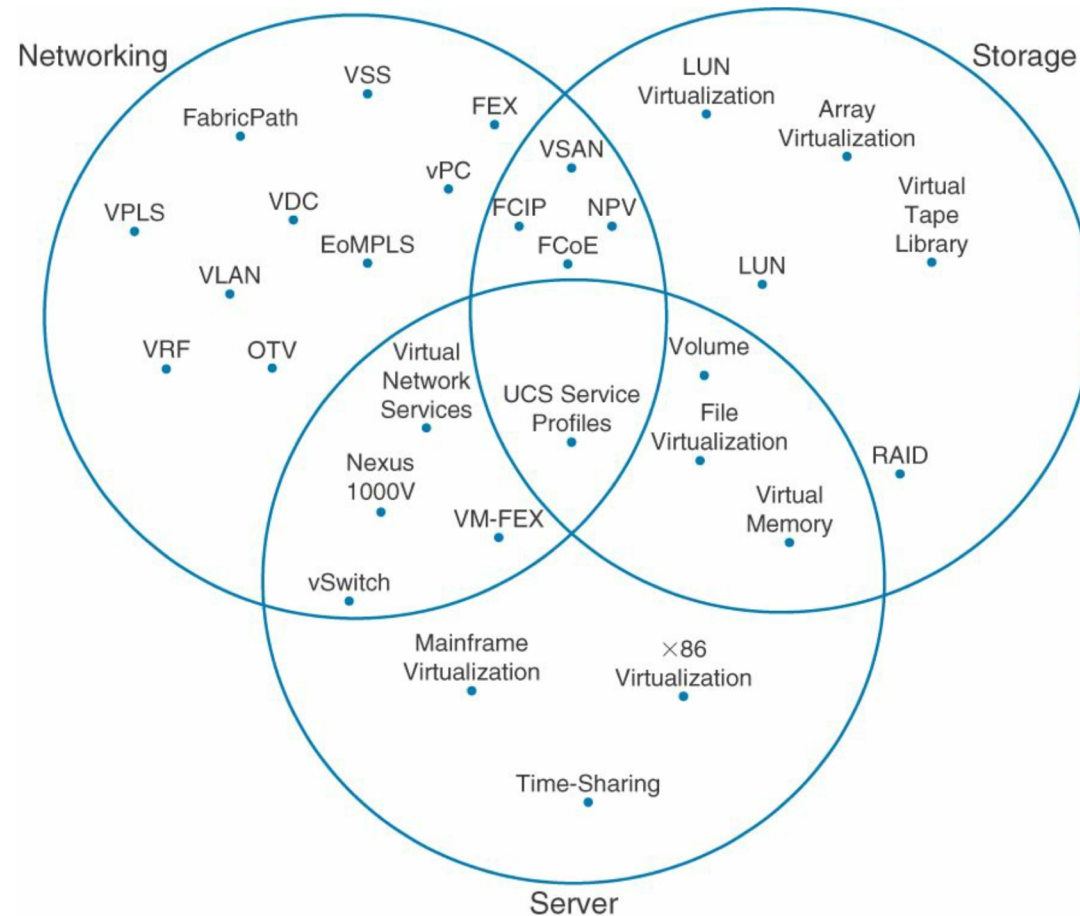


# Virtualización

# Virtualización, ¿dónde?

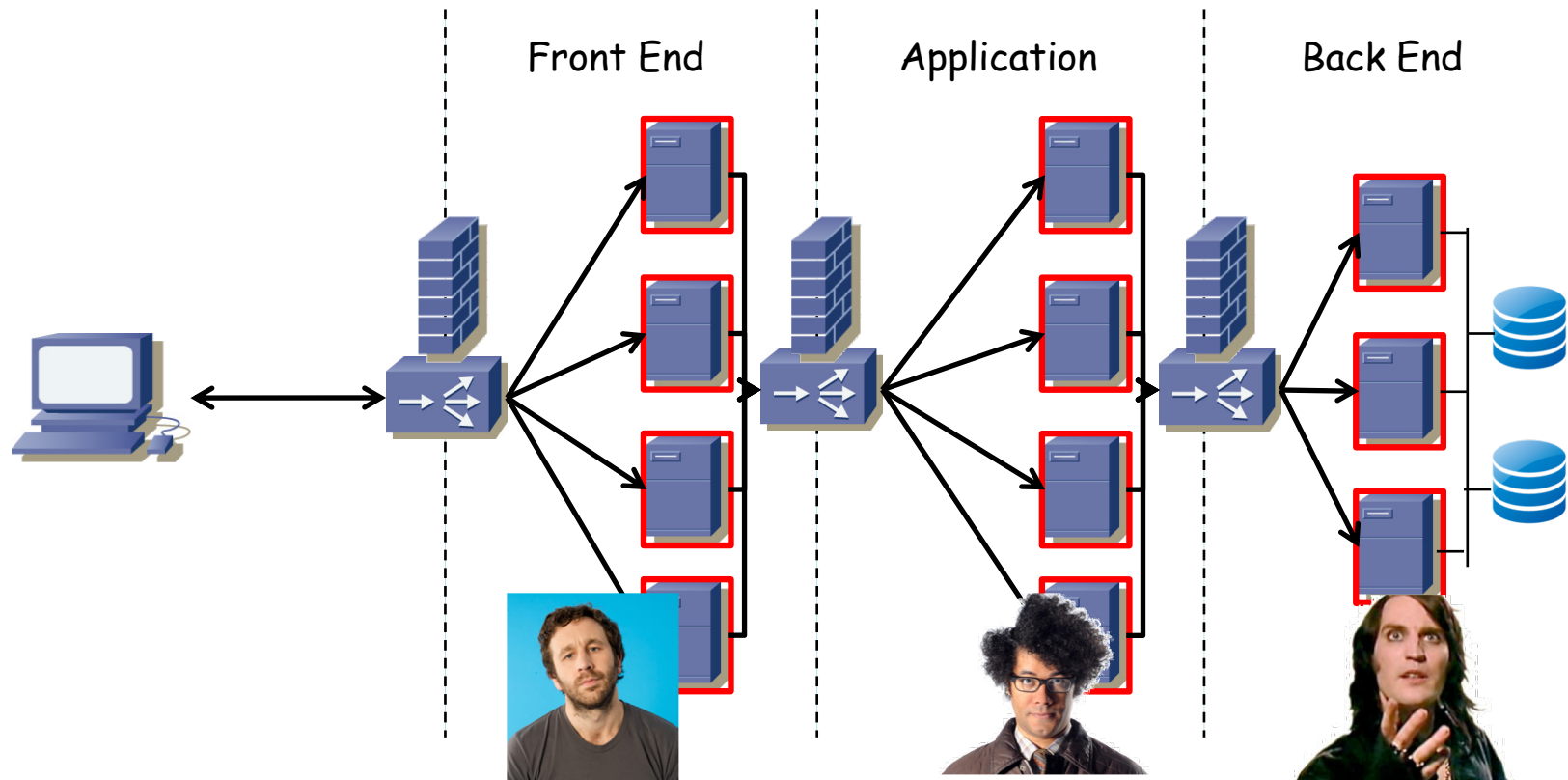
- Servidor
- Red
- Almacenamiento



# Virtualización de host

# Servidores

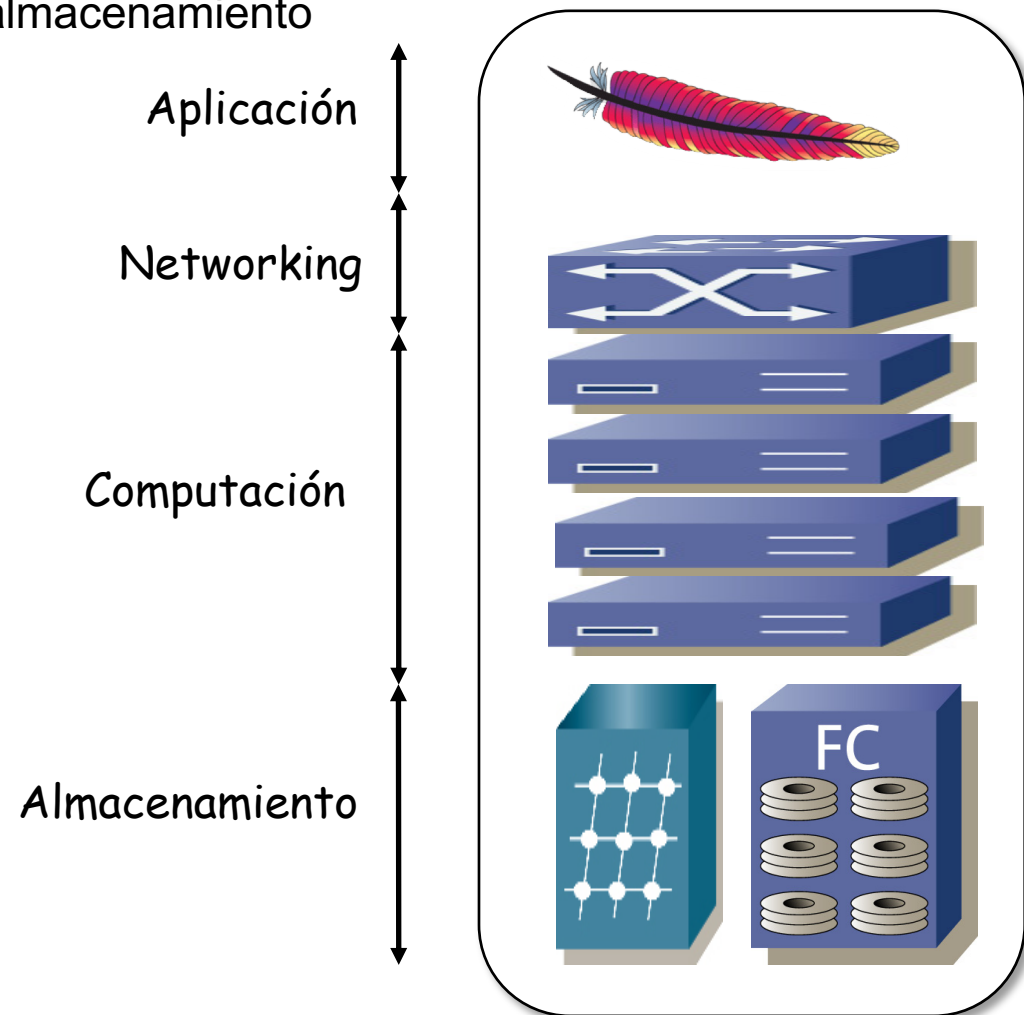
- Host físicos
- A día de hoy existen más interfaces de red virtuales que físicos
- ¿De qué estamos hablando?



# *Application Silos*

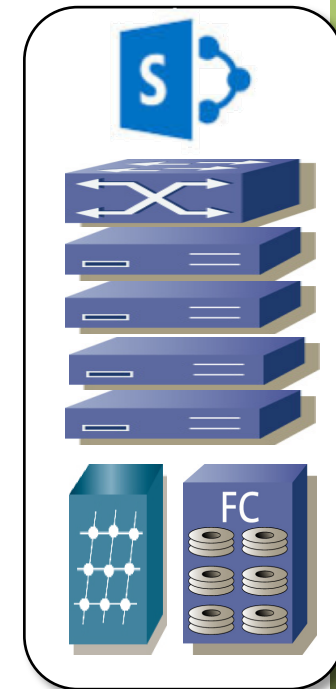
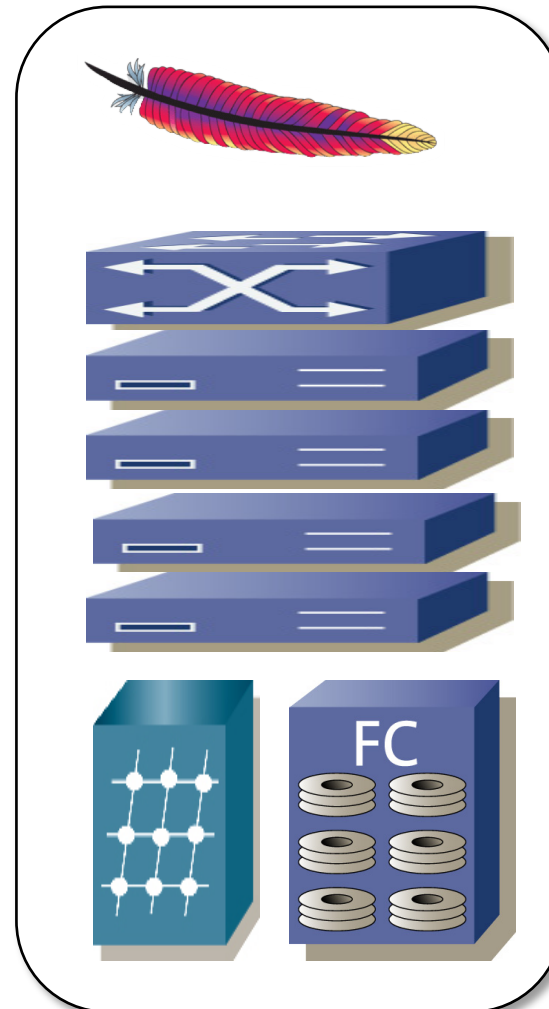
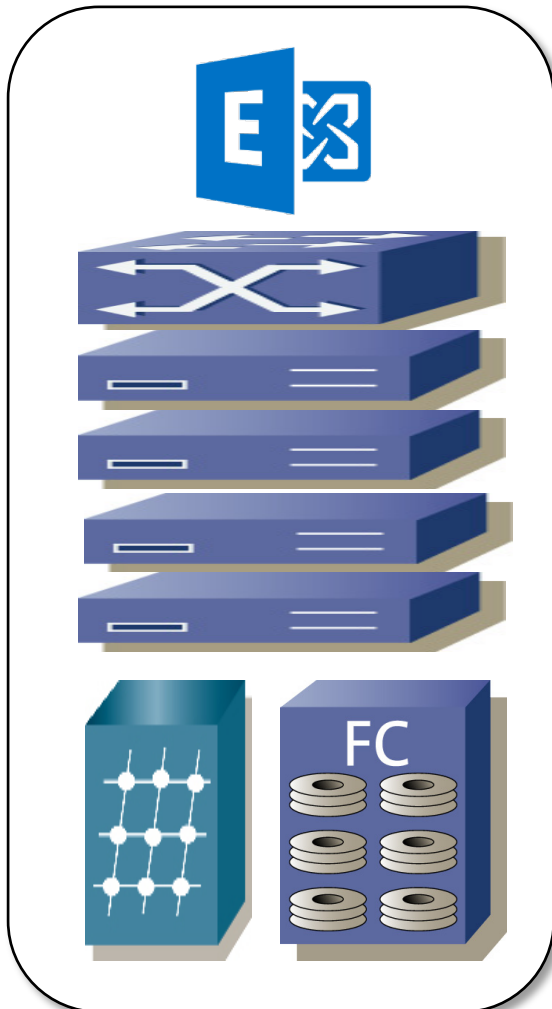
# Application Silos

- En el entorno distribuido una nueva aplicación (software servidor) se desplegaba sobre un hardware independiente
- Una relación 1:1 entre la aplicación y el hardware servidor
- O como mucho 1:N porque tengamos múltiples servidores
- A esto habría que añadirle el almacenamiento
- Y la electrónica de red



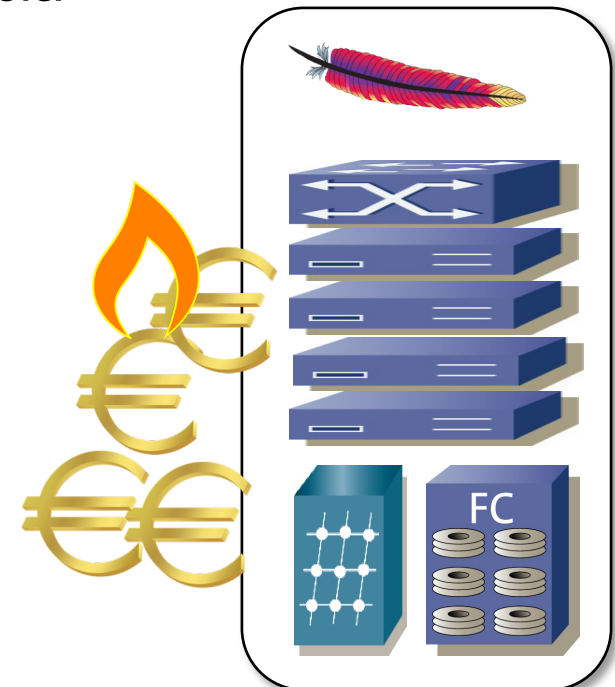
# Application Silos

- ¿Y si tenemos otra aplicación?
- Cada una sus servidores y almacenamiento
- El hardware no es reutilizable por otras aplicaciones



# *Application Silos*

- La utilización (CPU) de los servidores es muy baja
- Esto es así para soportar incrementos de carga
- Si no es baja entonces ante un incremento de carga no es rápido provisionar nuevo hardware
- Lo mismo sucede con la utilización de los discos
- Esto se multiplica por el número de aplicaciones
- Pero ocupan todo el tiempo el espacio
- Y están encendidos, consumiendo potencia
- Y necesitando refrigeración
- Esto ha cambiado con la virtualización
- Consolidación

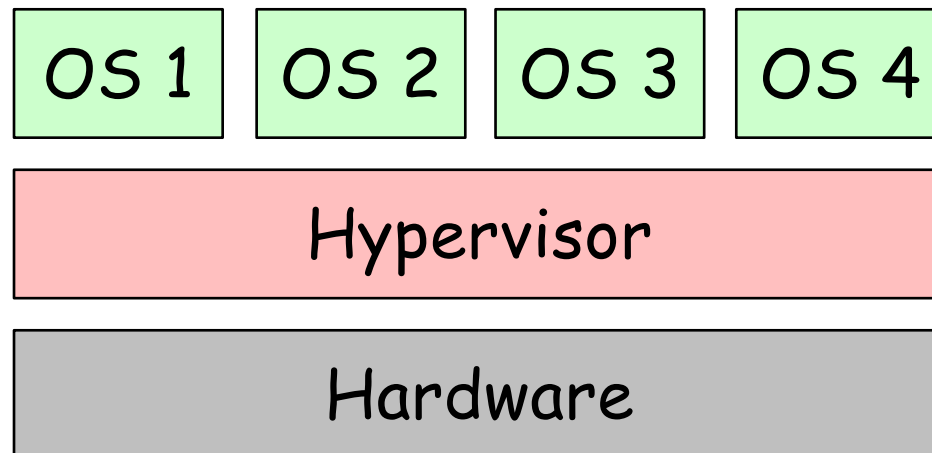




# Tecnologías tras la virtualización de host

# ¿Virtualización?

- La idea básica de virtualización del host es bastante conocida
- Una capa software intermedia hace creer a un sistema operativo que tiene hardware dedicado
- En realidad esto lo hemos visto antes (...)



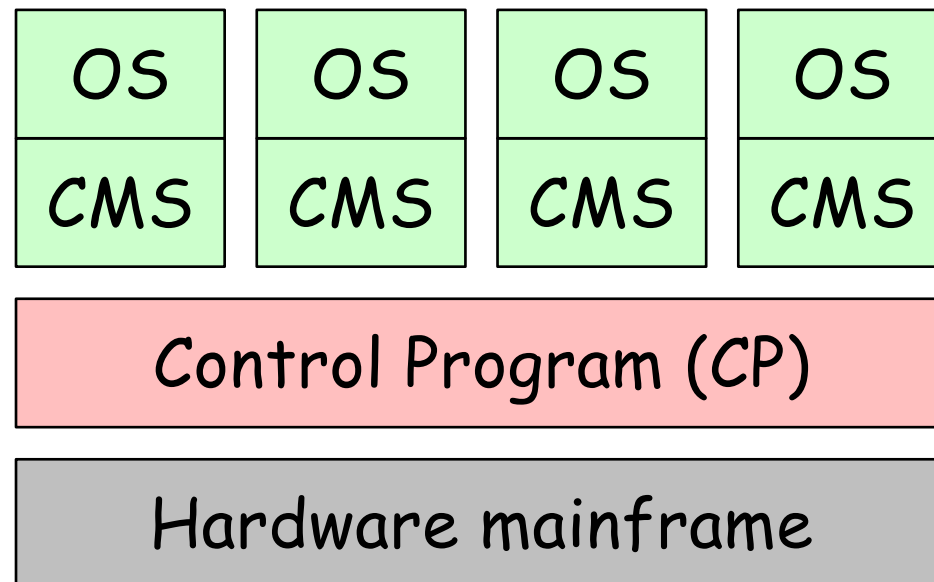
# ¿Virtualización?

