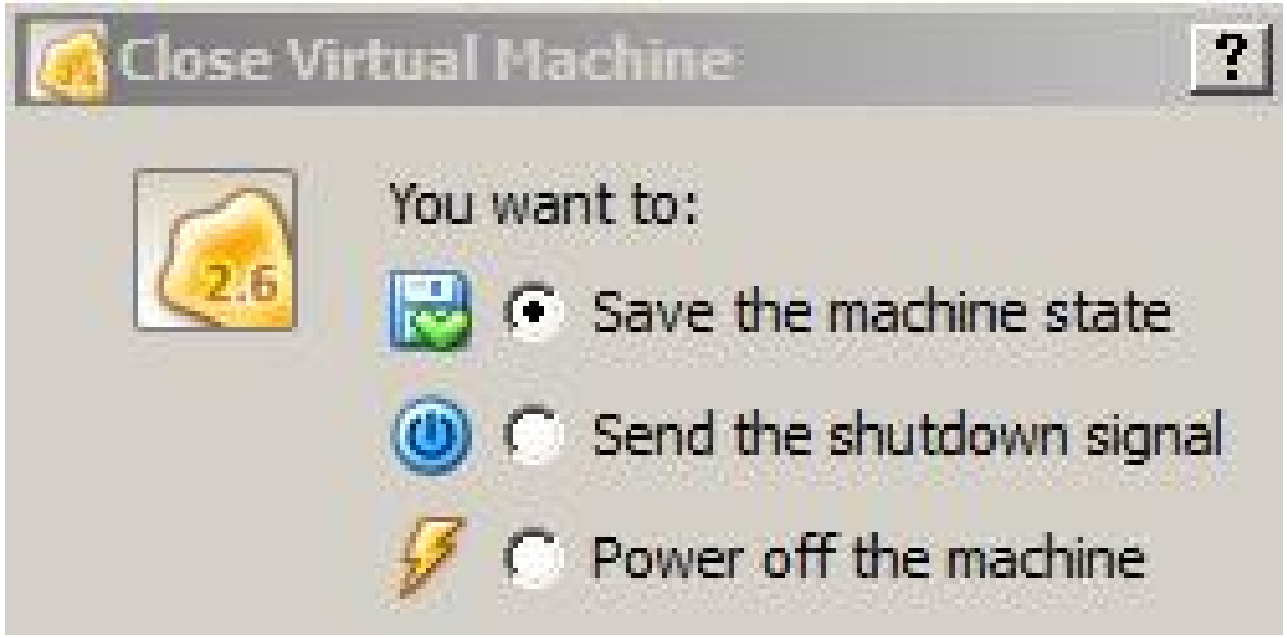
  Intérêt pédagogique

- CLI

```
ifconfig eth0 0.0.0.0      no IP addr
ifconfig eth1 0.0.0.0
brctl addbr b1             add bridge
brctl addif b1 eth0        add interface
brctl addif b1 eth1
brctl show                 show config
ifconfig b1 up
```

- http://www.linuxfoundation.org/collaborate/workgroups/networking/bridge
- http://linux.die.net/man/8/brctl

Close VM (§1.8.6) & Start VM



- **Save the machine state** permet de conserver l'état du système lors d'un prochain Start

- Les 2 autres choix produisent un **redémarrage** (boot) lors du prochain Start

Ex 3 : Couches présentes dans l'émulation du disque

- Déterminer le modèle en couche correspondant au scénario suivant :
  Guest = CentOS

  Emulation du disque SATA par Vbox

- Lors d'un accès fichier (lire par exemple), un mécanisme de fragmentation est présent pour adapter la taille du fichier à celle des blocs mémorisés sur le disque.

  Quelle couche effectue cette fragmentation ?

- Les termes suivants doivent être présents sur la figure : FS-Ext4, FS-Size, Disk-Bloc-Size, SATA controller, …

Synthèse du labo Vbox 2 (1/2)

- **Quel est l'intérêt de Internal Network ?**

  Offrir un(des) réseau(x) ethernet virtualisé(s)

  Aucun échange avec les modes NAT ou Bridge

- **Expliquer le mode Promiscuous activé au labo §6**

  Par défaut (Promiscuous = deny), une VM reçoit les paquets qui lui sont destinés (filtre sur adr Eth + broadcast)

  La VM-Switch ne reçoit donc pas le paquet ARP response car il est de type unicast contrairement au paquet ARP request

  Ce filtre est désactivé avec Promiscusous = Allow All comme le fait Wireshark sur une interface physique

Synthèse du labo Vbox 2 (2/2)

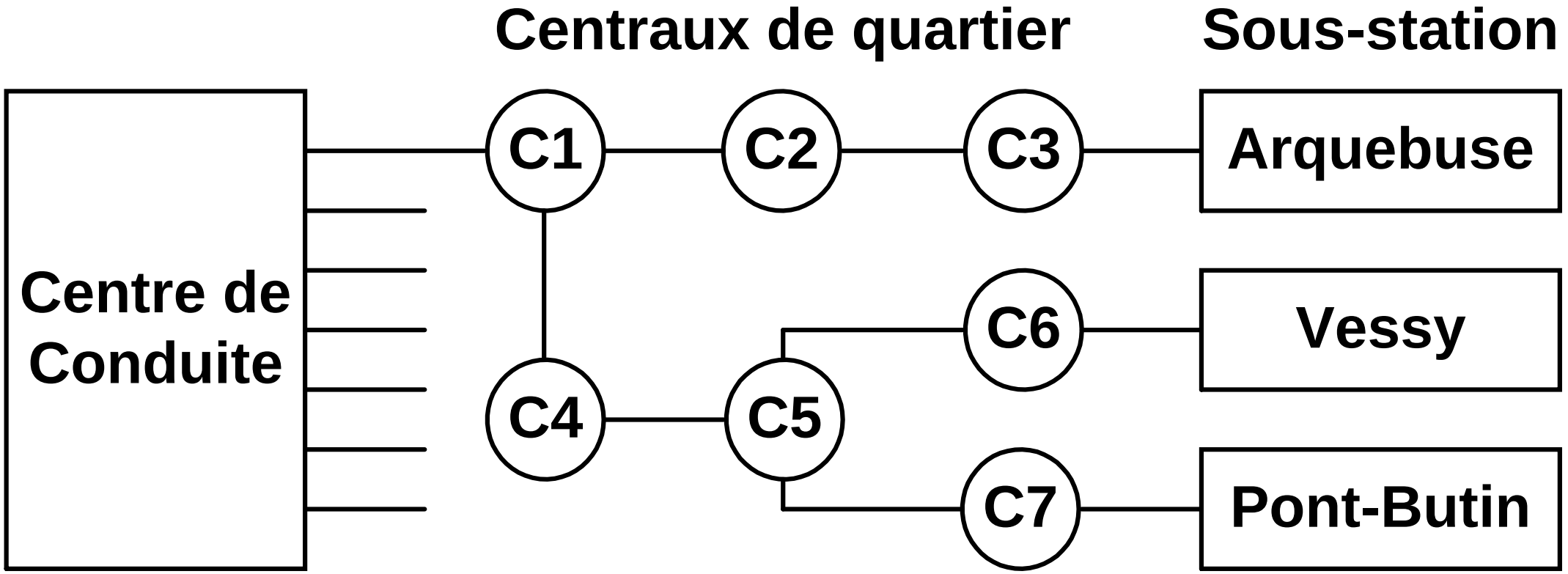
- **Quelles ont été les principales difficultés rencontrées au labo §6 ?**

  Utiliser 2 Internal Networks → 1 de chaque côté du Switch

  Arrêter la VM client – clone – reset MAC adresse –
  ok si on respecte la procédure du labo 1

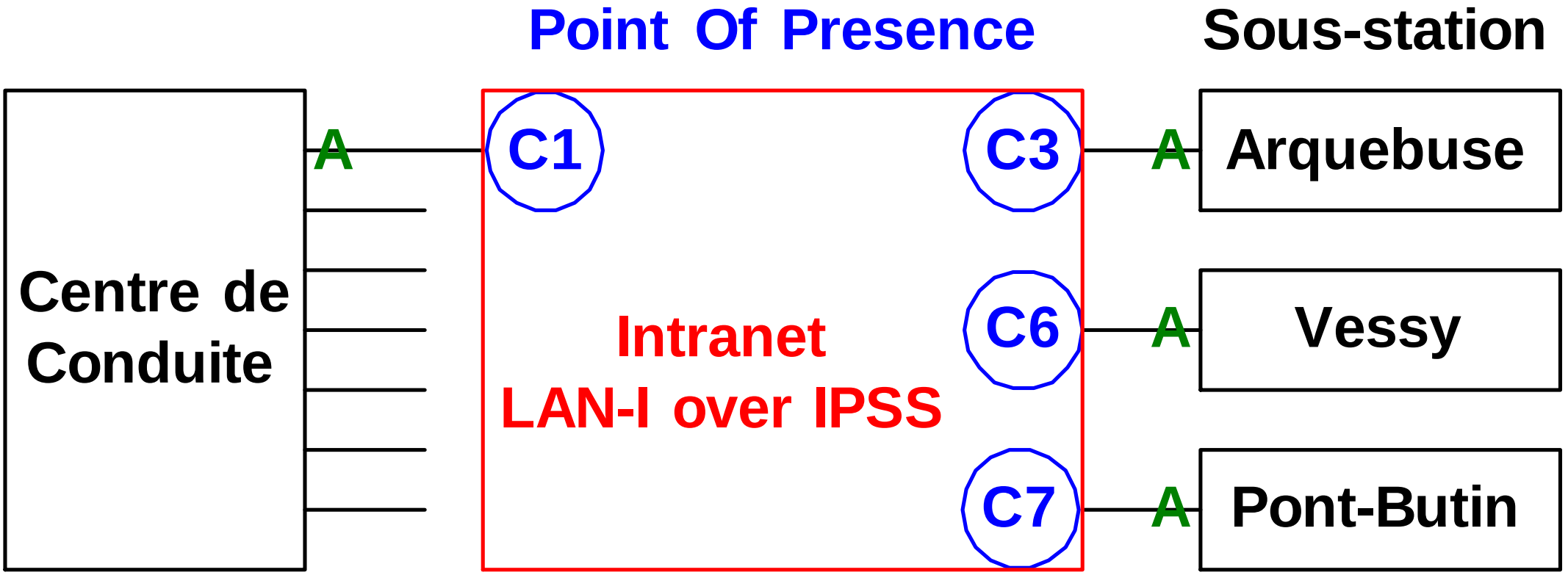
  Placer les interfaces de la VM Switch en mode Promiscuous (dans protocole labo)

Migration du réseau SIG-Eau vers …



- Multipoint cuivre + modems
- Transmission asynchrone à 600 bit/s
- Protocole de type polling (commande - réponse)
- http://www.tdeig.ch/projet/SIG_Dev_Durable.pdf