- Es la misma idea detrás de las VLANs
- Los hosts de cada VLAN la ven como si estuvieran ellos solos en la LAN



# ¿Virtualización?

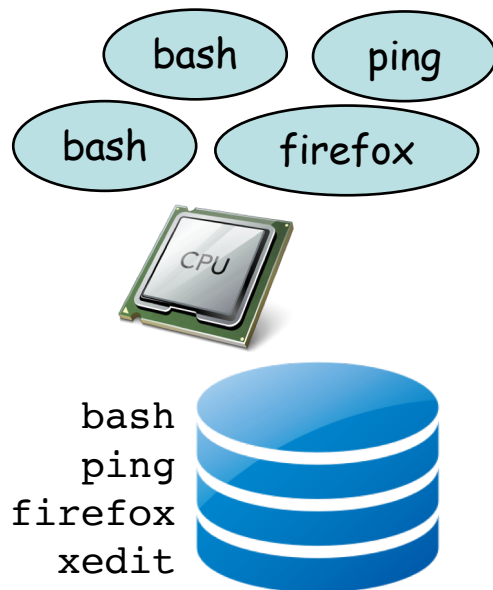
- En el entorno informático es un concepto muy antiguo
- A nivel de virtualización de sistemas operativos ya lo soportaban los mainframes en los 70s
- Comercialmente llega al entorno PC a principios de los 00s (VMware)



# Virtualización de memoria y multiproceso

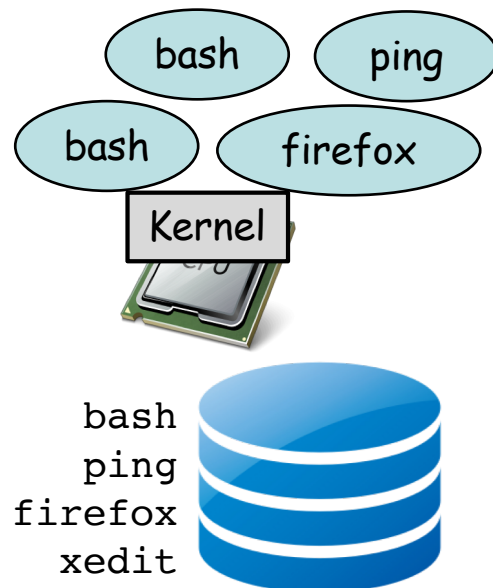
# Procesos

- Un proceso es un programa en ejecución
- /usr/bin/bash, /usr/bin/ping, /usr/bin/firefox, /usr/bin/xedit son ficheros en disco que contienen código ejecutable
- El Kernel crea una entidad de ejecución que llamamos "proceso"
- Tiene un hilo de ejecución (thread) así como otras estructuras asociadas en el kernel (fds, memoria, etc)
- Podemos estar ejecutando el mismo programa en varios procesos



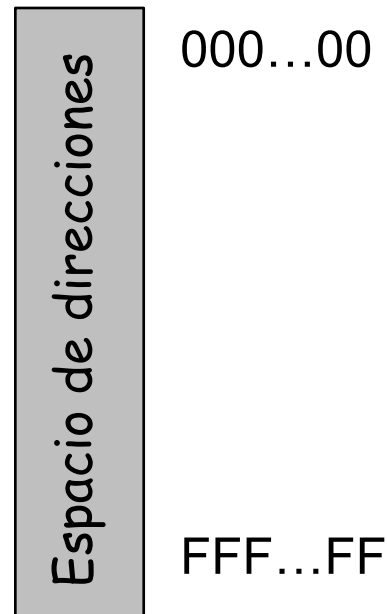
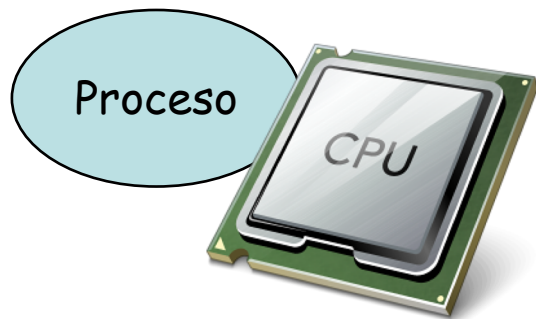
# Multiproceso: ¿Virtualización?

- Cuando varios procesos se ejecutan pero no disponemos de varias CPUs
- Cada proceso cree que dispone de la CPU pero se va alternando la ejecución entre procesos
- De nuevo se le está haciendo creer a alguien que dispone de ciertos recursos de forma exclusiva cuando no es así
- El Kernel del sistema operativo se encarga de gestionar el uso de los recursos físicos
- Gestiona pues el uso de la CPU y planifica la alternancia de los procesos en ejecución en la CPU



# Virtualización de memoria

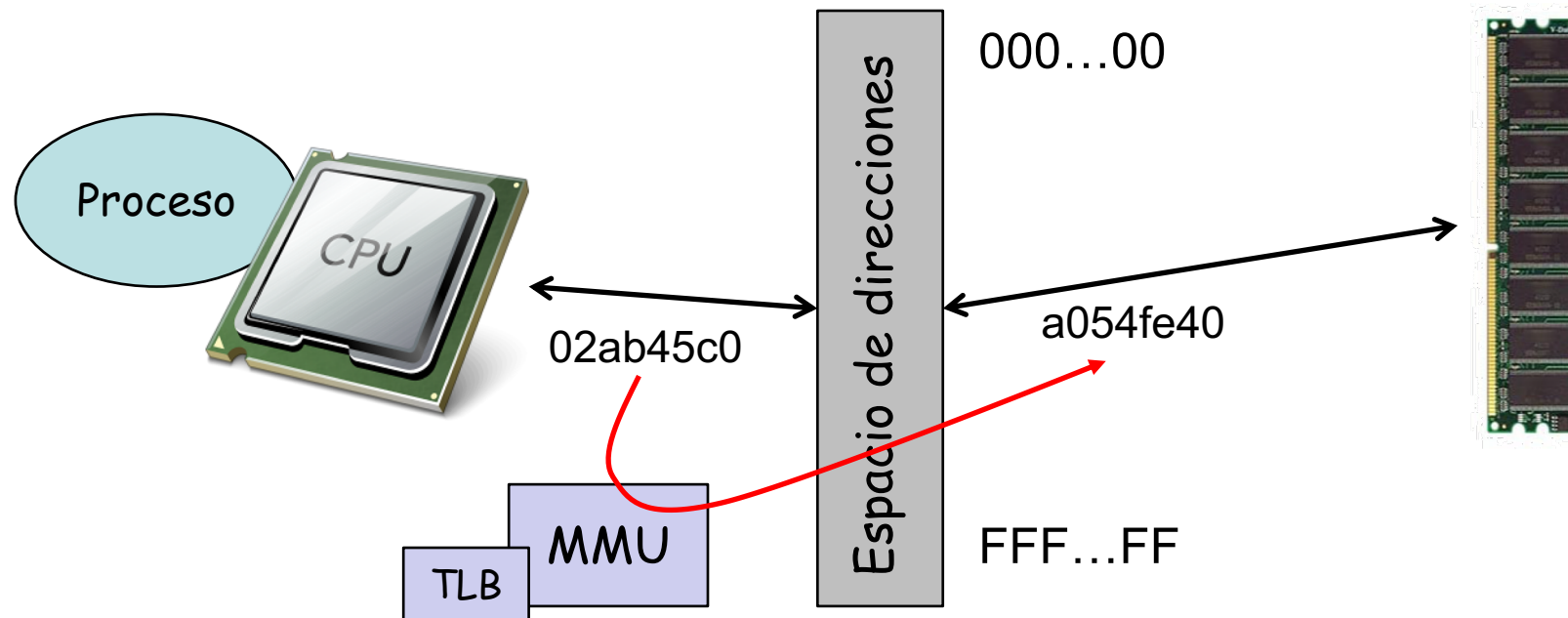
- El Kernel gestiona también el uso del recurso físico RAM
- Se puede conseguir que un proceso crea que dispone de toda la memoria
- De hecho podría ver más memoria que la existente
- Ve un espacio continuo de direcciones
- No puede acceder a la memoria en uso por otros procesos
- (...)





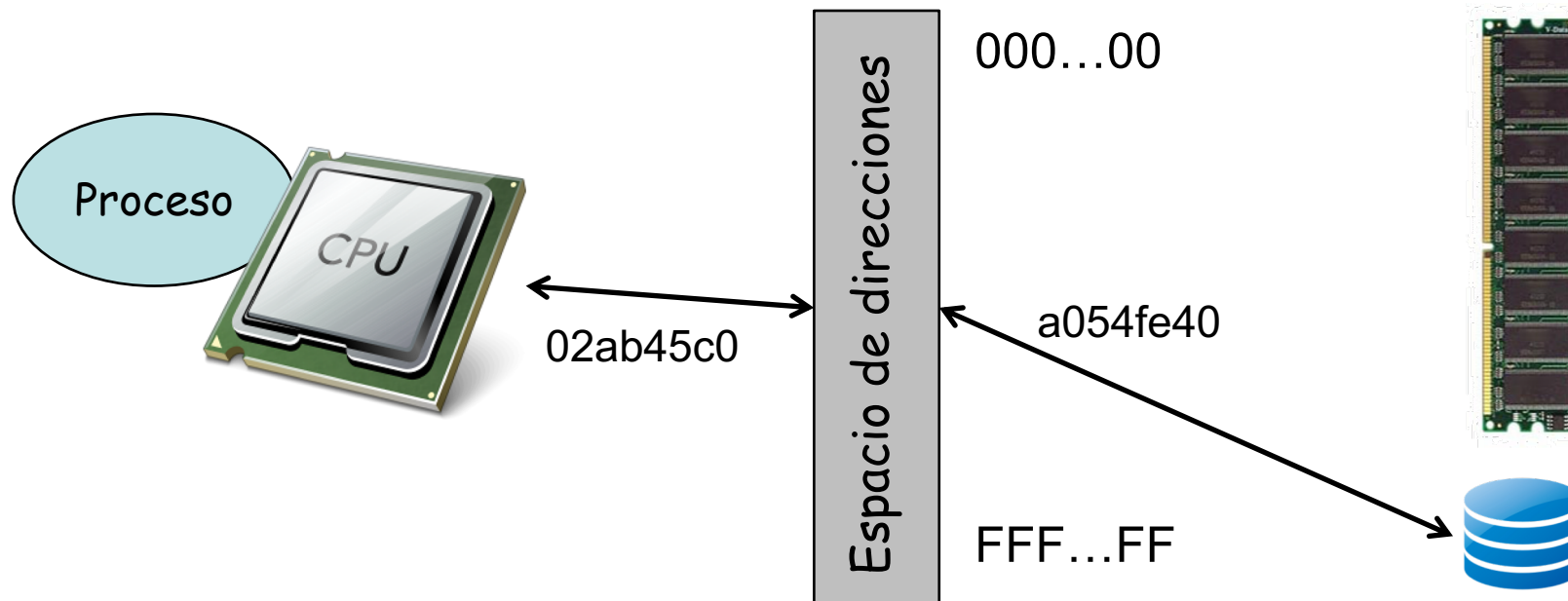
# Virtualización de memoria

- Cuando la CPU intenta acceder a una dirección de memoria se debe convertir la dirección *virtual* en la dirección física
- Con esa dirección física se puede acceder a la RAM (ignorando las posibles caches)
- Esto se hace por “páginas” (hoy suelen ser de 4 KiB)
- Esta conversión la hace la MMU (*Memory Management Unit*)
- Hoy en día es parte de la CPU
- Es decir, necesitamos (o al menos mejora el rendimiento) apoyo del hardware



# Virtualización de memoria

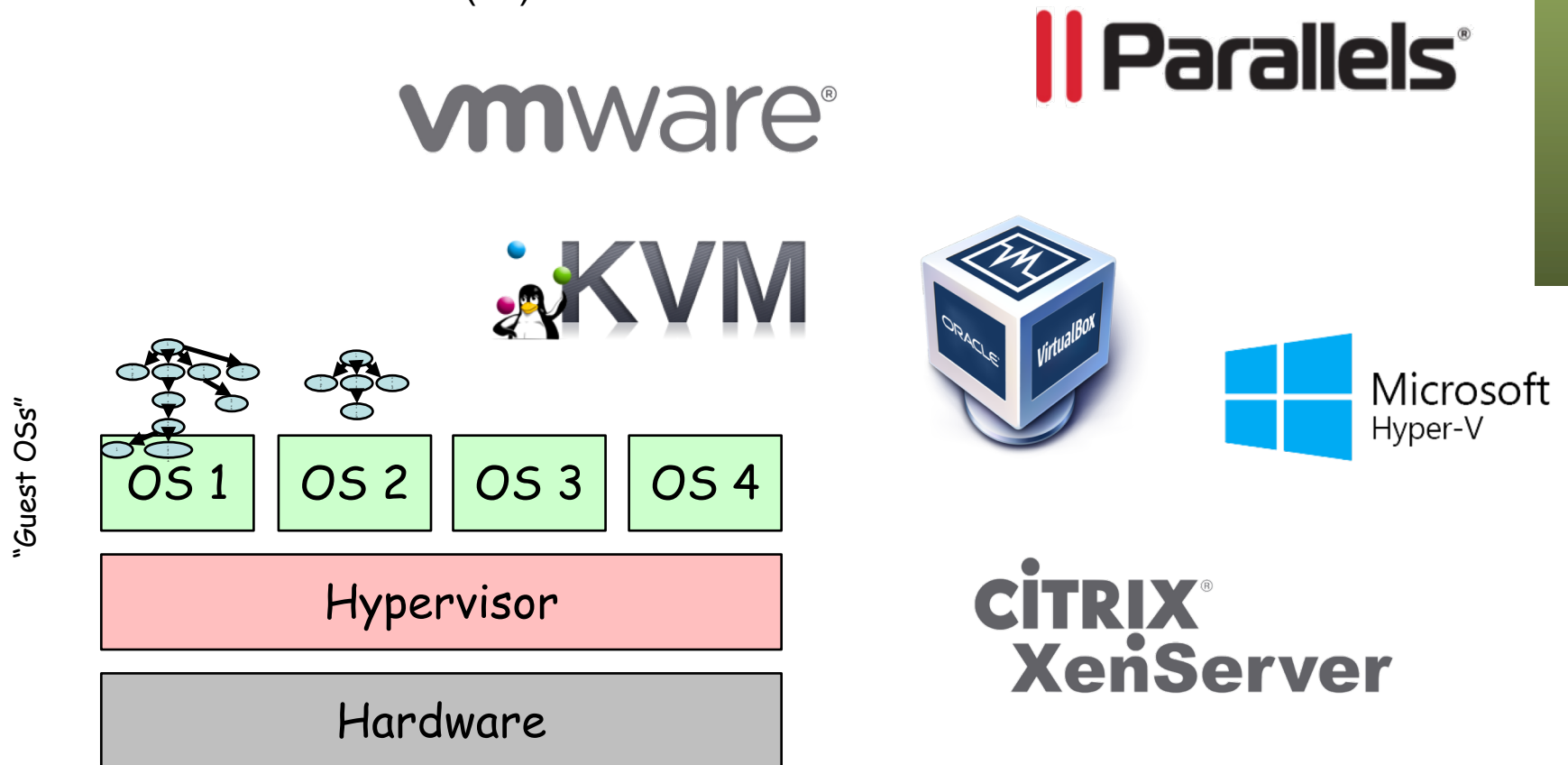
- El mapeo podría no llevar a memoria RAM sino a datos guardados en disco
- El disco es un dispositivo mucho más lento así que lo normal es mover los datos frecuentemente utilizados a RAM y los poco utilizados a disco



# El hypervisor

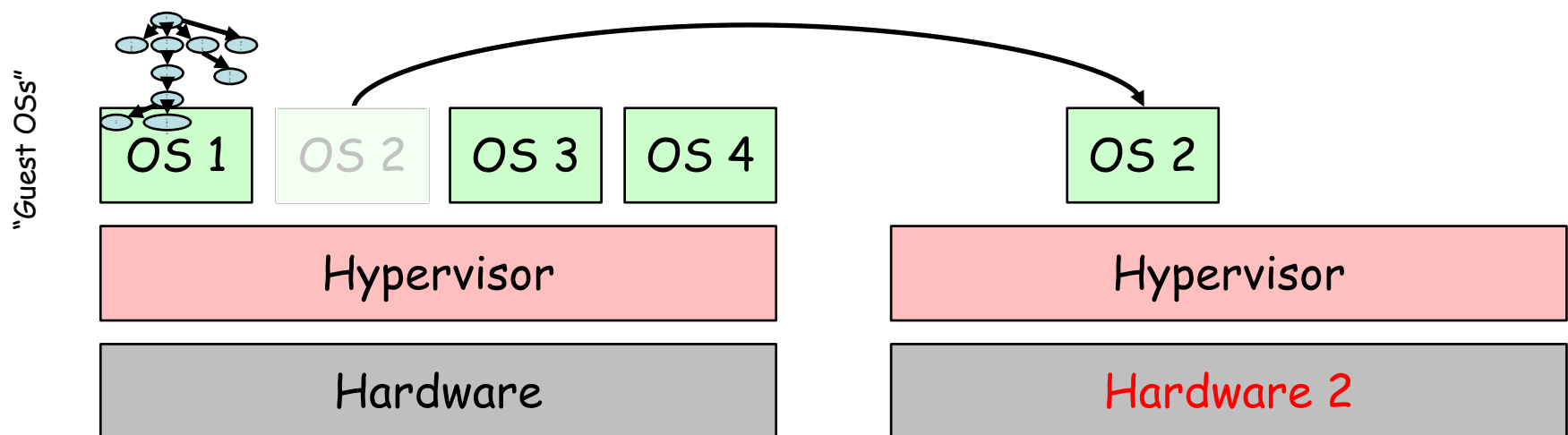
# Hypervisor

- Es una capa software entre el hardware y el sistema operativo “guest”
- También llamado “*Virtual Machine Monitor*” (VMM)
- Oculta el hardware real y puede presentar diferente hardware a cada máquina virtual
- Esas máquinas virtuales no necesitan cambios para funcionar en otro hypervisor aunque emplee un hardware diferente siempre que les presente el mismo hardware virtual (...)



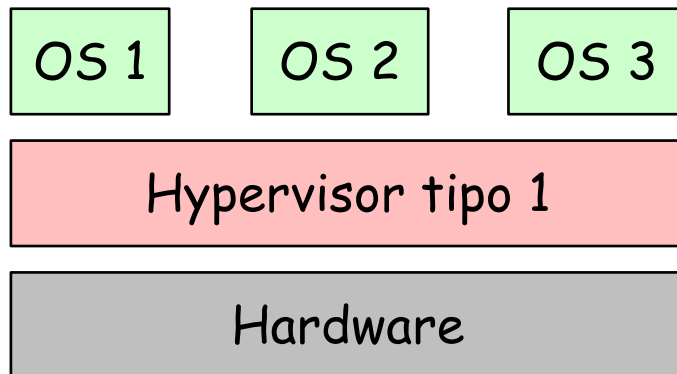
# Hypervisor

- Es una capa software entre el hardware y el sistema operativo “guest”
- También llamado “*Virtual Machine Monitor*” (VMM)
- Oculta el hardware real y puede presentar diferente hardware a cada máquina virtual
- Esas máquinas virtuales no necesitan cambios para funcionar en otro hypervisor aunque emplee un hardware diferente siempre que les presente el mismo hardware virtual
- La máquina virtual, todo su sistema operativo instalado y las aplicaciones, puede ser un solo fichero, sencillo de copiar a otra máquina
- Esto lo haremos normalmente con la VM apagada



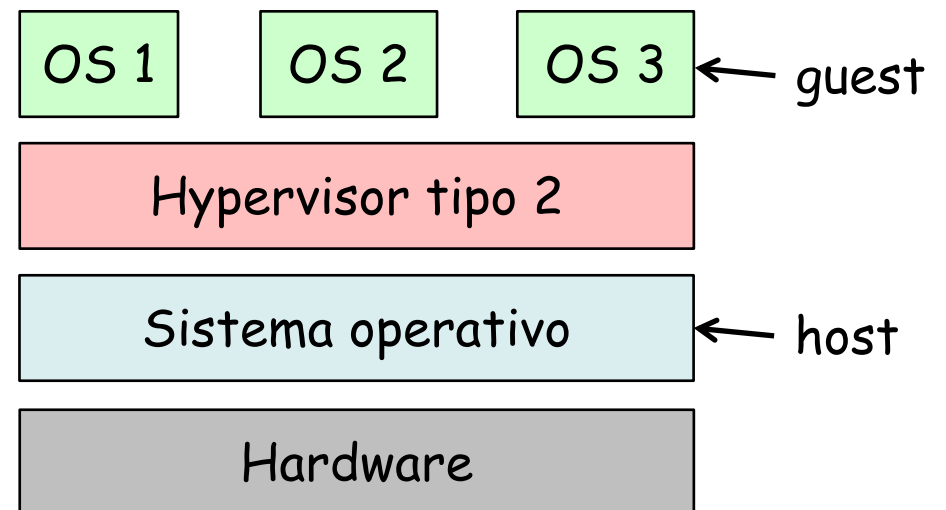
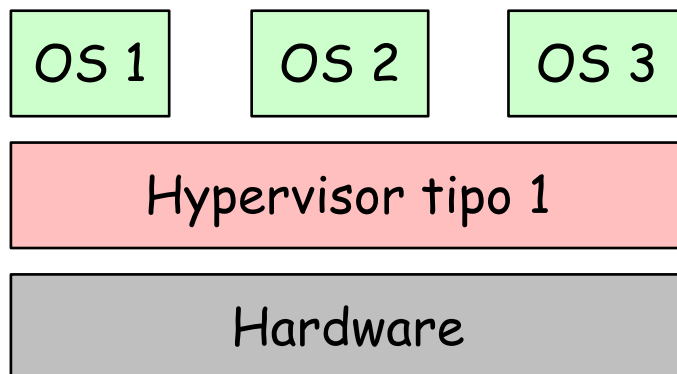
# Tipos de Hypervisores

- Tipo 1, nativo o “*bare-metal*”
  - Se ejecuta directamente sobre el hardware
  - Controla dicho hardware
  - Consume poco espacio y memoria
  - El mejor rendimiento potencial
  - El hypervisor debe contar con drivers para el hardware
  - Ejemplos: Citrix XenServer, Vmware ESXi, Microsoft Hyper-V, Linux KVM
- (...)