Variante proposée

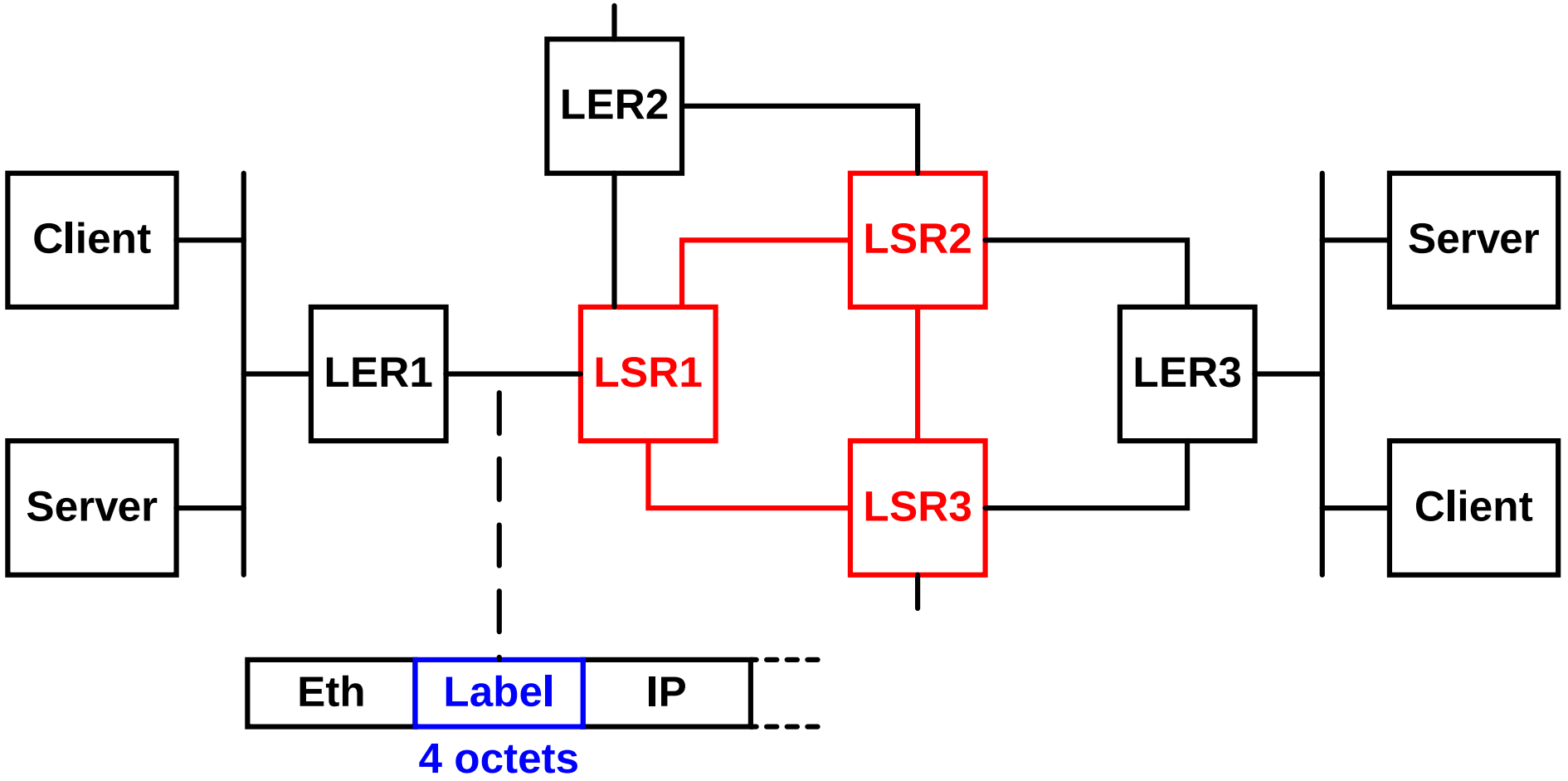


- Accès via liaison cuivre ADSL (64 – 2048 kbit/s)
- Accès fibre optique
- Aucun accès depuis internet
- Coût mensuel + frais installation

Adaptation

Rabbit

Multi Protocole Label Switching (MPLS)



- **Access network** composé de Label Edge Router (**LER**)
- Label de 4 octets ajouté entre couches ethernet et IP
- **Core network** composé de Label Switching Router (**LSR**)

Réseau privé virtuel

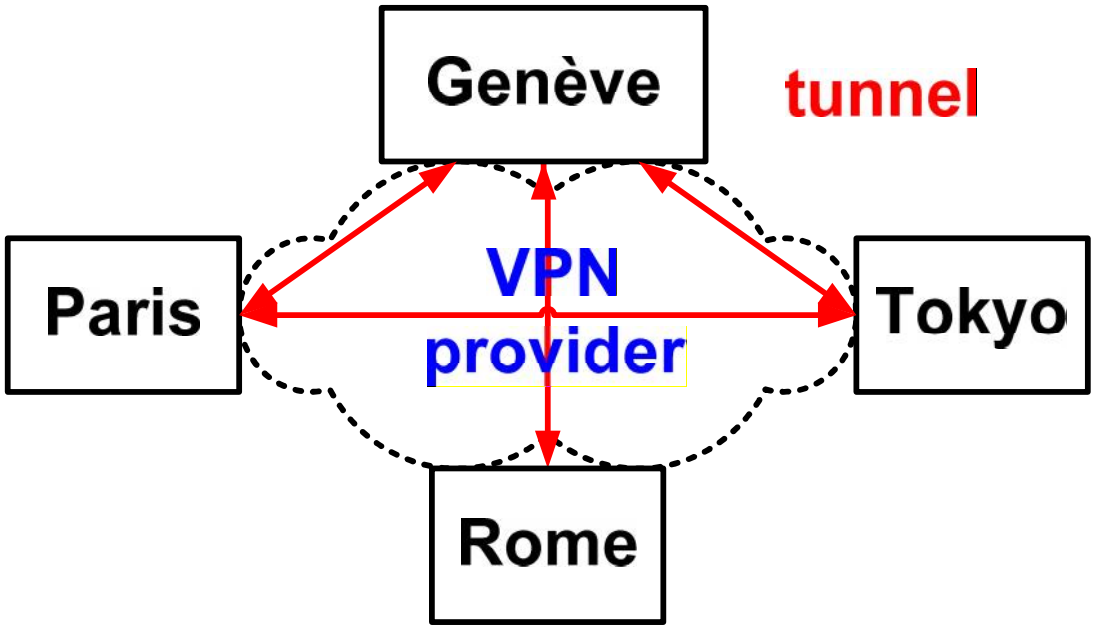
- Utiliser le réseau **internet** à la place de lignes spécialisées (leased lines), et réaliser des économies, tel est le principe du réseau privé virtuel (**VPN : Virtual Private Network**)

- Un **tunnel** sécurisé (tunneling protocol) sera ainsi créé à travers le réseau internet

- Le concept de réseau privé virtuel englobe diverses fonctionnalités liées à la sécurité : **authentification, contrôle d'accès, chiffrement** (analogie avec SSL)

VPN contrôlé par l'opérateur

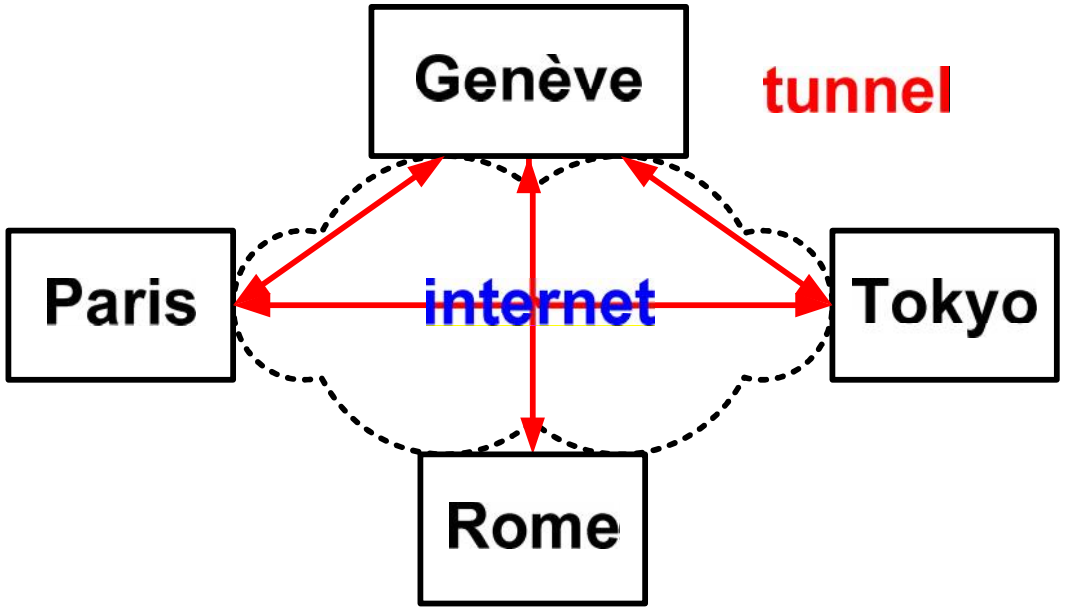
- Opérateur offre un service VPN sur son infrastructure IP-MPLS



- Plan d'adressage privé
- Partage des ressources de l'opérateur qui doit garantir cloisonnement et disponibilité
- Par défaut : ni confidentialité, ni intégrité
  → Faire confiance à l'opérateur