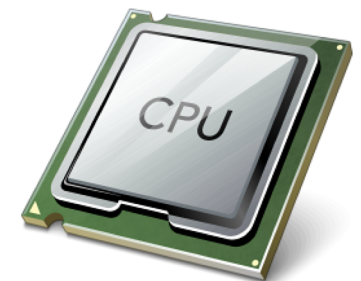
# Tipos de Hypervisores

- Tipo 2 o “hosted”
  - El hypervisor corre como una aplicación sobre un sistema operativo convencional
  - El sistema operativo guest sobre el hypervisor
  - El sistema operativo host tiene un impacto en el rendimiento
  - Es más frecuente la existencia de drivers para el hardware
  - Ejemplos: VMware Workstation, VMware Server, Microsoft Virtual PC, Parallels Workstation, VirtualBox, QEMU



# Virtualización de la CPU

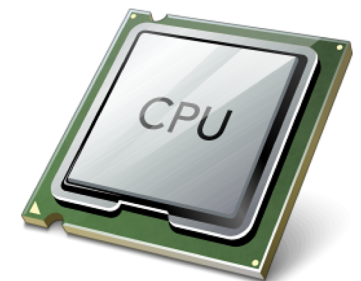
- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
  - Hace traducción (*on-the-fly*) de instrucciones (*binary translation*)
  - Se sustituyen las instrucciones no virtualizables por otras equivalentes
  - No requiere modificar el OS instalado
  - Ejemplos: VMware, Microsoft Virtual Server, Linux KVM, Parallels, VirtualBox, QEMU
- (...)





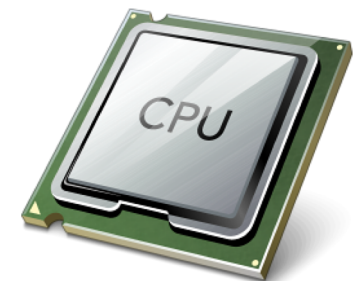
# Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
- *Paravirtualization (OS assisted virtualization)*
  - Se modifica el sistema operativo guest sustituyendo las instrucciones no virtualizables
  - Requiere menos sobrecarga en ejecución pero hay que poder modificar el código de ese sistema operativo guest
  - Ejemplos: Xen, VMware (VMTools), Virtualbox (additions), UML
- (...)



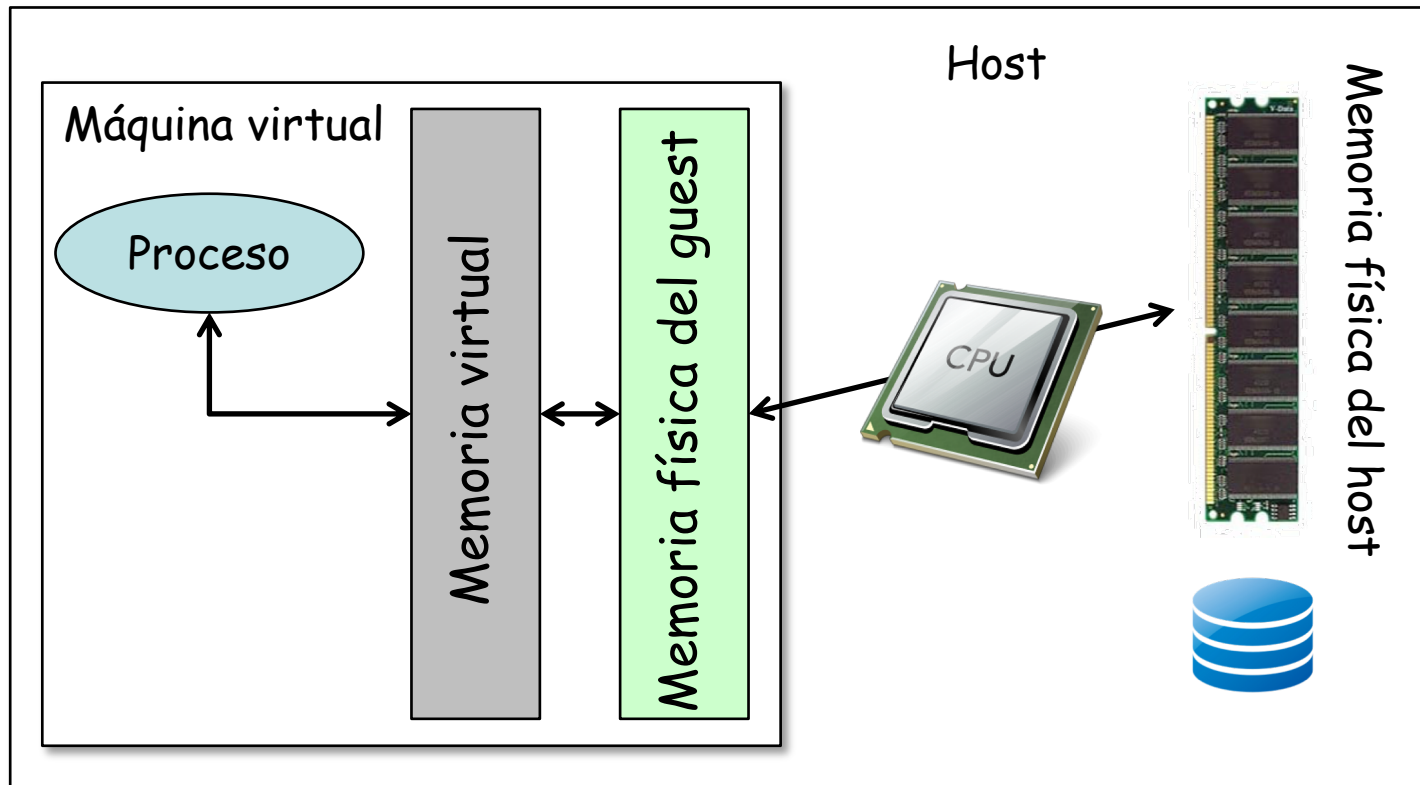
# Virtualización de la CPU

- El kernel de un sistema operativo está pensado para ejecutarse con máximos privilegios
- Ciertas instrucciones de la CPU no son sencillas de virtualizar y no se pueden dejar ejecutar a un proceso
- *Full virtualization*
- *Paravirtualization (OS assisted virtualization)*
- *Hardware-assisted virtualization*
  - El hardware se encarga de la traducción de instrucciones privilegiadas
  - Requiere soporte por el hardware (Intel VT-x, AMD-V)
  - Ejemplos: VMware, Microsoft, Parallels, Xen, Virtualbox



# Virtualización de RAM (doble)

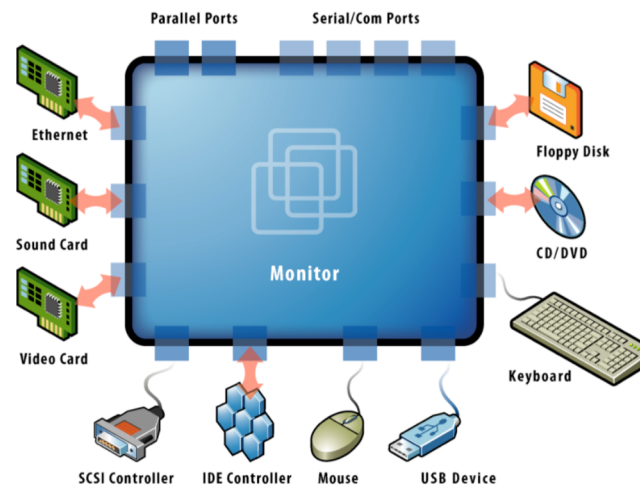
- El sistema operativo guest emplea memoria virtual y la mapea a lo que él cree que es memoria física
- Eso no puede ser la auténtica memoria física, así que debe ser de nuevo mapeada
- *Shadow page tables* para hacerlo por software o *nested paging* (Second Level Address Translation) por hardware si lo soporta la CPU
- Hay que virtualizar la MMU



# Virtualización de dispositivos

# Virtualización de dispositivos

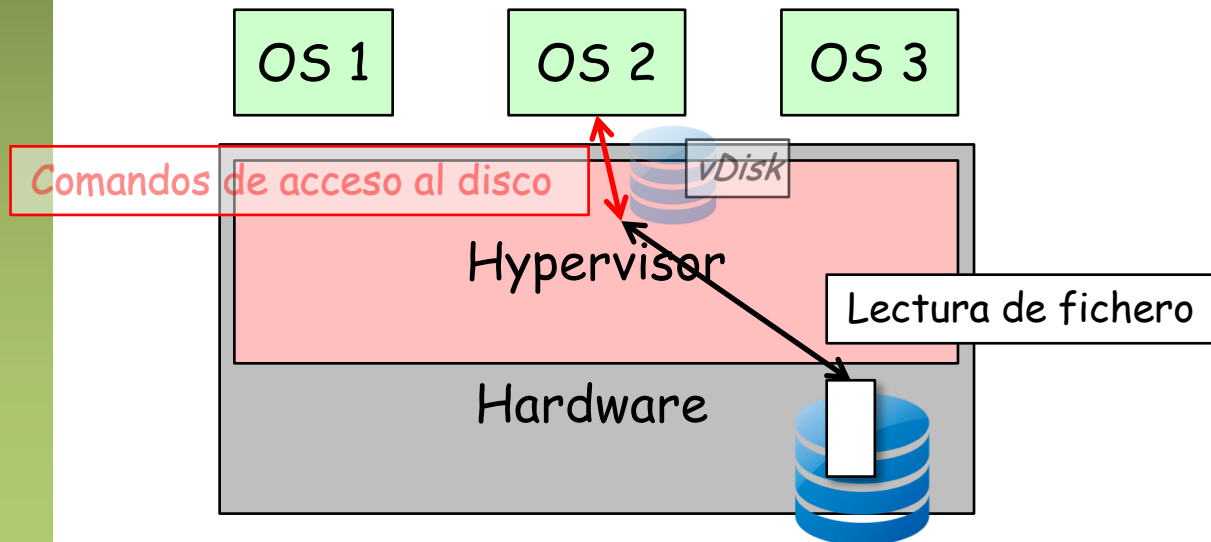
- El VMM presenta a la VM unos dispositivos comunes, de forma que sean fácilmente soportados
- Puede tener varias opciones, por ejemplo ofrecerle al guest diferentes modelos de tarjeta de red
- El hardware puede tener soporte para ser virtualizado



# Acceso a disco

# Acceso a disco desde la VM

- El hypervisor ofrece un HD virtual al guest que responde a algún tipo de interfaz de comunicación con discos
- Por ejemplo un interfaz SCSI (más sobre esto más adelante)
- Son mensajes de lectura y escritura en bloques del disco
- De la máquina virtual se reciben los comandos y los bloques a leer se mapean por ejemplo en bloques de un fichero en disco del host
- Volveremos al acceso a disco en los servidores ...

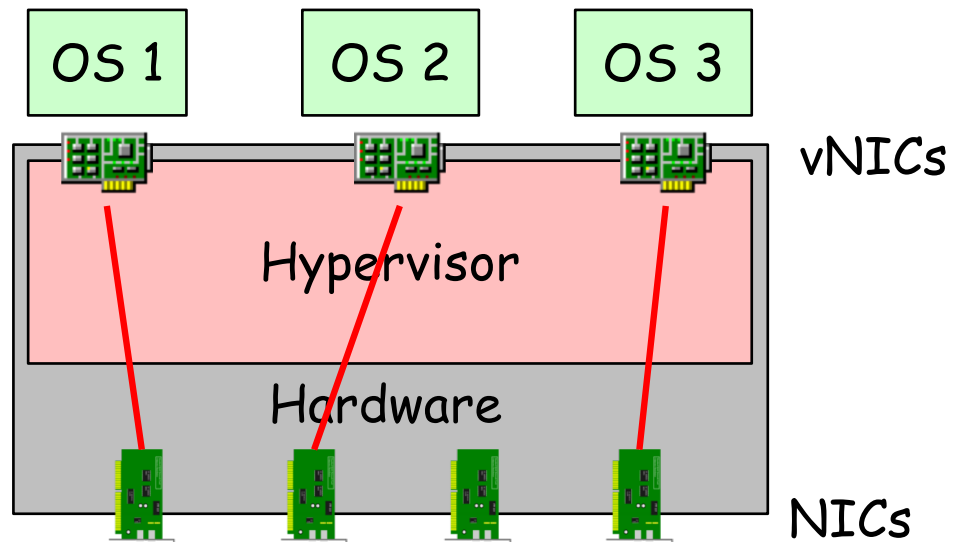


# Networking con VMs



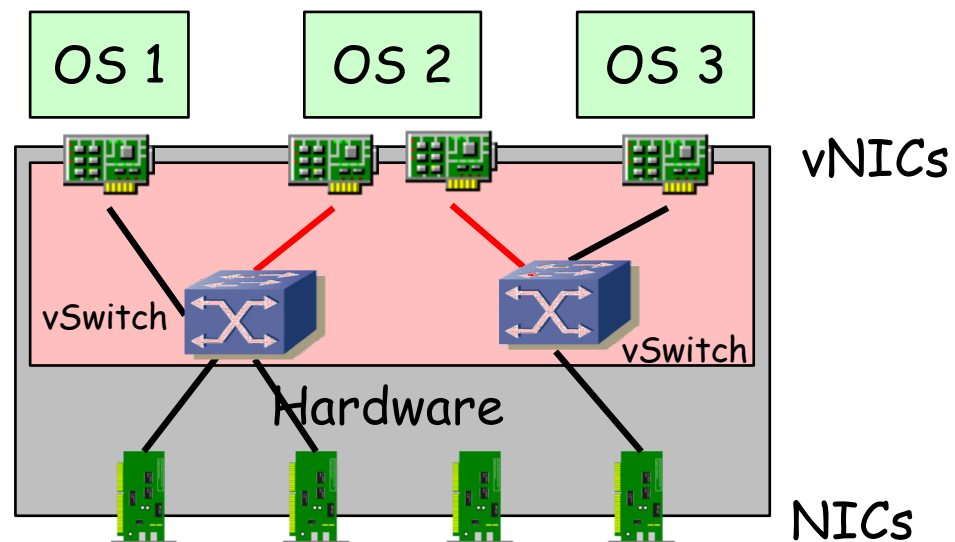
# Virtual NIC

- Las NICs reales pueden ser de diferentes modelos que las virtuales
- Puede haber una relación 1:1 entre NIC y vNIC
- (...)



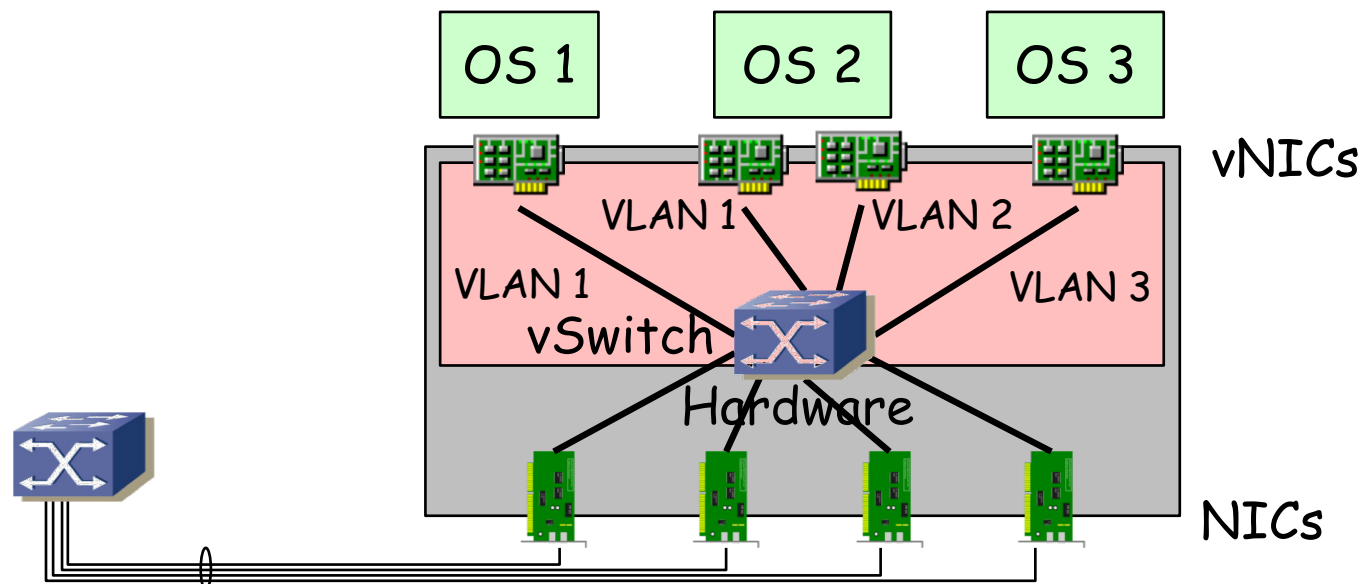
# Virtual NIC

- Las NICs reales pueden ser de diferentes modelos que las virtuales
- Puede haber una relación 1:1 entre NIC y vNIC
- Puede implementarse un conmutador Ethernet en software
- Se suele llamar un vSwitch o VEB (Virtual Ethernet/Embedded Bridge)
- La dirección MAC de la vNIC suele ser diferente de la MAC de la NIC
- OUI reservado para la empresa desarrolladora del hypervisor
- Puede haber varias vNICs en la misma VM
- Puede haber varios vSwitches

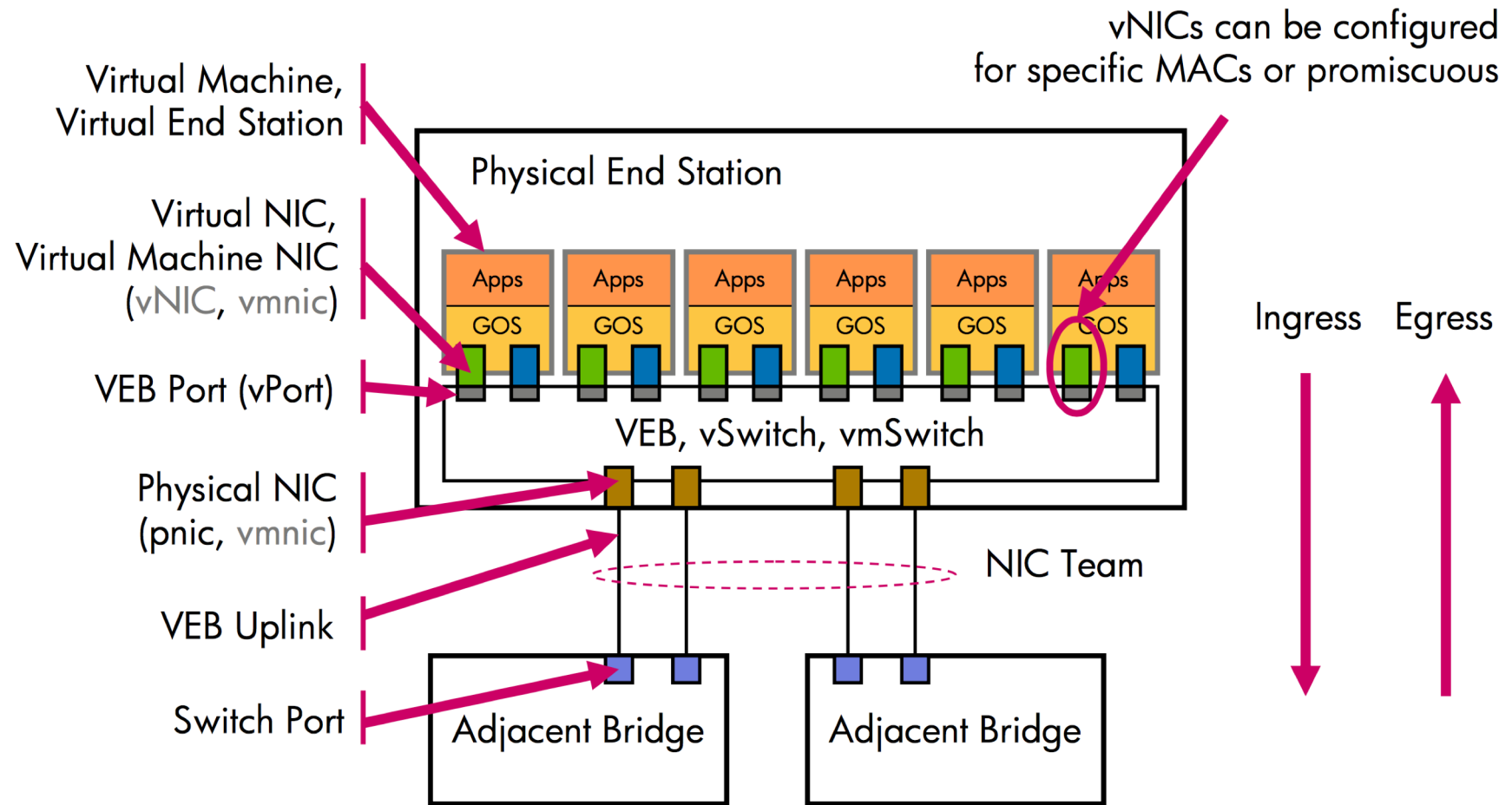


# Virtual Switch

- Se pueden asignar los puertos a VLANs diferentes
- Las NICs soportan 802.1Q
- Y agregación (802.3ad) o *NIC teaming*
- El vSwitch tiene más información sobre los hosts que la que puede tener un puente hardware (sabe sus MACs sin usar aprendizaje)
- Puede estar implementado enteramente en software o parte en hardware (normalmente funcionalidades en la NIC)
- Puede estar desarrollado junto con el hypervisor o por otra empresa y así gestionarse como parte del entorno de virtualización o de red

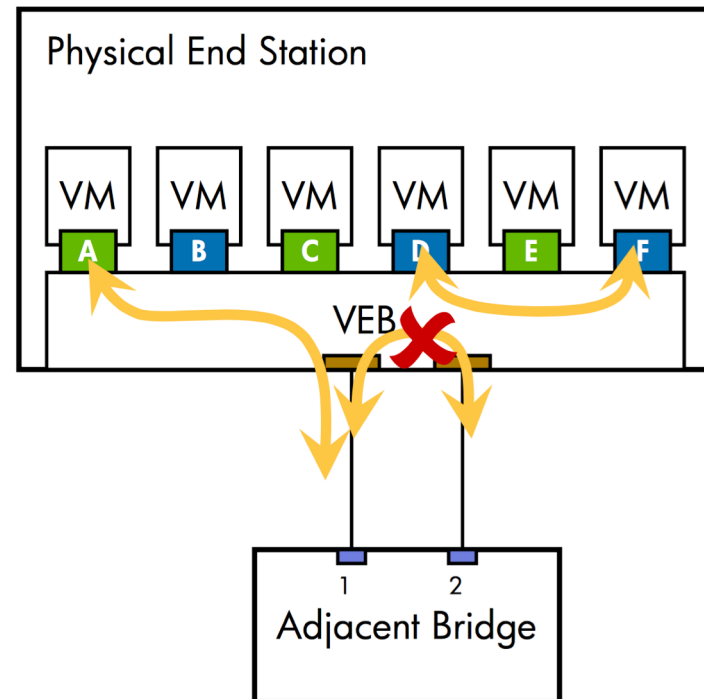


# Virtual Switch



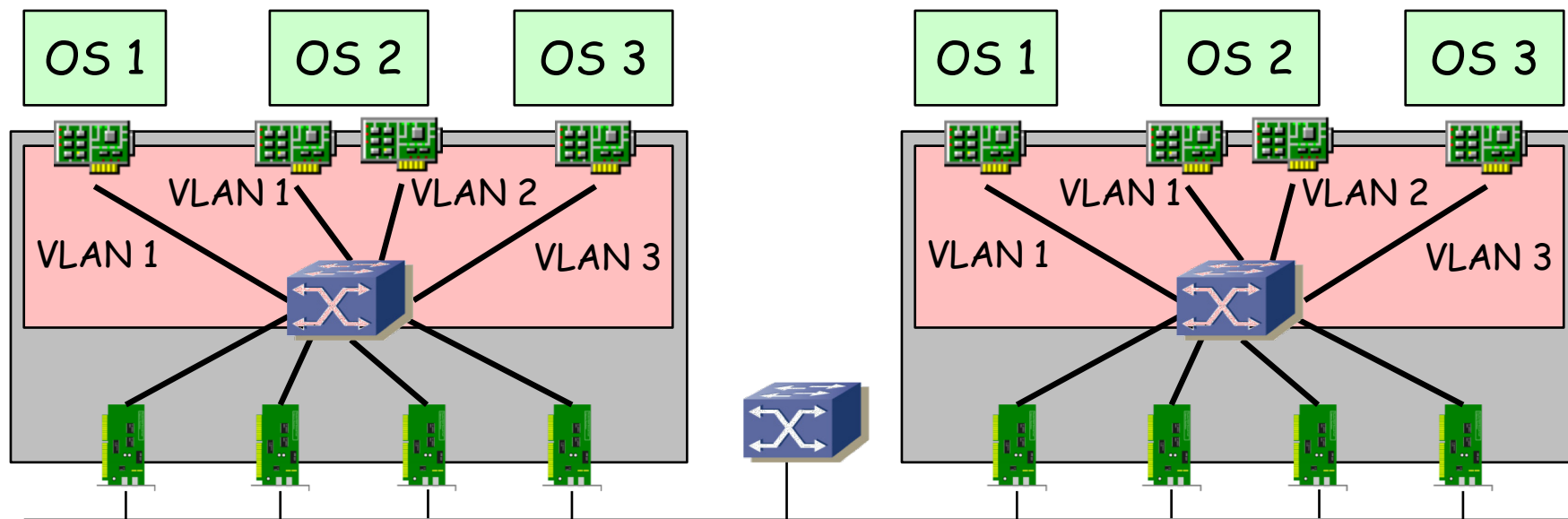
# Virtual Switch

- No reenvía entre los puertos hacia la infraestructura de red
- No participa en el STP
- No necesita hacer aprendizaje, solo tiene las MACs de las VMs estáticamente y el resto debe estar en el exterior
- Pero hay que configurar políticas en sus puertos lógicos
- Probablemente no tenga las funcionalidades de un switch físico (QoS, ACLs, etc)
- ¿De quién es la gestión?



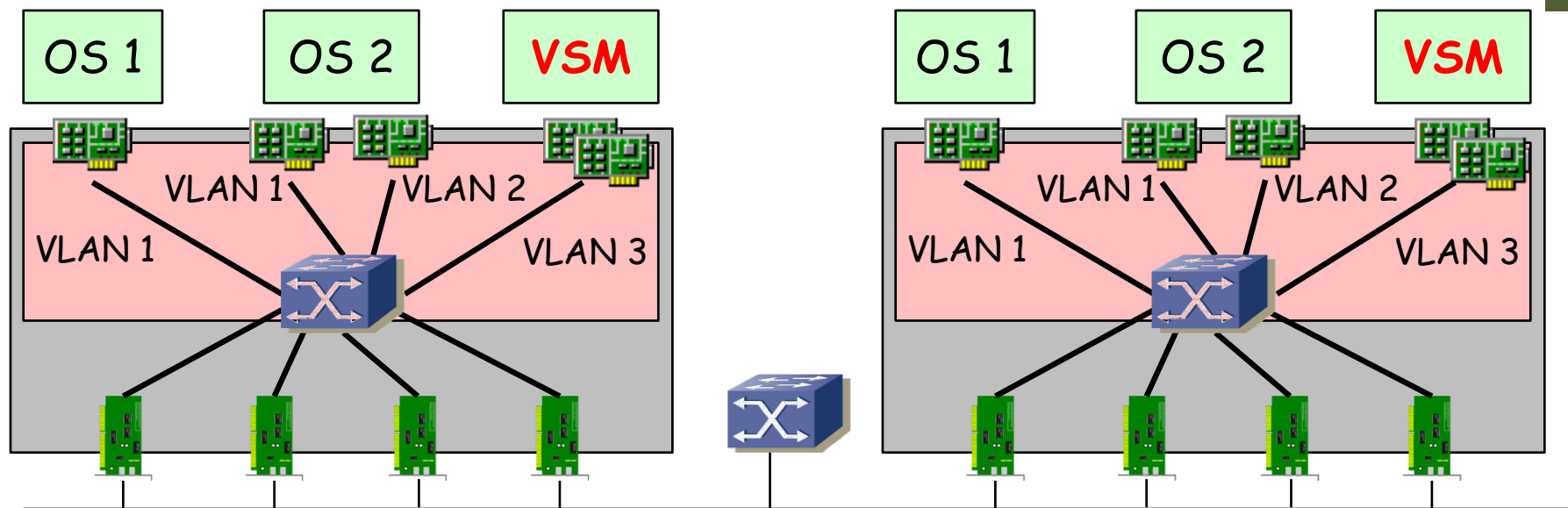
# Virtual Switch

- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- ¿Y el supervisor? (...)



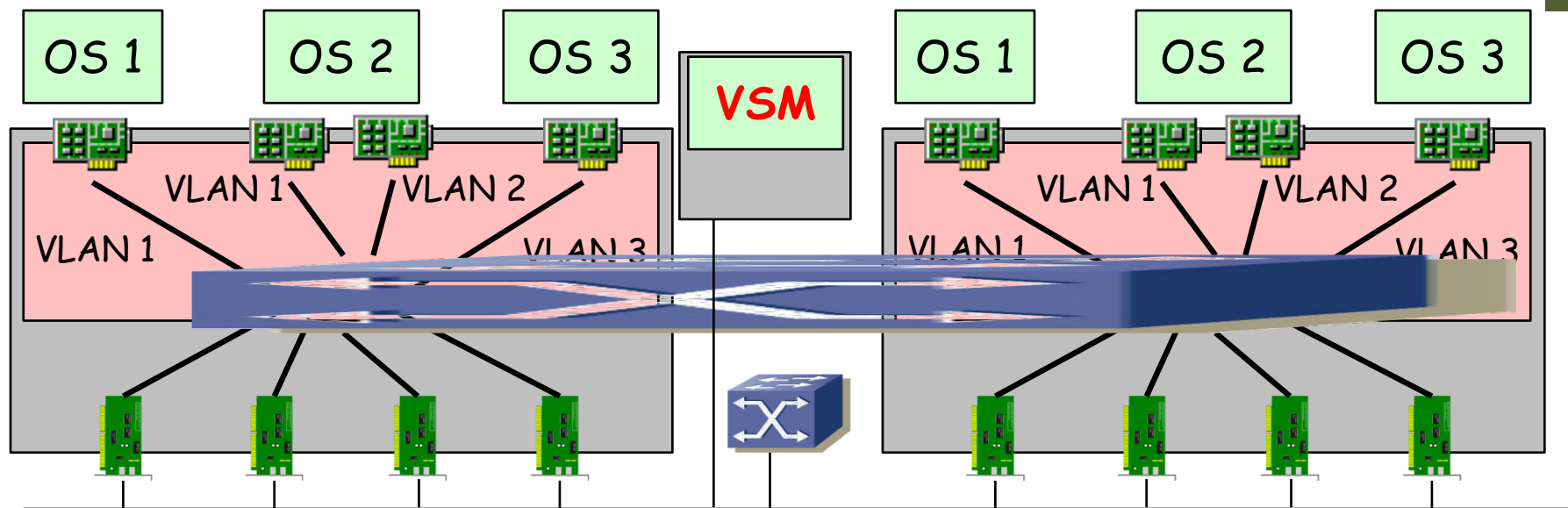
# Virtual Switch

- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- El supervisor corre como una máquina virtual (Ej: Cisco 1000v)
- (...)



# Virtual Switch

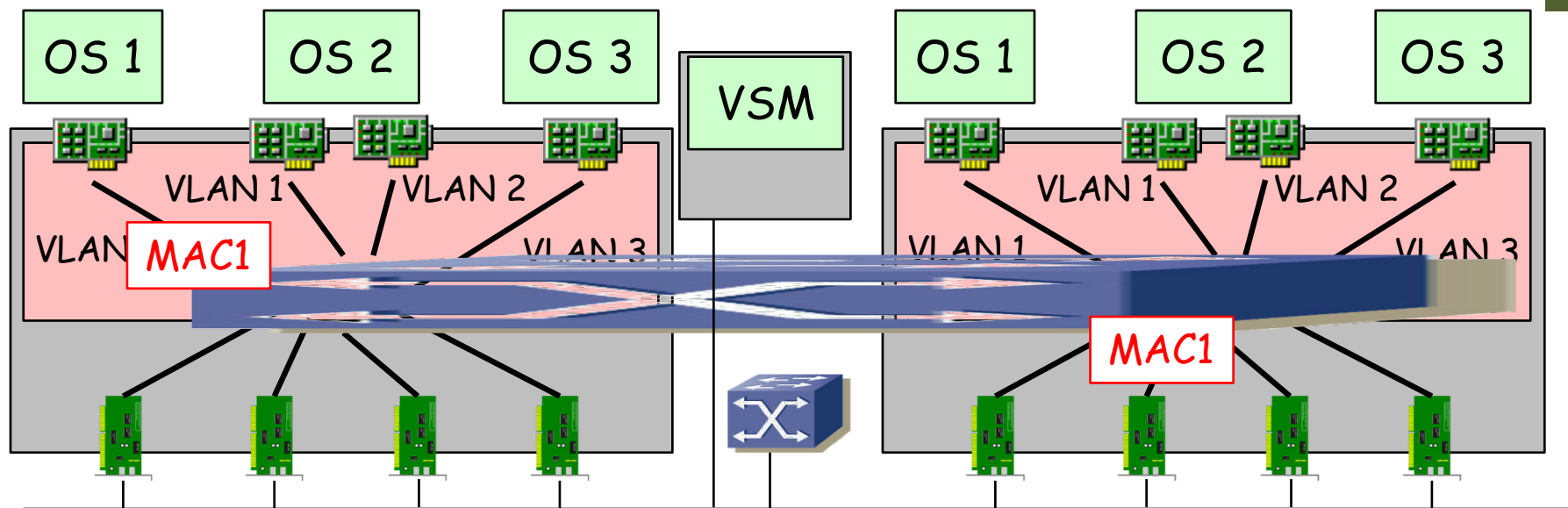
- Este virtual switch puede estar compuesto, igual que uno hardware de:
  - Módulo controlador/supervisor virtual (plano de control)
  - Módulos con los puertos Ethernet virtuales (plano de datos)
- En ese caso, el elemento en cada host es el módulo de puertos
- El supervisor corre como una máquina virtual (Ej: Cisco 1000v)
- Vale con un supervisor para controlar varios hosts y entonces es como si todos formaran un switch virtual
- Ese supervisor podría correr en su propio hardware (Ej: Cisco 1100)





# Virtual Switch

- Cada host mantiene su propia tabla de reenvío
- El switch de un host no tiene conocimiento de las MACs aprendidas en otro, ni aunque formen parte del mismo switch distribuido
- Es decir, aunque hablemos de un switch distribuido NO hay una base de datos de filtrado única
- Eso quiere decir que una dirección MAC puede aparecer más de una vez, dado que puede aparecer en todas las tablas de host



# Caso de prácticas

# VirtualBox



- Hypervisor de tipo 2



The screenshot shows the VirtualBox website homepage. At the top left is the VirtualBox logo. The main heading is 'VirtualBox' in a large blue font. Below it is the text 'Welcome to VirtualBox.org!'. The main content area contains three paragraphs of text describing the product, its supported operating systems, and its development status. A large blue button with white text says 'Download VirtualBox 6.1'. Below this is a 'Hot picks' section with three bullet points. At the bottom, there is the Oracle logo and a footer with links for 'Contact', 'Privacy policy', and 'Terms of Use'.

**VirtualBox**

**Welcome to VirtualBox.org!**

VirtualBox is a powerful x86 and AMD64/Intel64 [virtualization](#) product for enterprise as well as home use. Not only is VirtualBox an extremely feature rich, high performance product for enterprise customers, it is also the only professional solution that is freely available as Open Source Software under the terms of the GNU General Public License (GPL) version 2. See "[About VirtualBox](#)" for an introduction.

Presently, VirtualBox runs on Windows, Linux, Macintosh, and Solaris hosts and supports a large number of [guest operating systems](#) including but not limited to Windows (NT 4.0, 2000, XP, Server 2003, Vista, Windows 7, Windows 8, Windows 10), DOS/Windows 3.x, Linux (2.4, 2.6, 3.x and 4.x), Solaris and OpenSolaris, OS/2, and OpenBSD.

VirtualBox is being actively developed with frequent releases and has an ever growing list of features, supported guest operating systems and platforms it runs on. VirtualBox is a community effort backed by a dedicated company: everyone is encouraged to contribute while Oracle ensures the product always meets professional quality criteria.