VPN contrôlé par l'utilisateur

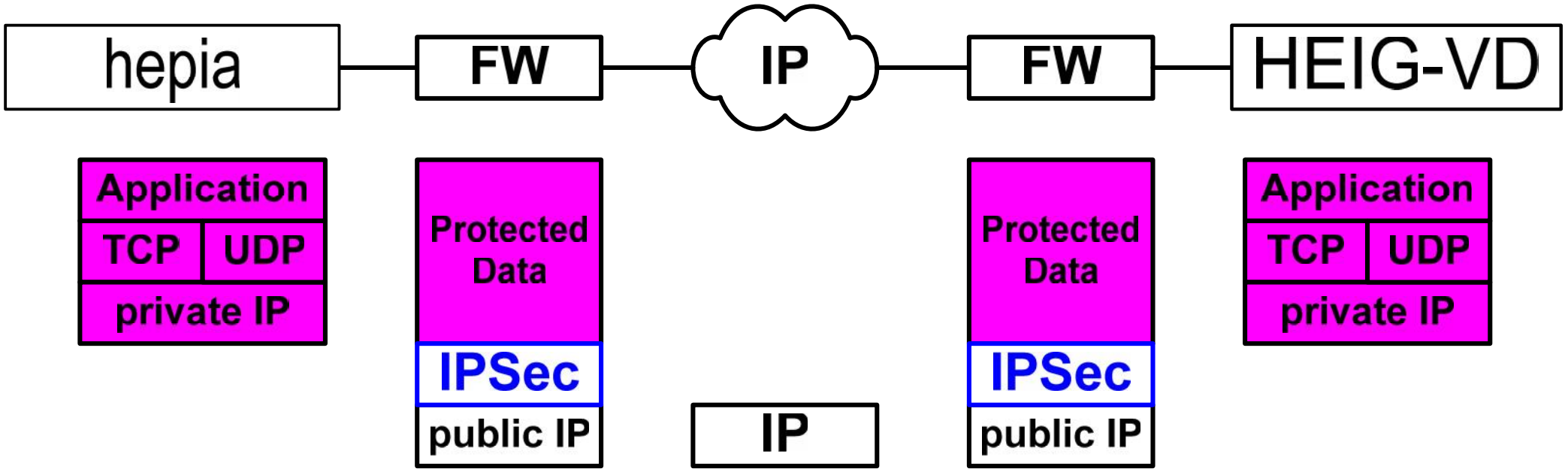
- Utilisateur dispose d'une connexion internet best effort et configure ses équipements (routeurs, firewalls, ordinateur) pour respecter les normes SSL, IPSec, PPTP, L2TP, …



- Plan d'adressage privé

- Pas de qualité de service

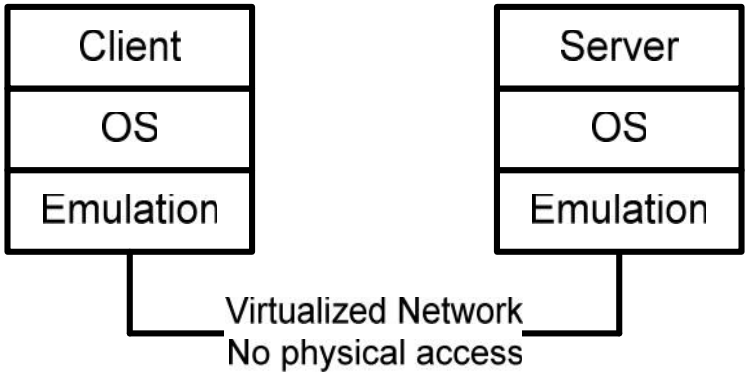
VPN entre hepia et HEIG-VD



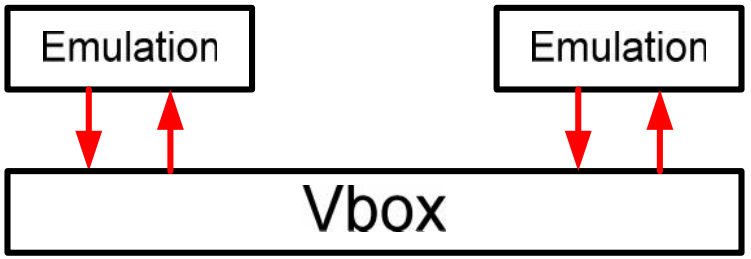
- 2 firewalls (Checkpoint - Linux) gèrent IPSec (analogue à SSL)
- Adresses IP privées et publiques → mode tunnel
- Tunnel créé entre les firewalls
- Host ne voit pas IPSec
- Services d'authentification mutuelle, de confidentialité et d'intégrité

Paravirtualisation avec virtio

- L'émulation du matériel NIC semble inutile dans ce scénario



- virtio utilise un ensemble de tampons (FIFO) partagés entre Host (Vbox) et Guest



- La même architecture est utilisée avec l'hyperviseur KVM
- On parle de paravirtualisation → slide 9
- Un contrôleur Ethernet est un composant matériel complexe (voir Annexe 5). Les pilotes (drivers) le sont également

Avantages de virtio

- Moins de lignes de code à maintenir

  La partie **front-end** est située côté VM (Guest)

  La partie **back-end** est située dans Vbox ou KVM (Host)

- Meilleure performance en évitant les copies de tampons

  Mécanisme asynchrone

- Modèle générique pour les périphériques (console, PCI, …)

```
lspci ¦ grep Ethernet
Ethernet controller: Red Hat, Inc Virtio network device
```