**Download VirtualBox 6.1**

**Hot picks:**

- Pre-built virtual machines for developers at [Oracle Tech Network](#)
- **Hyperbox** Open-source Virtual Infrastructure Manager [project site](#)
- **phpVirtualBox** AJAX web interface [project site](#)

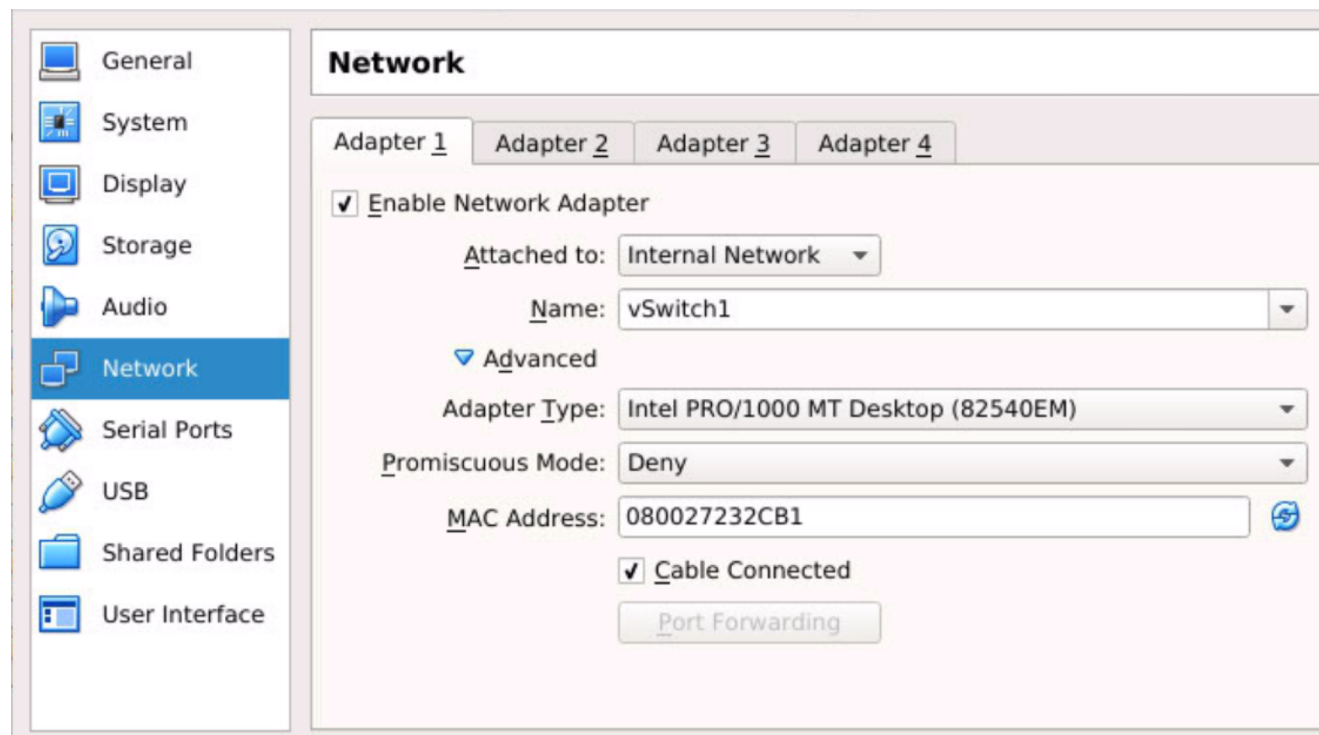
**ORACLE**

[Contact](#) – [Privacy policy](#) – [Terms of Use](#)

# VirtualBox



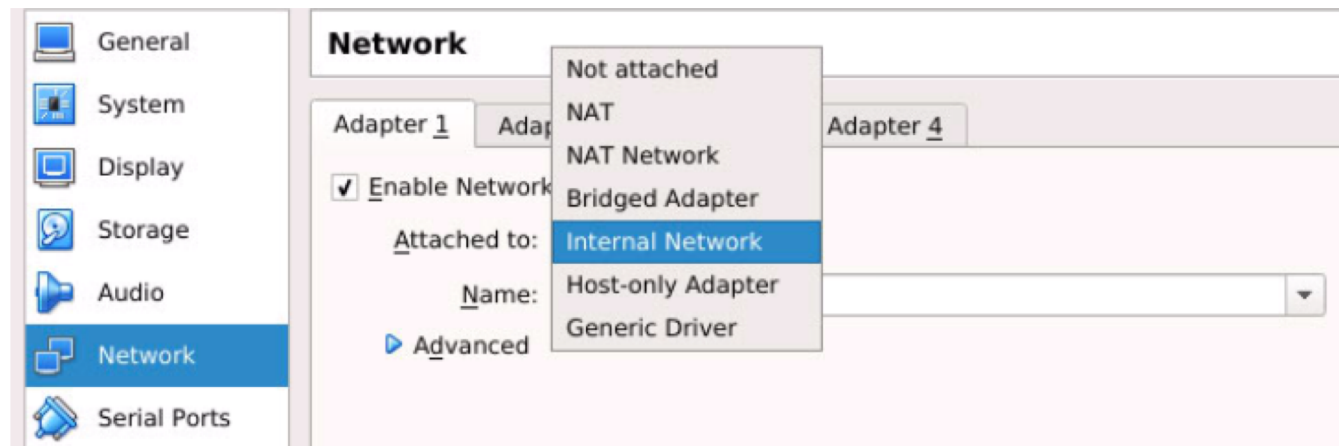
- Cada VM puede tener varias vNICs (4 desde el GUI)
- Puede virtualizar diferentes modelos de NIC
- Podemos especificar la dirección MAC (cuidad con tener varias VMs con la misma dirección MAC)



# VirtualBox



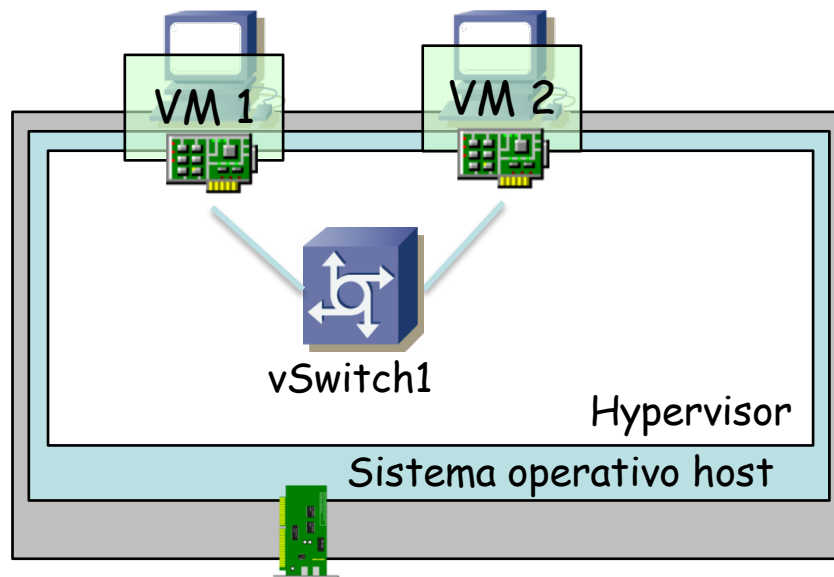
- Nos interesa en especial todo lo relacionado con interconexión
- Cada vNIC puede estar conectada en un escenario de red diferente
  - Not attached
  - NAT
  - NAT Network
  - Bridged Adapter
  - Host-only Adapter
  - Generic Driver



# Internal Network



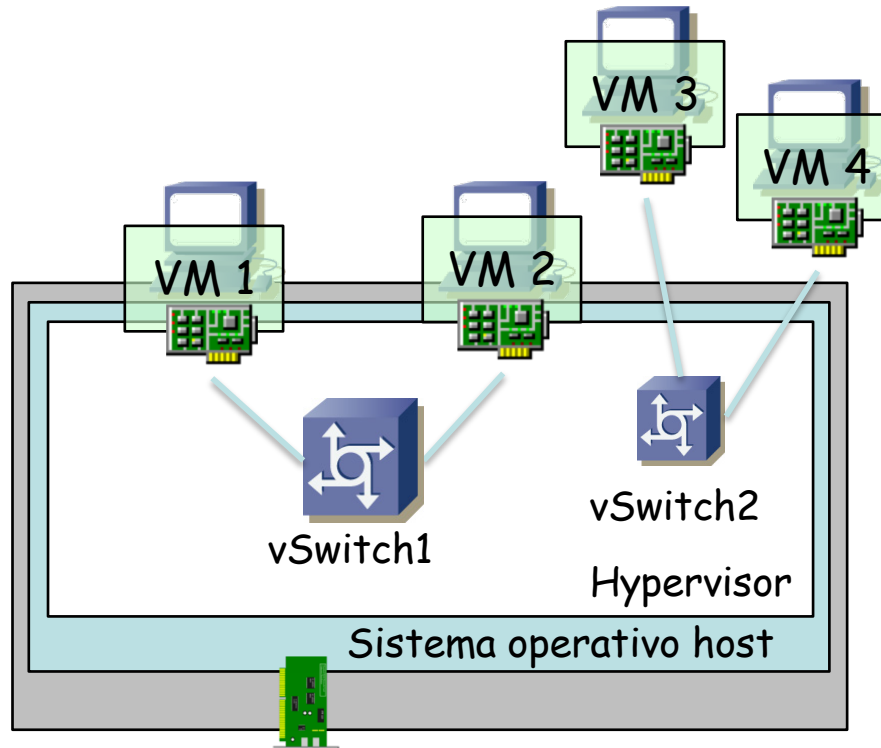
- Comunicación entre las VMs mediante vSwitch
- No con el exterior ni con el host
- Podemos emplear IP o cualquier otro protocolo sobre Ethernet



# Internal Network



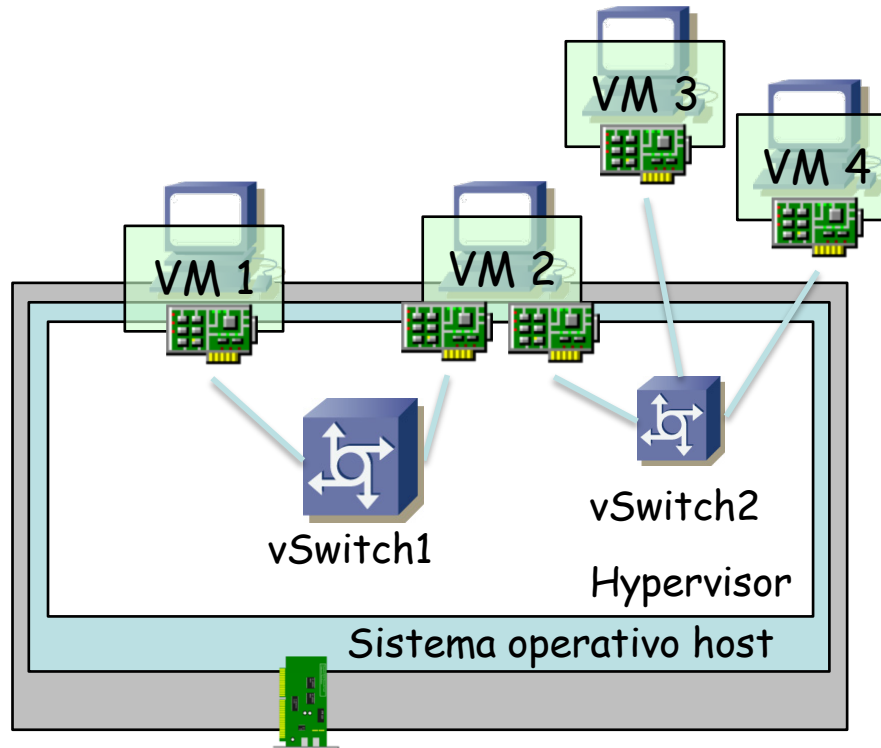
- Podemos crear varias Internal Networks
- Aisladas entre ellas
- (...)



# Internal Network



- Podemos crear varias Internal Networks
- Aisladas entre ellas
- Una VM podría tener varias vNICs
- Podrían estar conectadas a diferentes internal networks
- (...)

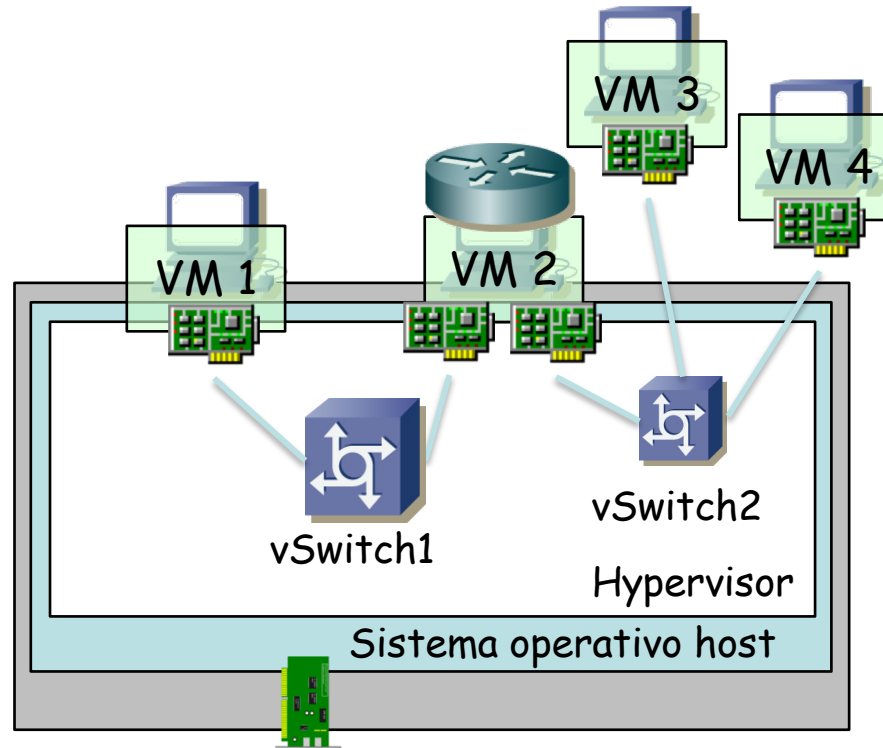




# Internal Network



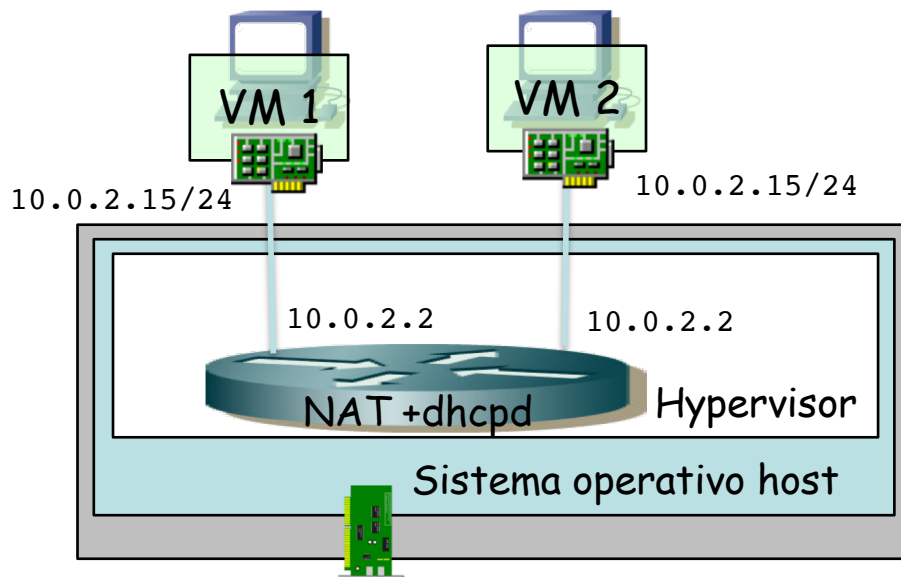
- Podemos crear varias Internal Networks
- Aisladas entre ellas
- Una VM podría tener varias vNICs
- Podrían estar conectadas a diferentes internal networks
- La VM podría actuar como router entre subredes IP en cada LAN construida con un vSwitch



# NAT



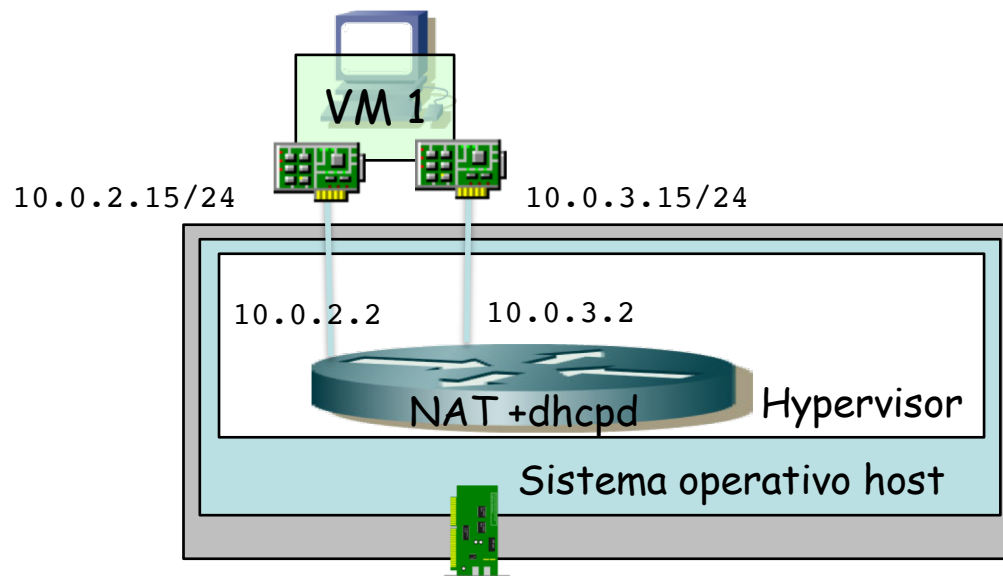
- VMs acceden al exterior a través de un NAT en el hypervisor
- Si hay varias VMs con interfaz en NAT les ofrece la misma dirección IP
- Va a saber a qué VM reconducir el tráfico por la tabla dinámica del NAT
- El tráfico al exterior sale del NAT con la dirección MAC origen del host



# NAT



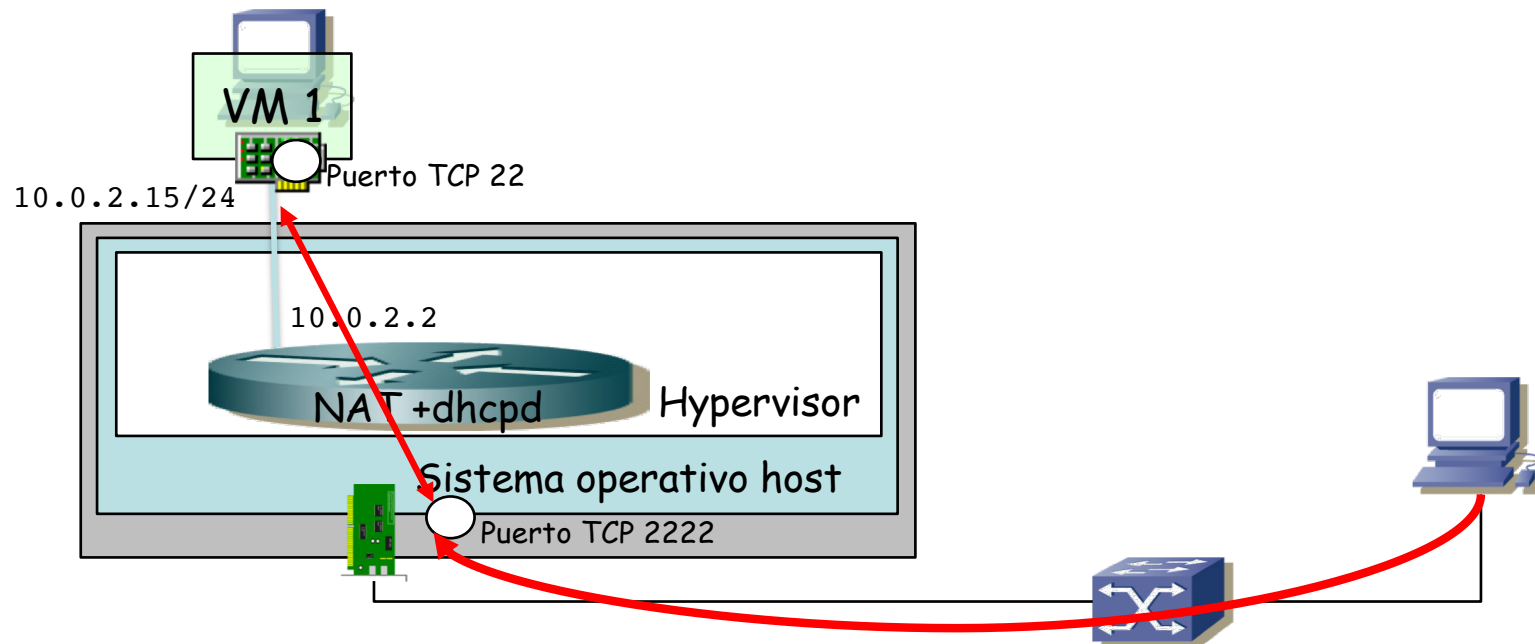
- VMs acceden al exterior a través de un NAT en el hypervisor
- Si hay varias VMs con interfaz en NAT les ofrece la misma dirección IP
- Va a saber a qué VM reconducir el tráfico por la tabla dinámica del NAT
- El tráfico al exterior sale del NAT con la dirección MAC origen del host
- Una VM podría tener dos interfaces en NAT, entonces sí los pone en diferentes subredes



# NAT



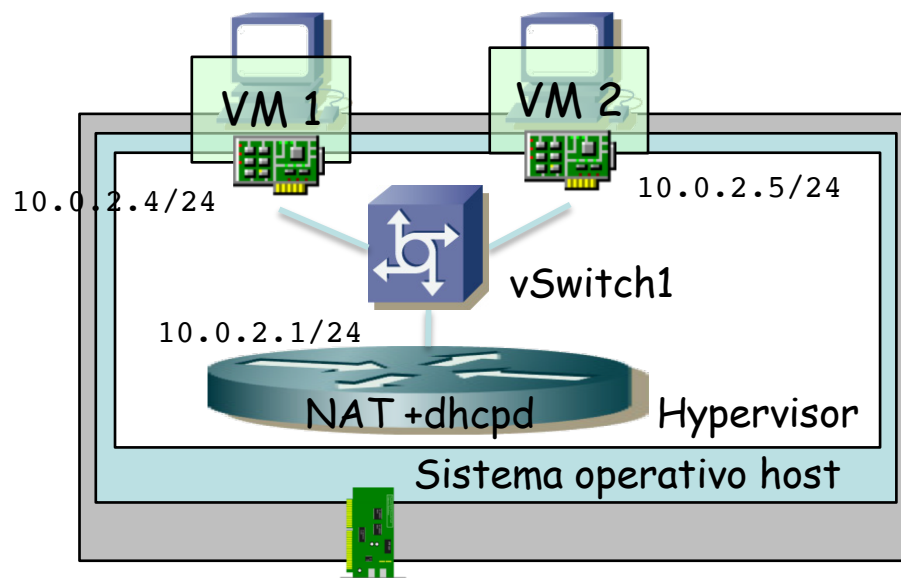
- Podemos introducir reglas de redirección (*Port Forwarding*)



# NAT Network



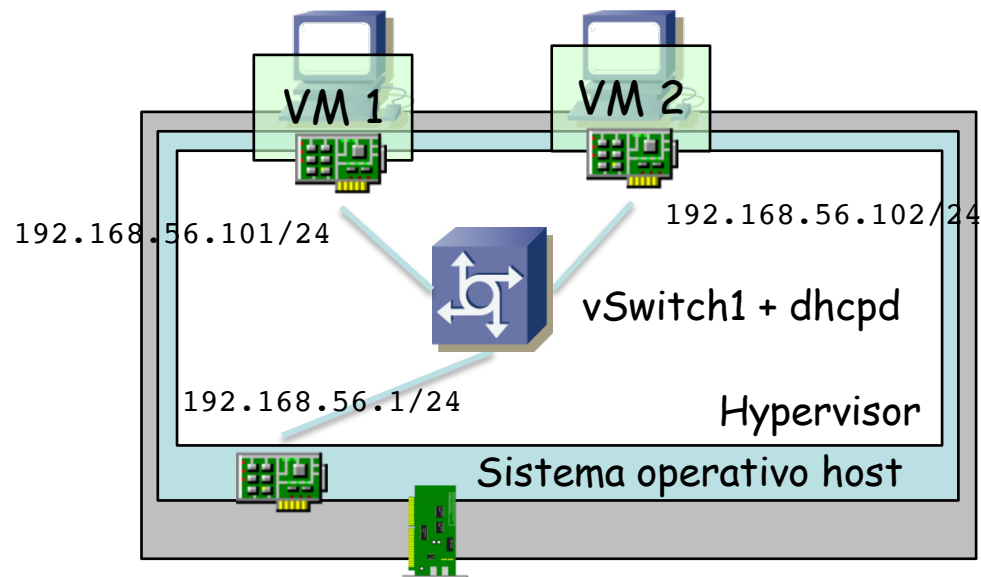
- En este caso las VMs unidas a la misma “Nat Network” están conectadas a una “Internal Network” en la cuál está también el NAT
- Permite la comunicación entre las VMs
- Igualmente se puede hacer *Port Forwarding*



# Host only



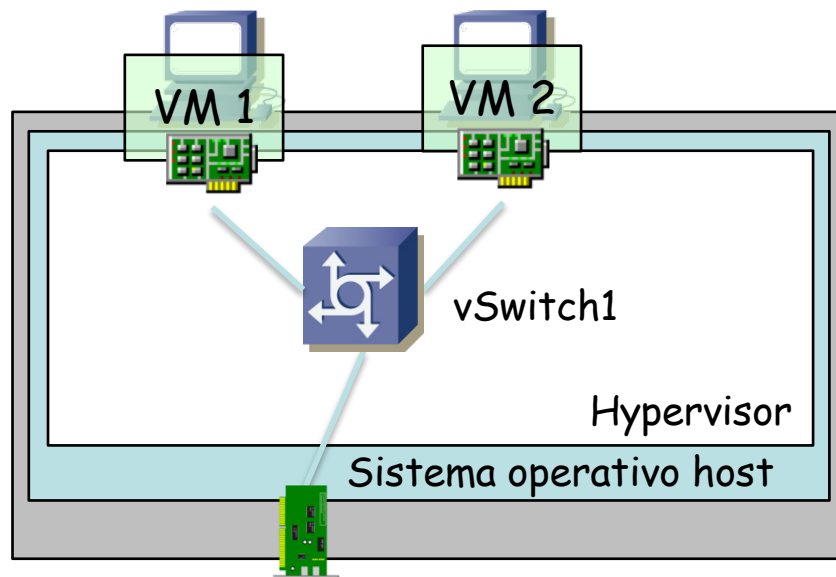
- Comunicación entre las VMs
- Comunicación con el host (crea un interfaz lógico en él)
- No con el exterior



# Bridged network



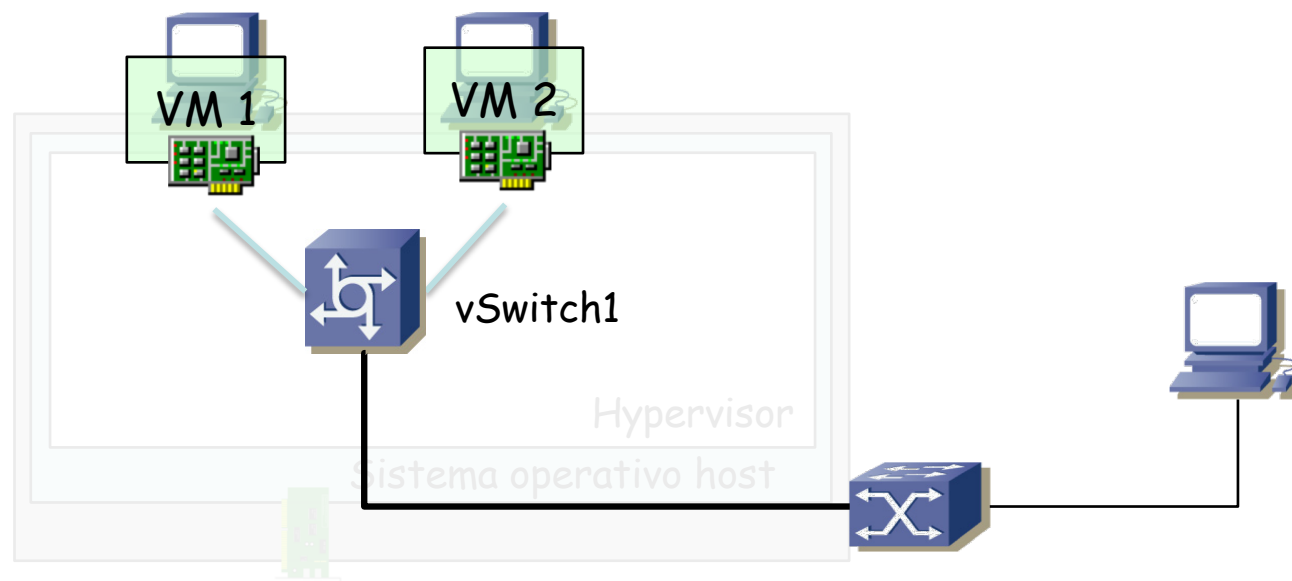
- Puentea con la NIC física (si hay varias se puede elegir cuál)
- Las tramas Ethernet salen con dirección MAC origen de la VM



# Bridged network



- Puentea con la NIC física (si hay varias se puede elegir cuál)
- Las tramas Ethernet salen con dirección MAC origen de la VM
- De cara al exterior son hosts en la LAN

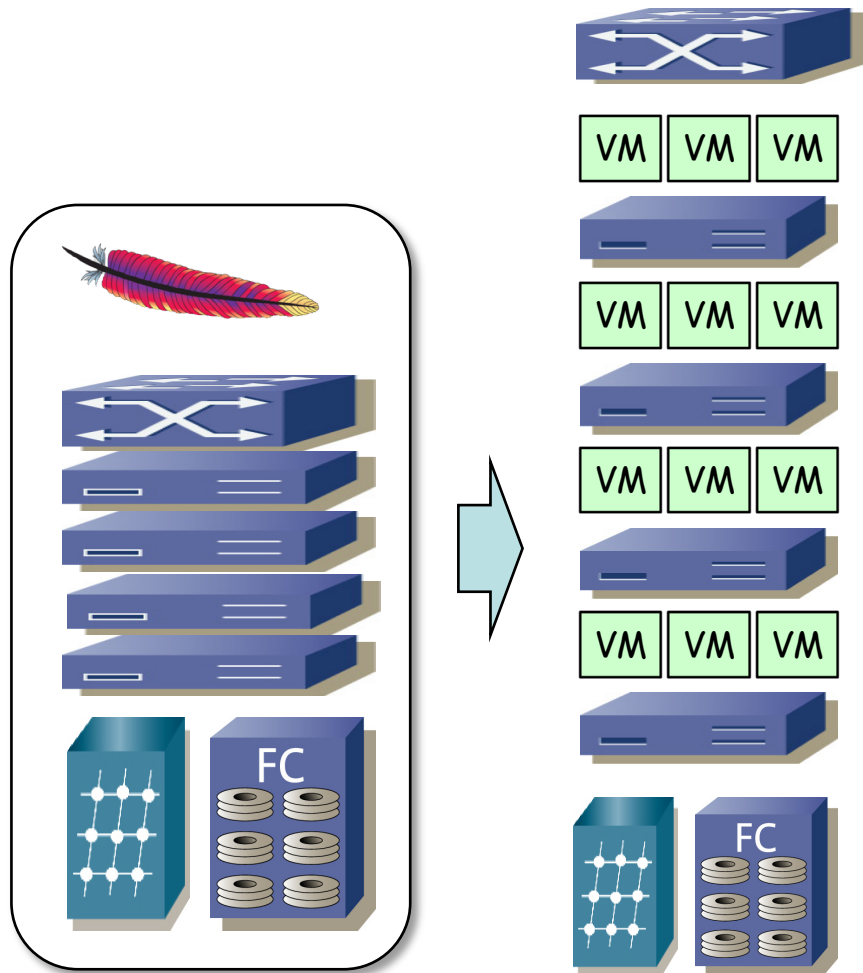




# Evolución del Silo

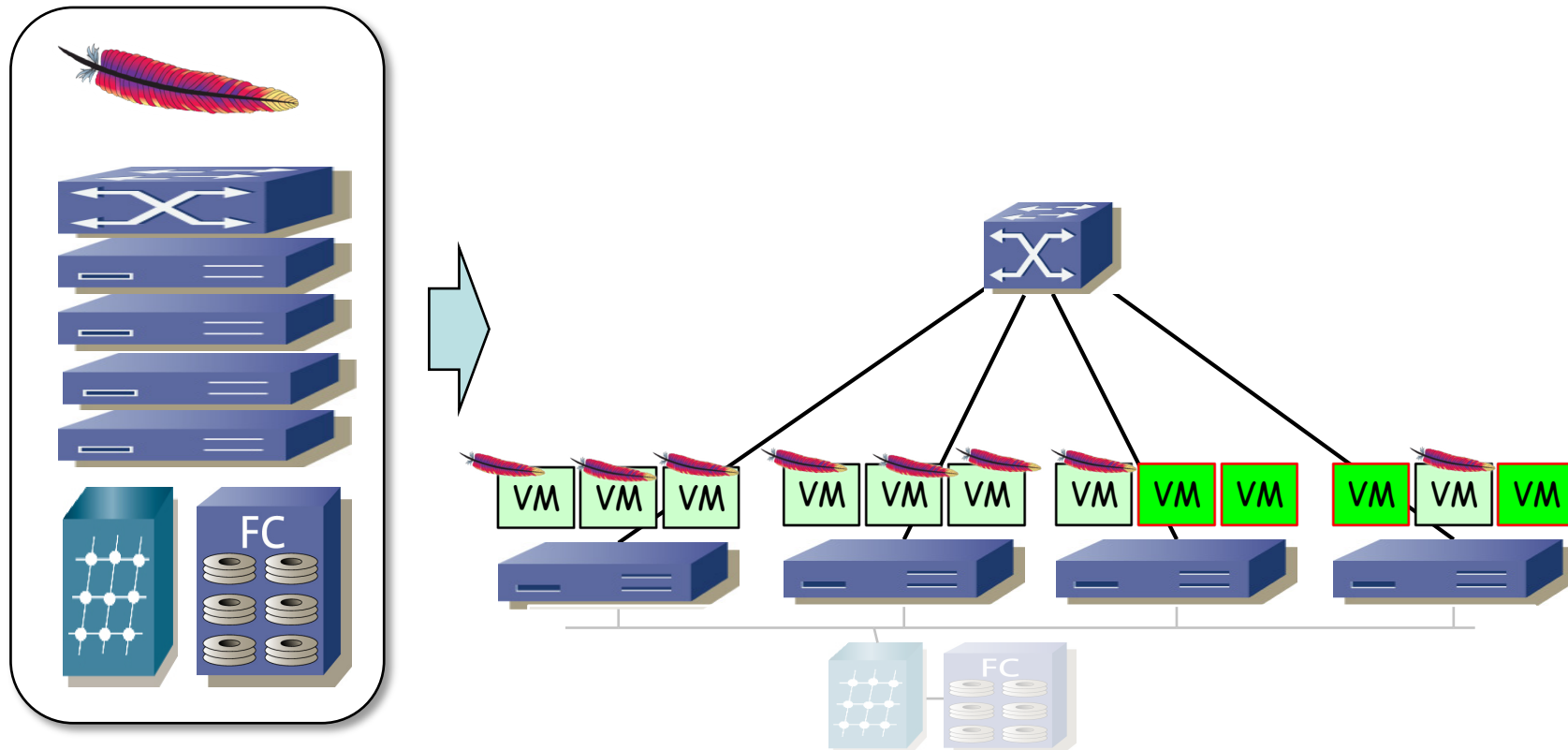
# Evolución del App. Silo

- VMs



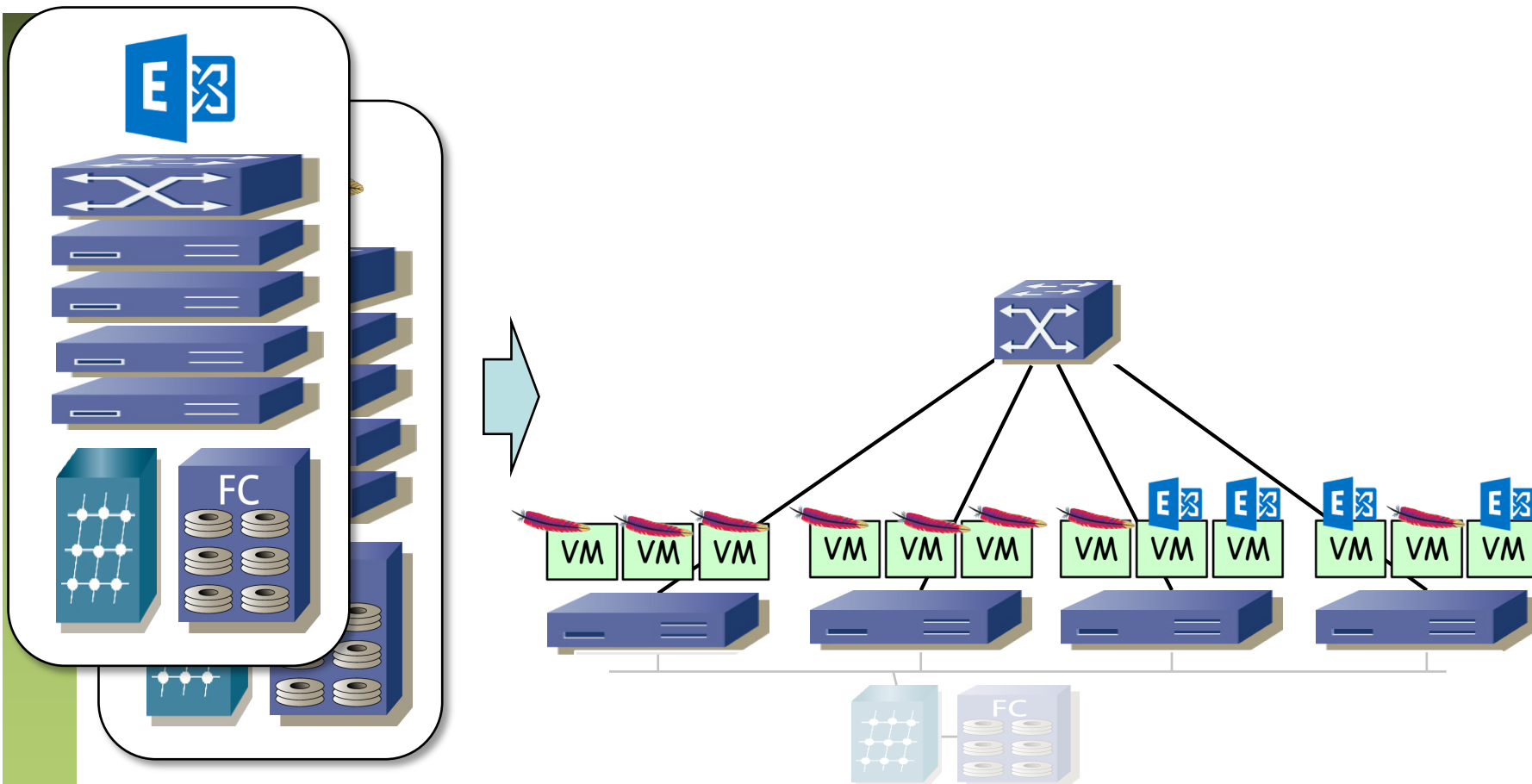
# Evolución del App. Silo

- VMs
- Pueden quedar recursos libres para otras VMs



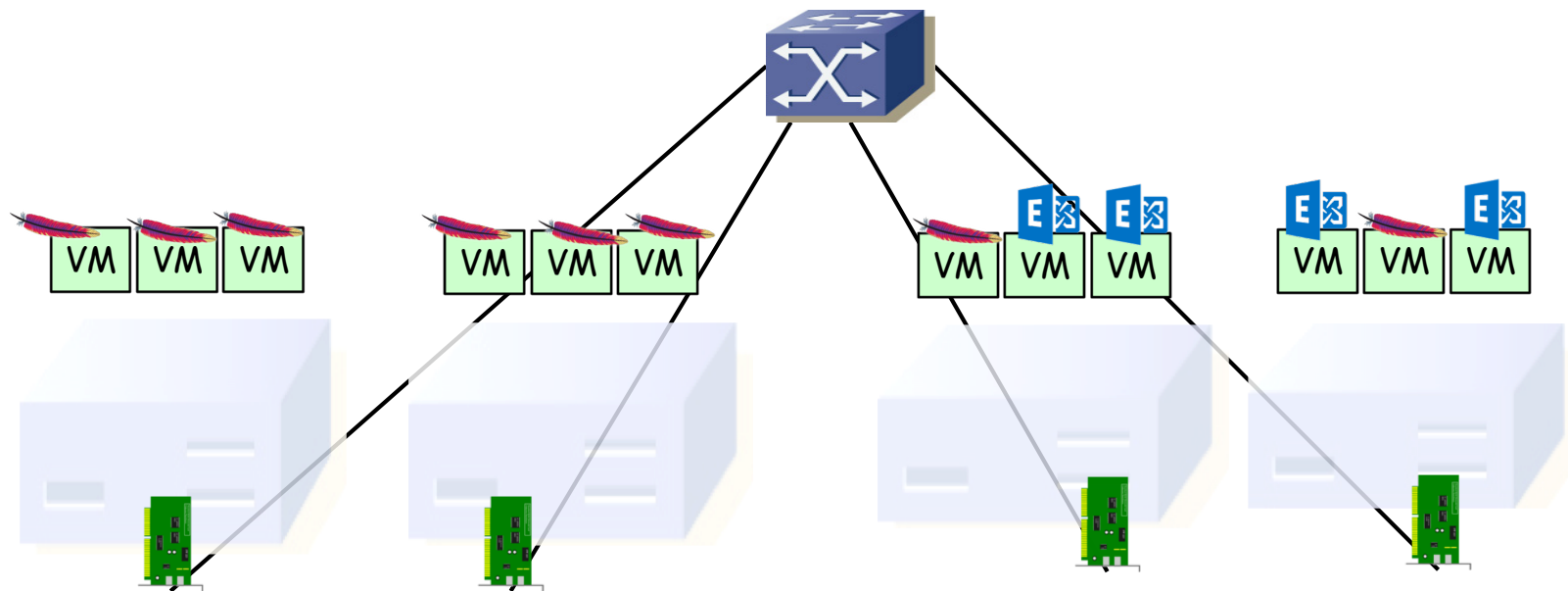
# Evolución del App. Silo

- VMs
- Pueden quedar recursos libres para otras VMs
- Consolidación



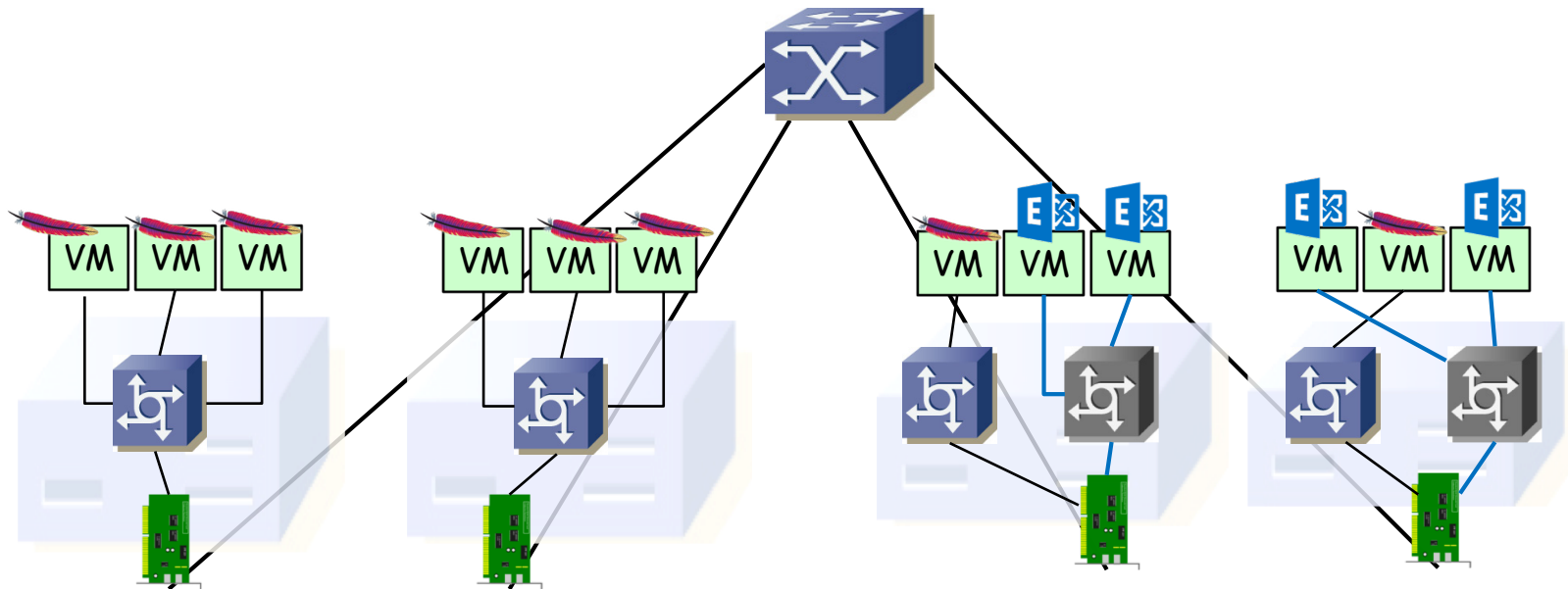
# Networking

- Ignoramos de momento el almacenamiento
- Supongamos servidores con una sola NIC