- Voir article de Rusty Russel

  http://ozlabs.org/~rusty/virtio-spec/virtio-paper.pdf

vboxmanage list hddbackends

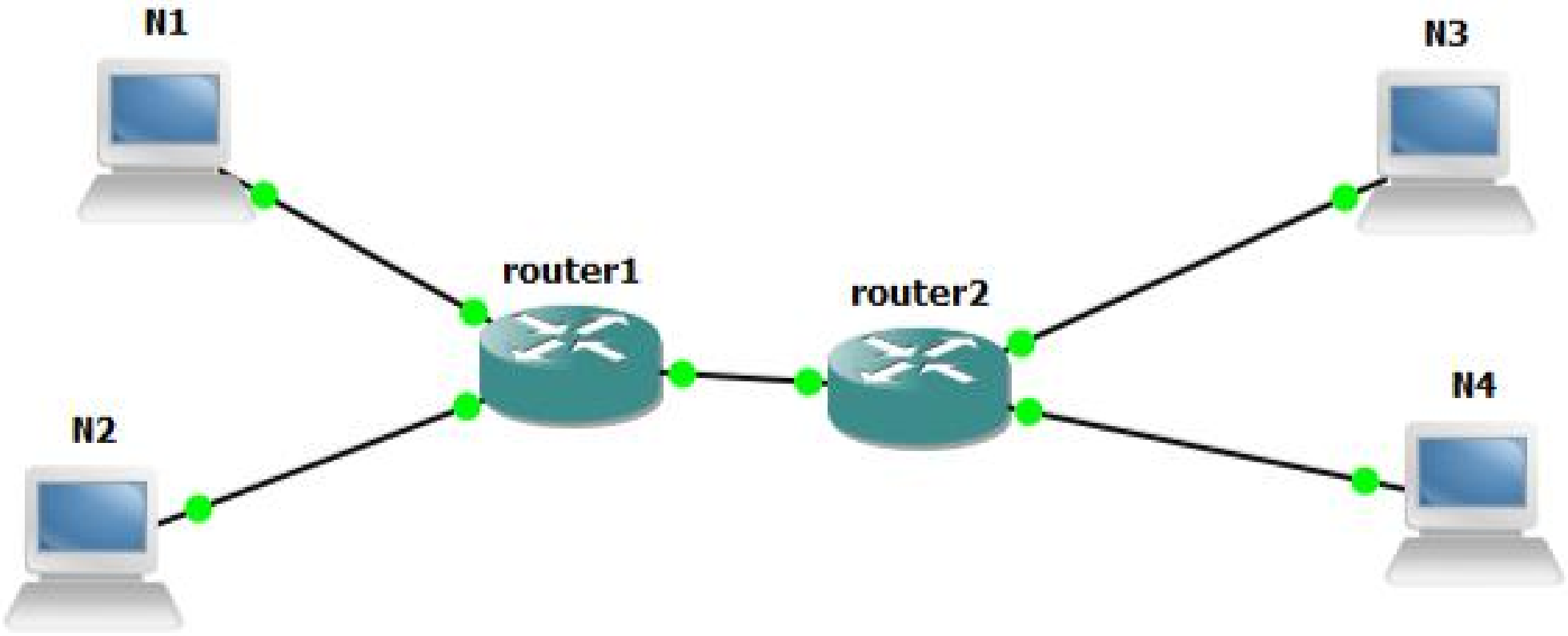
```
Backend 0: id='VMDK' description='VMDK'
  capabilities=0x027f extensions='vmdk (HardDisk)'

Backend 1: id='VDI' description='VDI'
  capabilities=0x0277 extensions='vdi (HardDisk)'

Backend 2: id='VHD' description='VHD'
  capabilities=0x0277 extensions='vhd (HardDisk)'

Backend 5: id='RAW' description='RAW'
  capabilities=0x0262 extensions='iso (DVD),cdr
  (DVD),img (Floppy),ima (Floppy),dsk (Floppy),vfd
  (Floppy)'
```

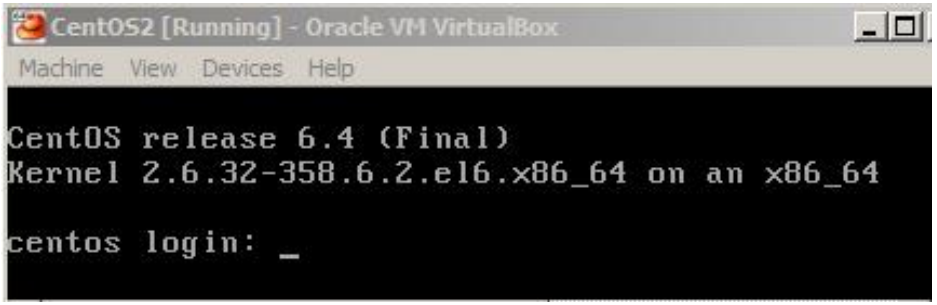
GNS3 : Graphical Network Simulator



- http://academy.gns3.com/
- Open source software (GUI) that simulate complex networks
- Uses emulators : Dynamips → Cisco, Qemu & Vbox
- Hardware emulated by GNS3 (needs Cisco licence)

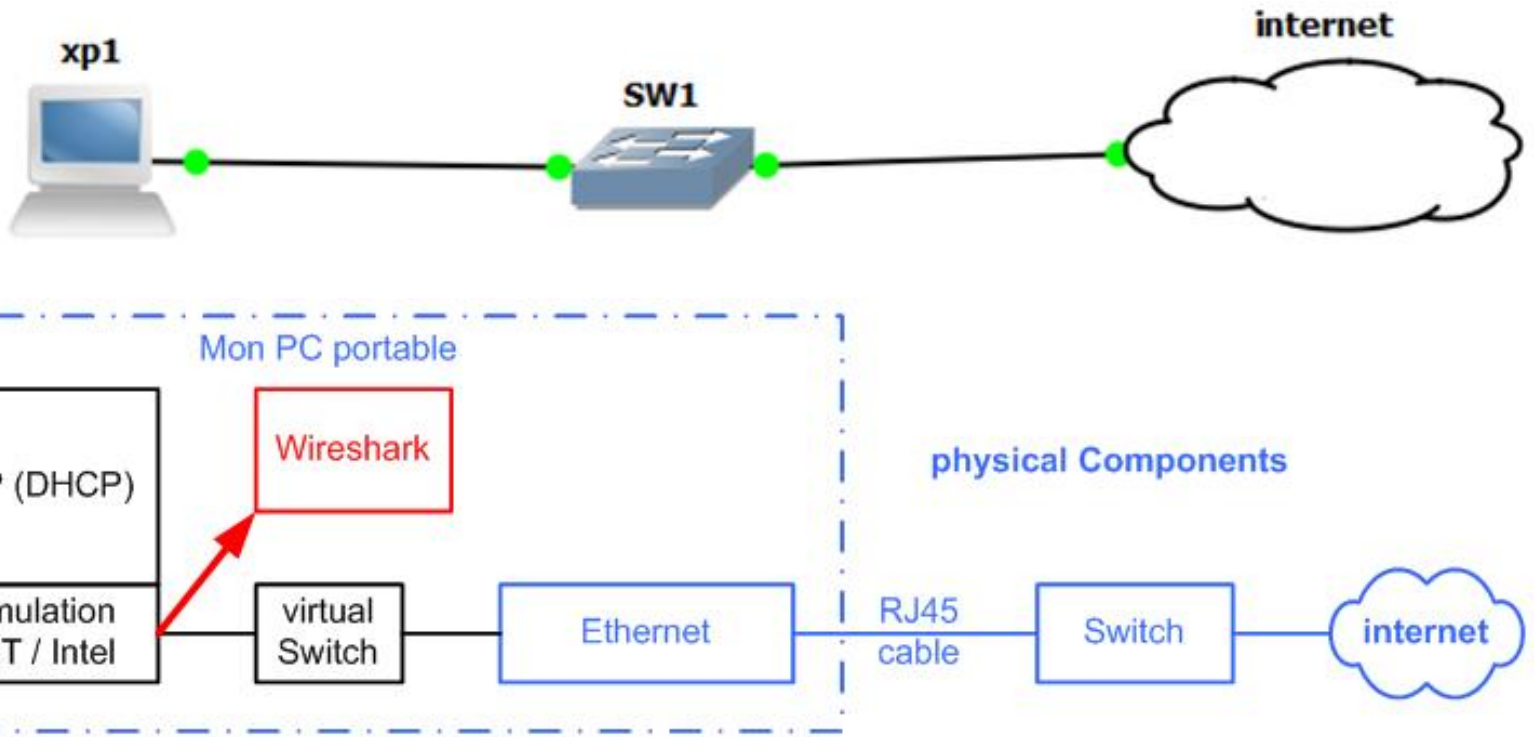
GNS3 : Communications

- By default, GNS3 uses
  Putty terminal on Windows

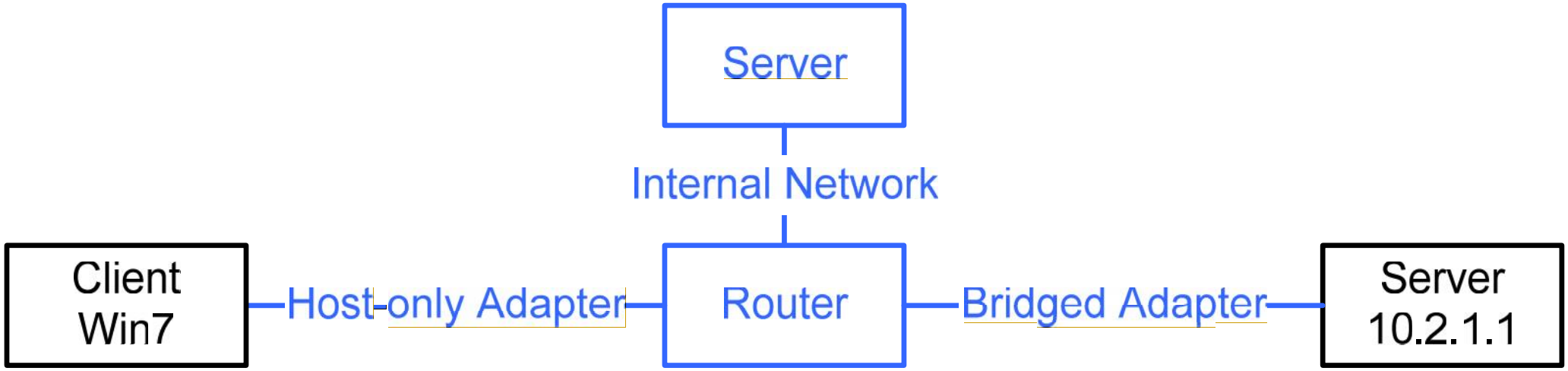


- Interconnect via Vbox – Host-only Adapter
  Choix bizarre qui ajoute une interface virtuelle par VM

- Physical &
  Virtual



Labo §8 : Virtual Router



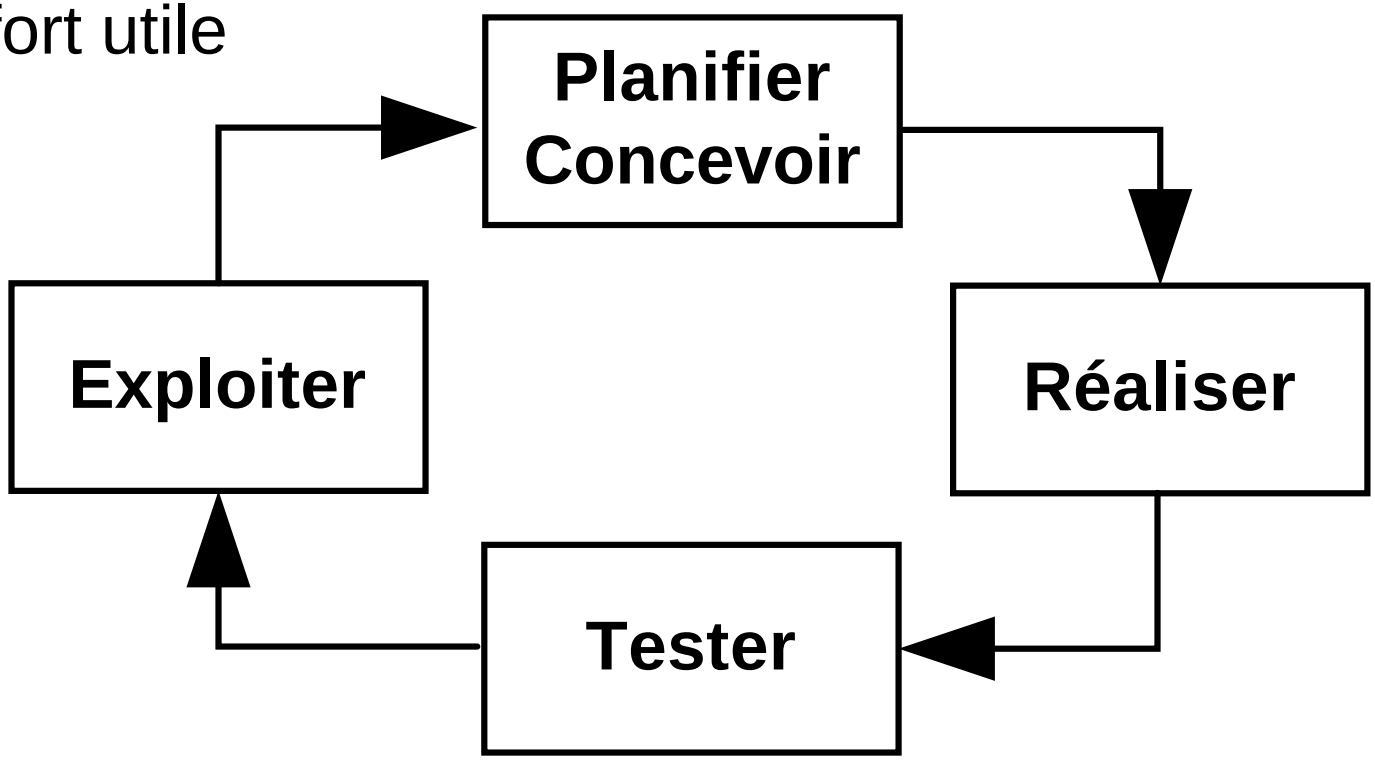
a) Quelle est la marche à suivre pour un test (ping) depuis Win7 sur le serveur virtuel ?

→ Démarche PDCA (Deming) → next slide

b) Pourquoi le test (ping) depuis Win7 sur le serveur 10.2.1.1 sera négatif ?

Roue de la sécurité (Deming) from Sécurité - Intrusions

- Le cycle PDCA *(Plan – Do – Check – Act)* de Deming demeure une référence fort utile



- Les cycles sont nombreux justifiés par de nouveaux besoins ou … des oublis
- Amélioration par la qualité
- Maintenir le niveau de sécurité constitue un *challenge* !!!