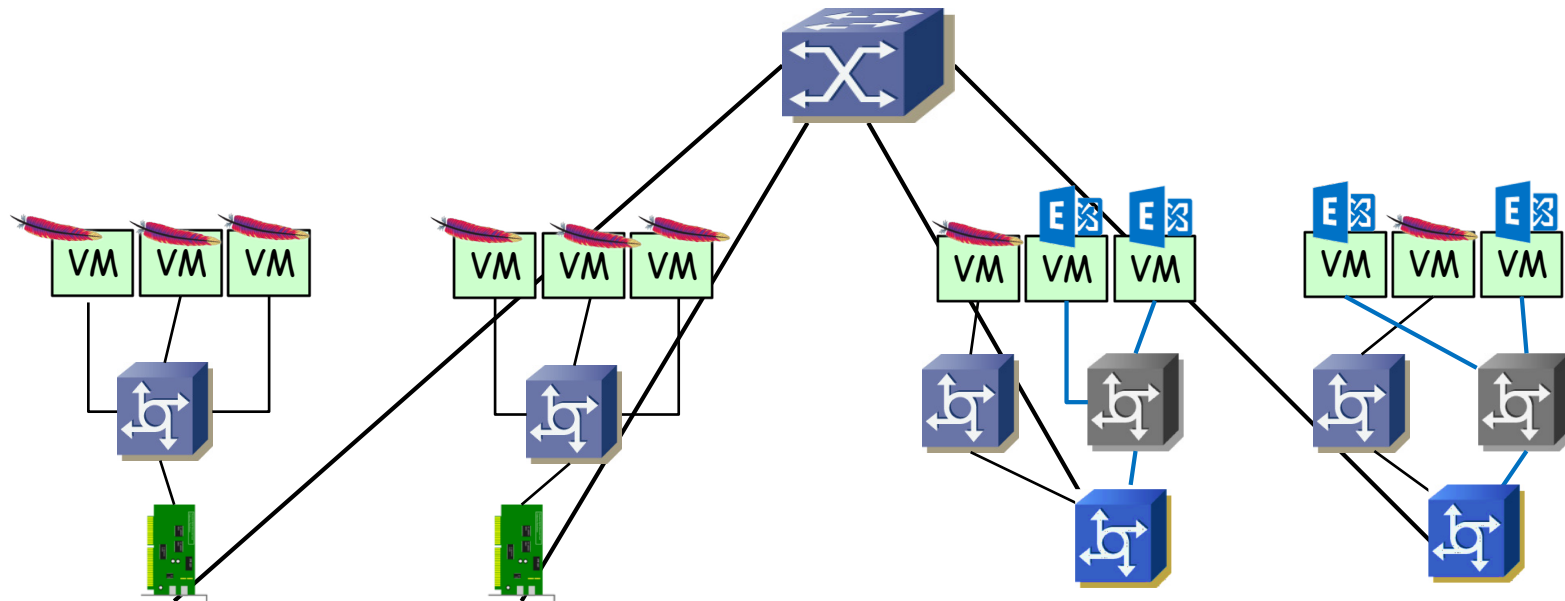
# Networking

- Ignoramos de momento el almacenamiento
- Supongamos servidores con una sola NIC
- Virtual Switches
- Por ejemplo diferente VLAN
- 802.1Q del host al switch físico



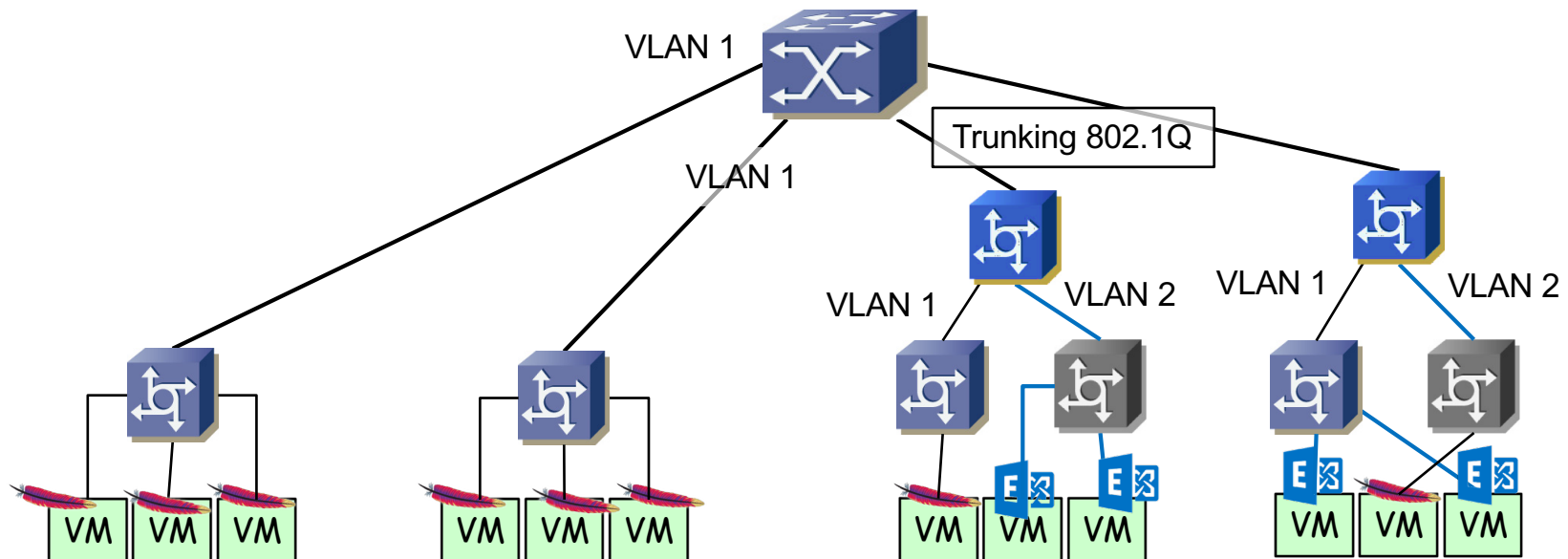
# Networking

- Ignoramos de momento el almacenamiento
- Supongamos servidores con una sola NIC
- Virtual Switches
- Por ejemplo diferente VLAN
- 802.1Q del host al switch físico



# Networking

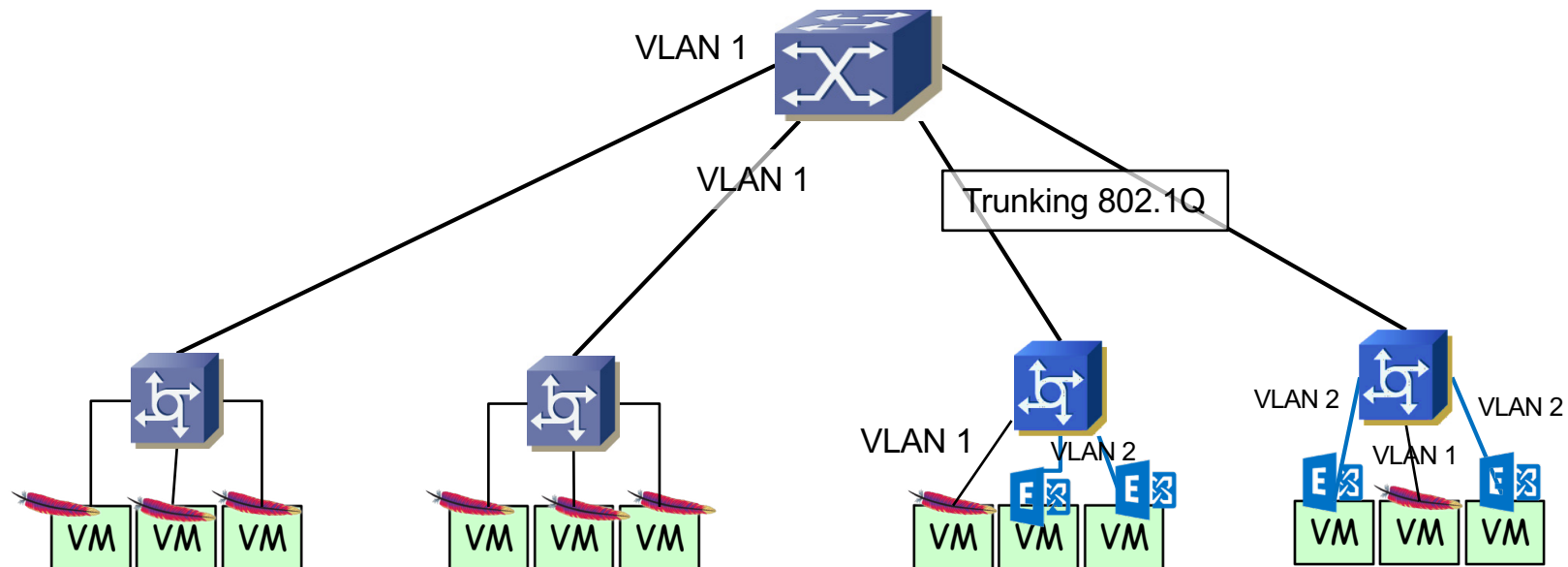
- Ignoramos de momento el almacenamiento
- Supongamos servidores con una sola NIC
- Virtual Switches
- Por ejemplo diferente VLAN
- 802.1Q del host al switch físico





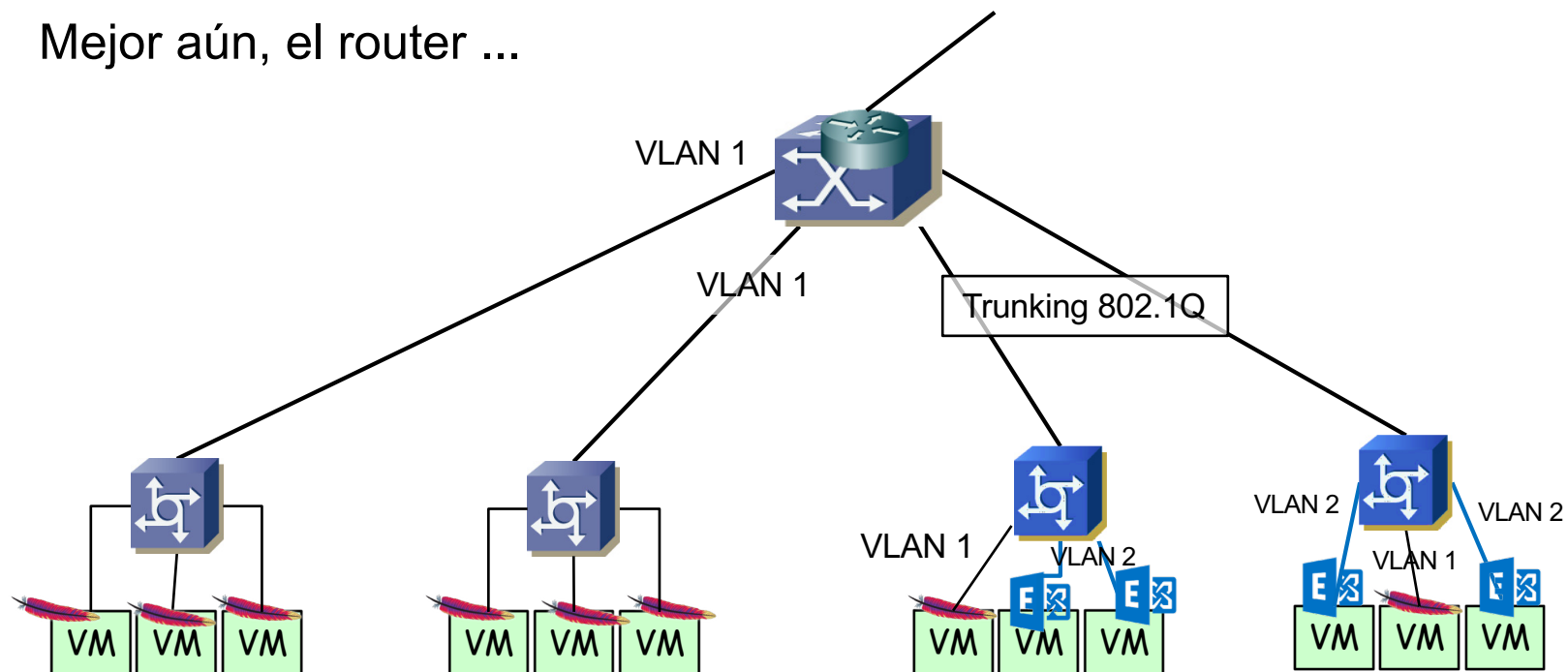
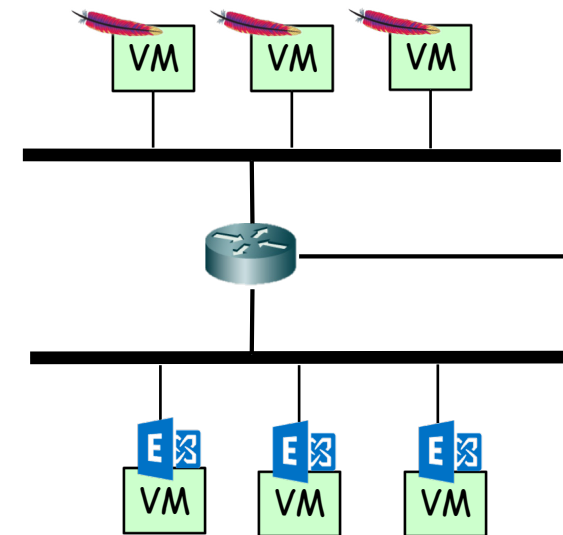
# Networking

- Ignoramos de momento el almacenamiento
- Supongamos servidores con una sola NIC
- Virtual Switches
- Por ejemplo diferente VLAN
- 802.1Q del host al switch físico
- O también vSwitches directamente con soporte de VLANs



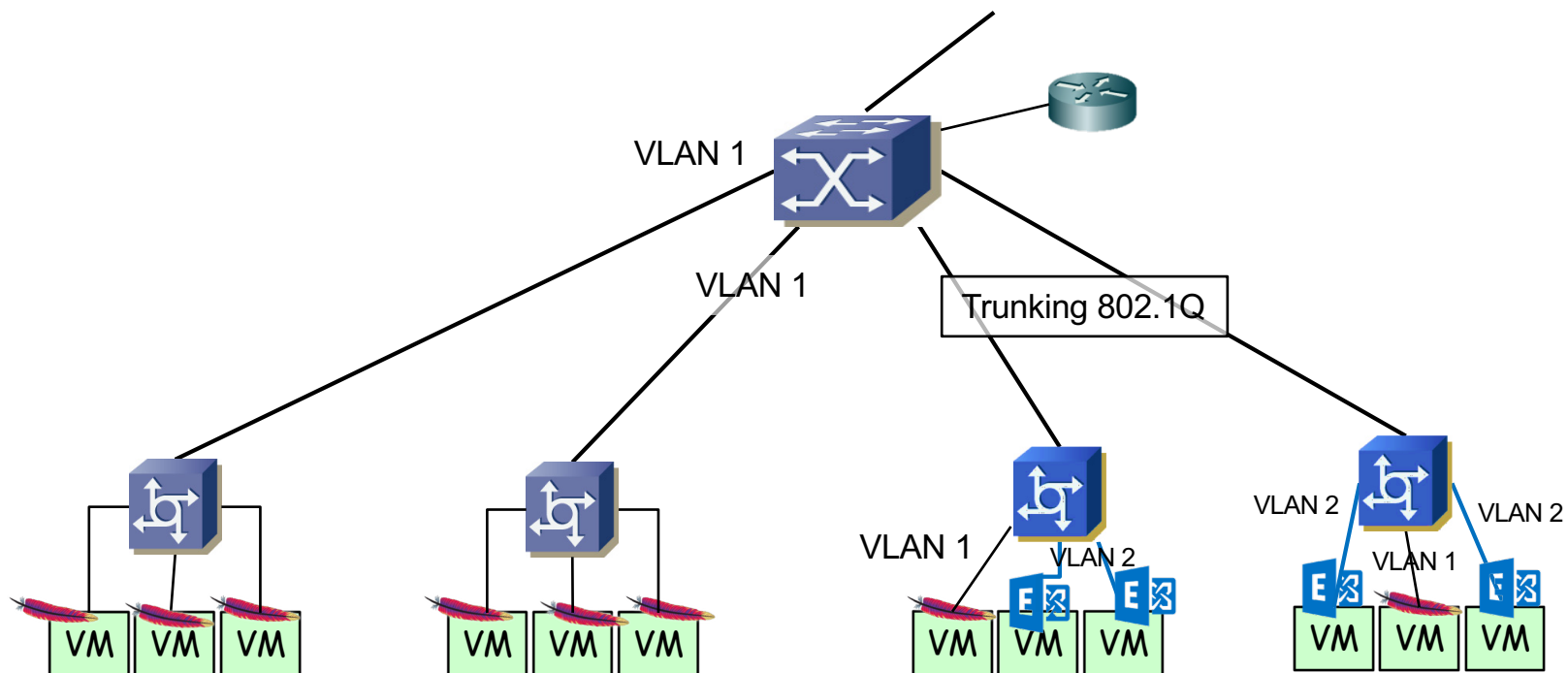
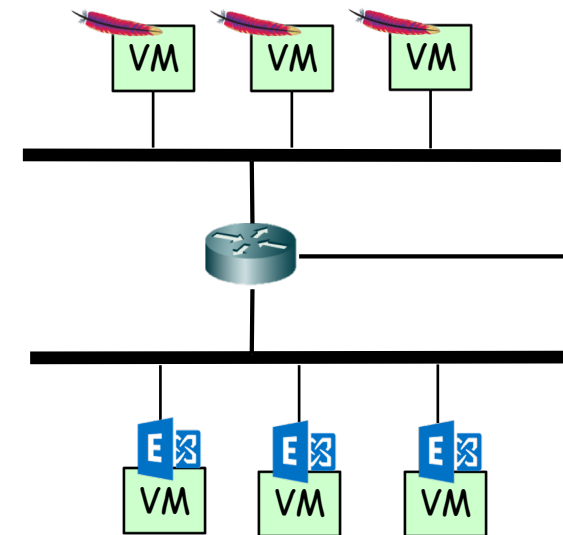
# Networking

- En capa 3 tenemos hosts en 2 subredes
- Conmutación en capa 2 es mezcla de switches físicos y vSwitches
- ¿Hacia el exterior?
- Por ejemplo el switch físico es capa 2/3 y enruta al exterior
- Podría enrutar entre esas subredes
- ¿Filtrado? Añadir FW ... ¿dónde? ...
- Mejor aún, el router ...