§6.5 Host-only Adapter Networking

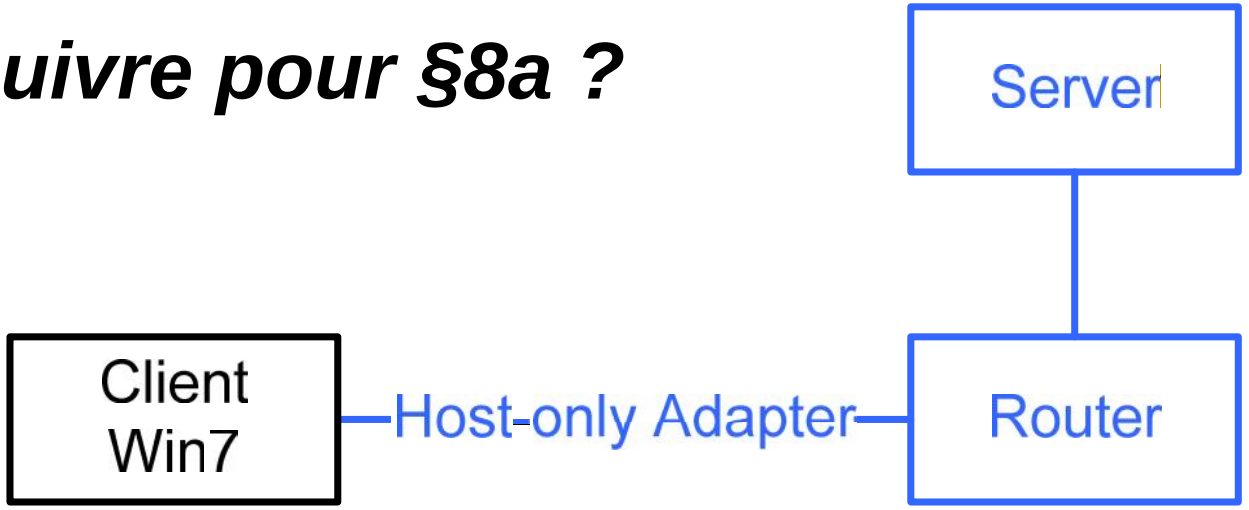
- Vbox creates a new software interface on the host which then appears next to your existing network interfaces.

- **Ethernet adapter VirtualBox Host-Only Network:**

```
Physical Address. . . . : 08-00-27-00-D4-E2
DHCP Enabled. . . . . . : No
IPv4 Address. . . . . . : 192.168.56.1(Preferred)
Subnet Mask . . . . . . : 255.255.255.0
Default Gateway . . . . :
```

- Configurer manuellement le routage !

Quelle est la marche à suivre pour §8a ?

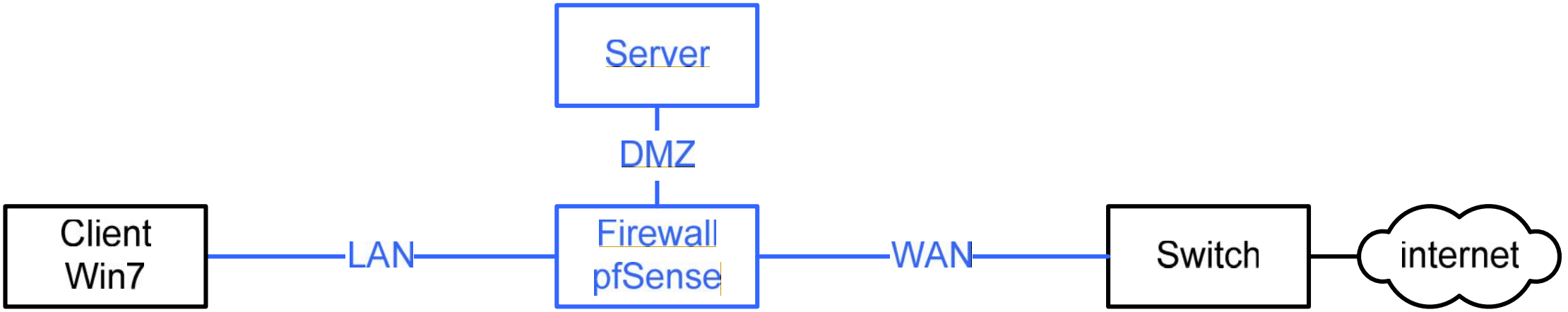


- CentOS_S.ova mise à dispo → IP = 2.1.1.2/8 R = 2.1.1.1
- **Faire un schéma avec adresses IP, masques, …**
- Clone → CentOS_R
- Configurer **`ifcfg-eth0`** & **`ifcfg-eth1`**
- Activer le routage (sécurité OS)

  `net.ipv4.ip_forward=1` dans **`/etc/sysctl.conf`**
- Supprimer la route par défaut (Win7)

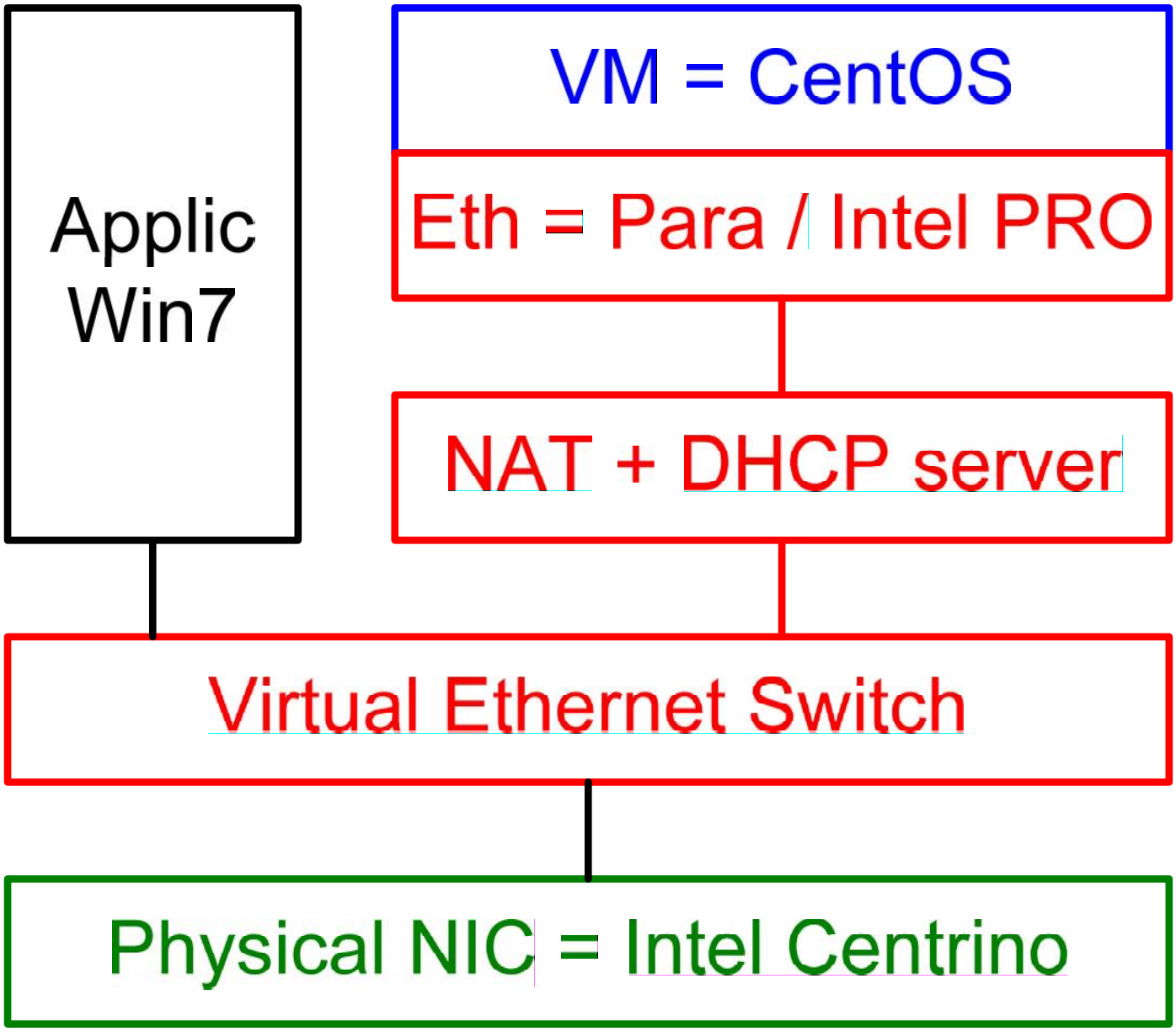
  `route delete 0.0.0.0` (droit admin)
- Ajouter la route

Labo §9 : Virtual Firewall (en réserve)



- Configurer et tester

Sol 1 : Détailler le réseau émulé par Vbox



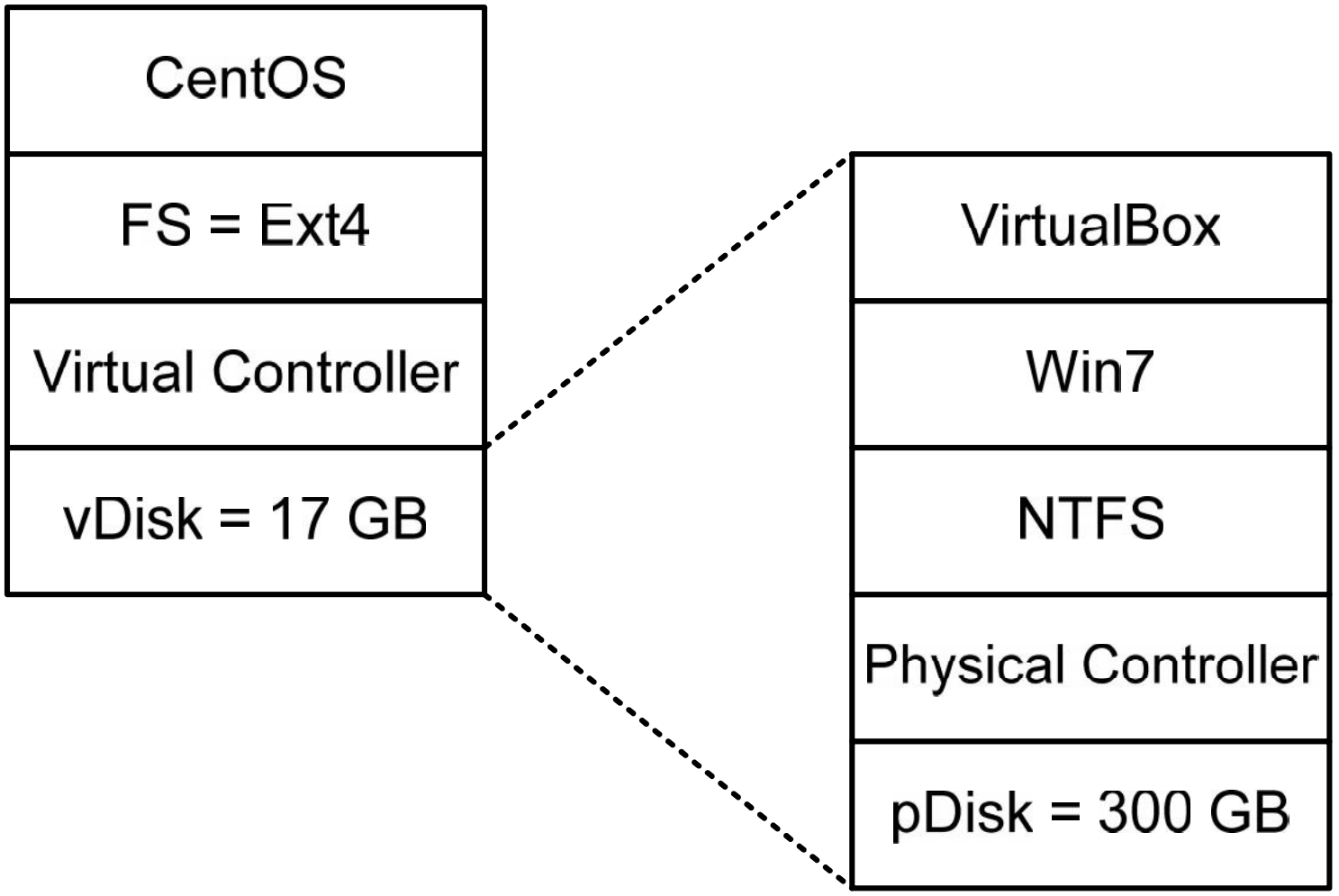
Réseau émulé par Vbox

Physical Network Interface Card

Sol 2 : Hardware Receive Frame Processing (Monitoring)

- Ethernet frame received by Network Interface Card (hardware)
- Preamble removed
- CRC checked (bad CRC dropped)
- If Promiscuous mode → capture all
  else capture if my Dest_MAC (unicast) or broadcast … multicast
- FIFO to kernel ring buffer (CPU - DMA)
- NIC generates an interrupt

Sol 3 : Couches présentes dans l'émulation du disque



- Voir Annexe 2 du labo Vbox

/etc/sysconfig/network-scripts/ifcfg-eth2

- Labo optionnel = configurer interface eth2 en mode DHCP


```
DEVICE=eth2
TYPE=Ethernet
#UUID                          used by Network Manager
ONBOOT=yes                     Device activated at boot-time
NM_CONTROLLED=yes              Network Manager
BOOTPROTO=dhcp                 boot protocol
HWADDR=08:00:27:xx:xx:xx
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=no
NAME="System eth2"
```