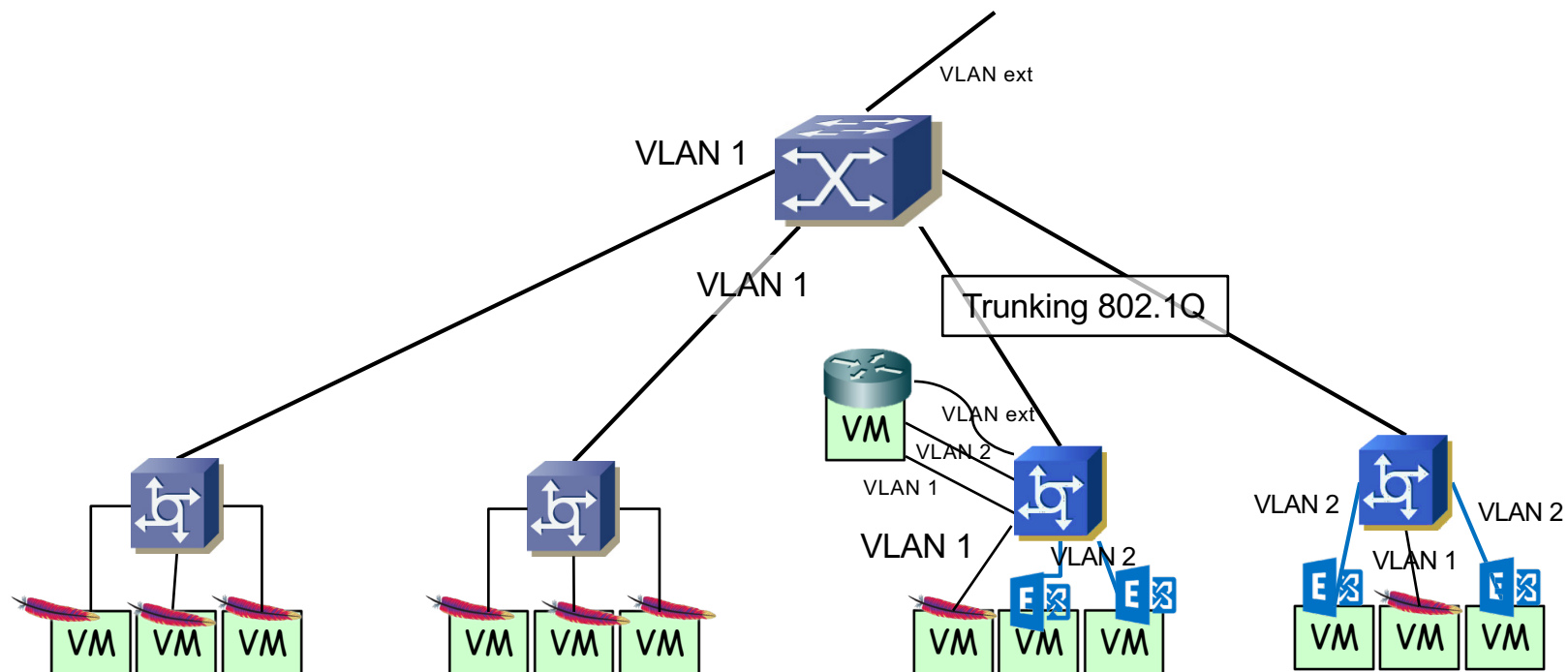
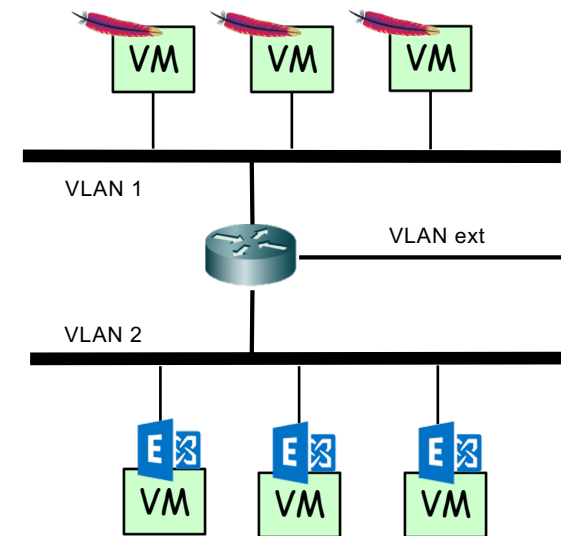
# Networking

- El router podría ser independiente del conmutador capa 2
- Con uno, dos, tres interfaces, con menos pero usando 802.1Q ... Múltiples opciones



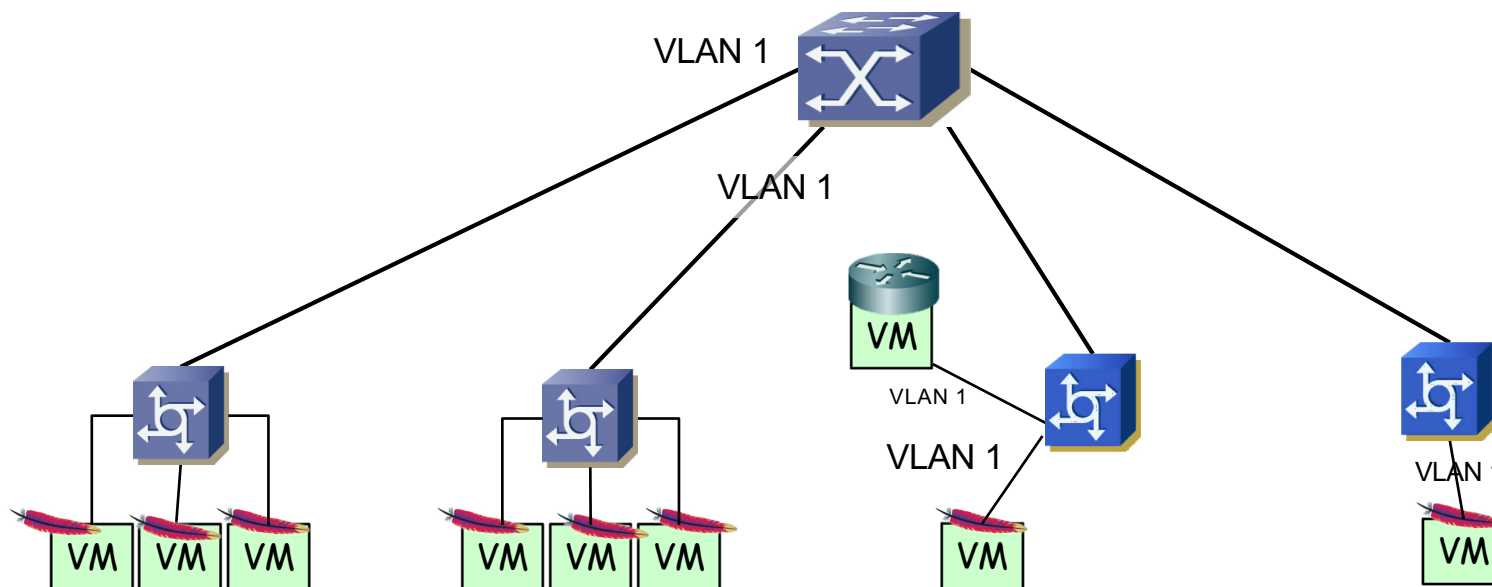
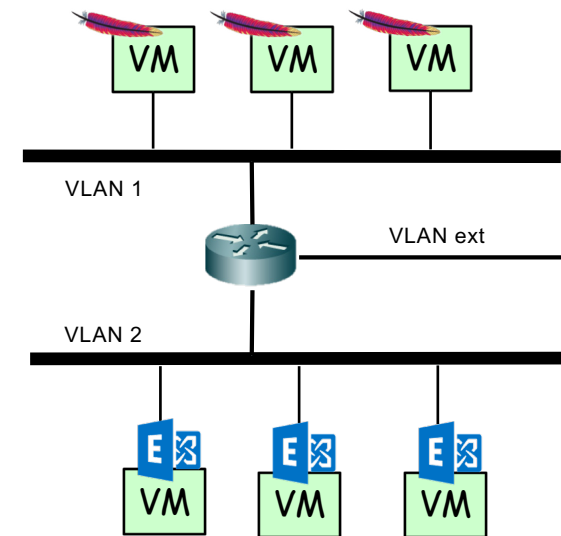
# Networking

- El router podría ser independiente del conmutador capa 2
- Con uno, dos, tres interfaces, con menos pero usando 802.1Q ... Múltiples opciones
- ¿Podría estar en un vSwitch?
- ¡ Podría ser una VM !
- ¿Cómo es ahora la red?



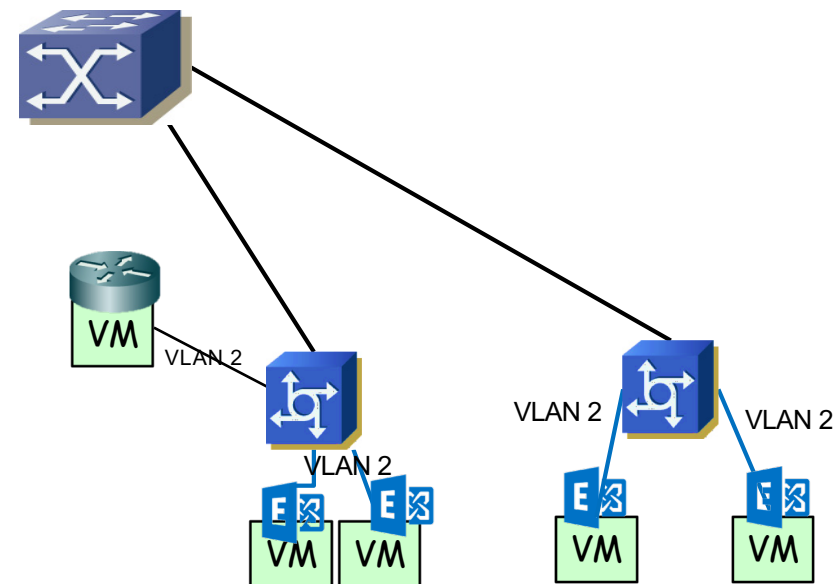
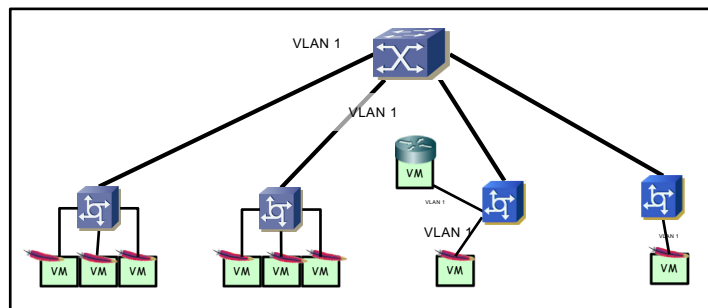
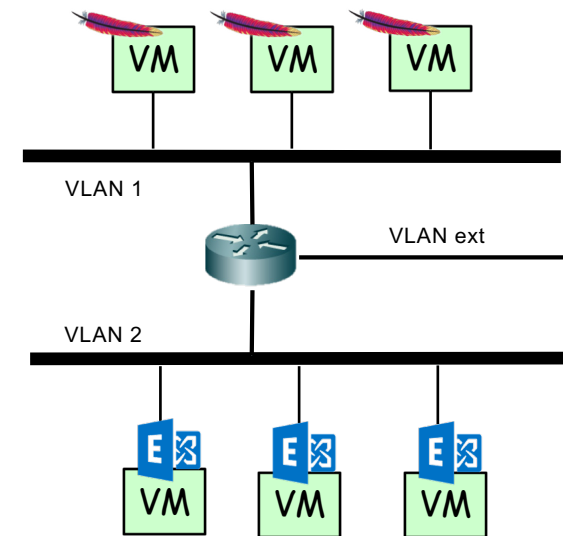
# Networking

- Nada ha cambiado en capa 3
- Capa 2 VLAN 1



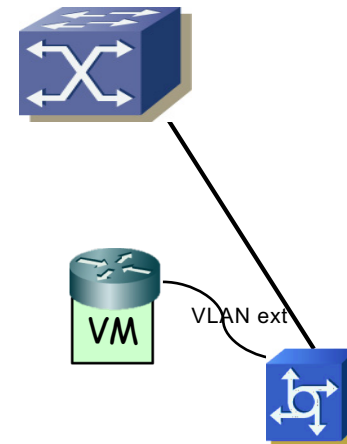
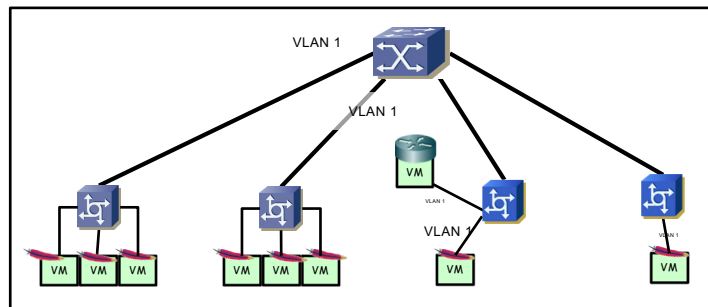
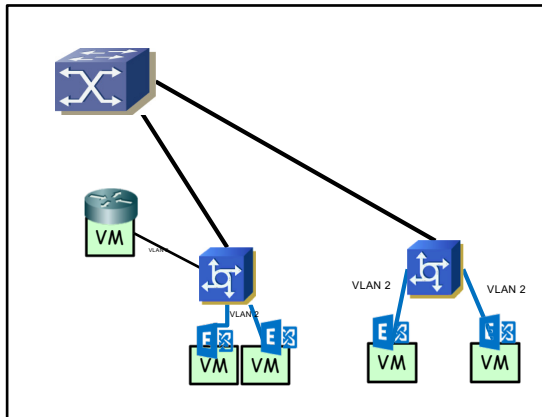
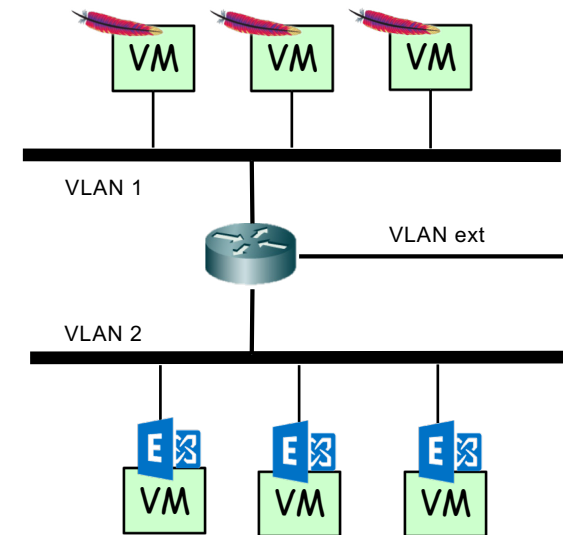
# Networking

- Nada ha cambiado en capa 3
- Capa 2 VLAN 1
- Capa 2 VLAN 2



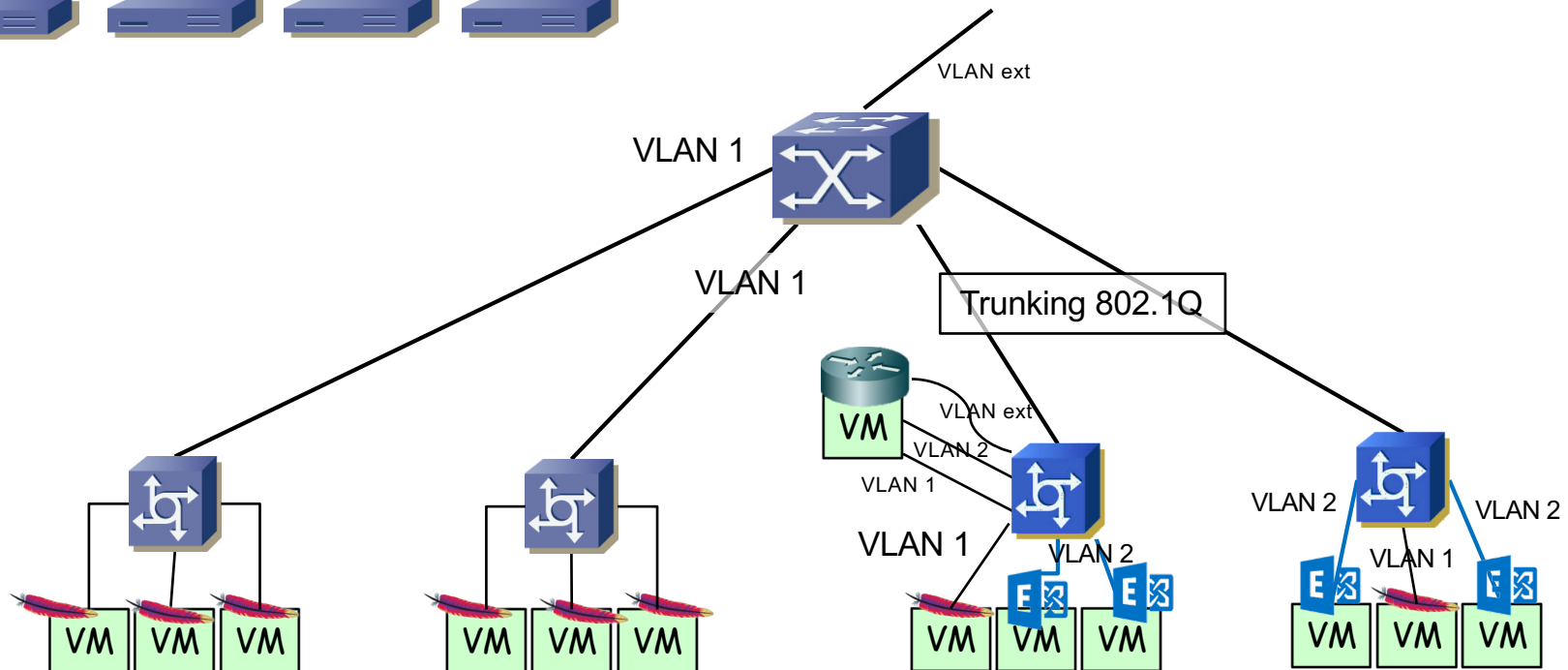
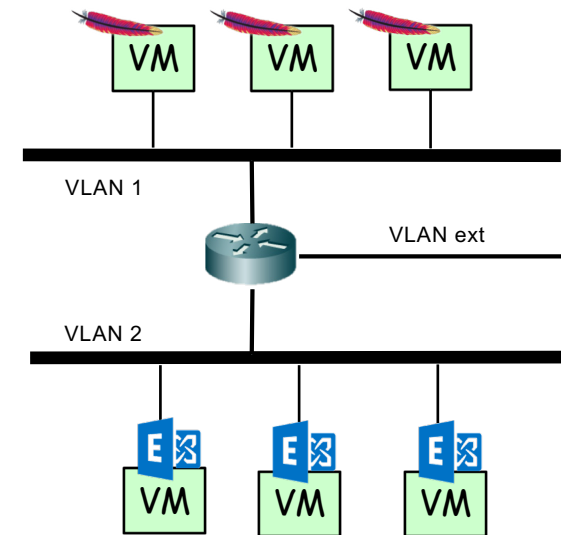
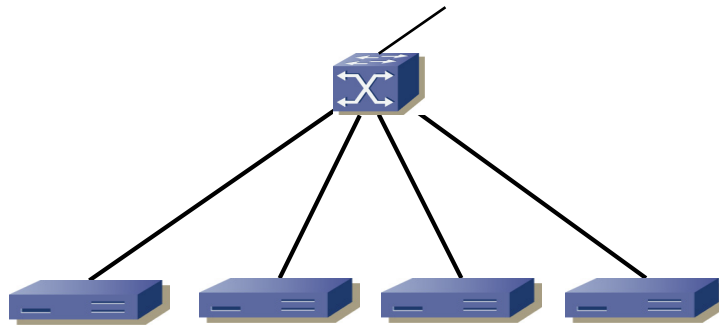
# Networking

- Nada ha cambiado en capa 3
- Capa 2 VLAN 1
- Capa 2 VLAN 2
- Capa 3 VLAN ext



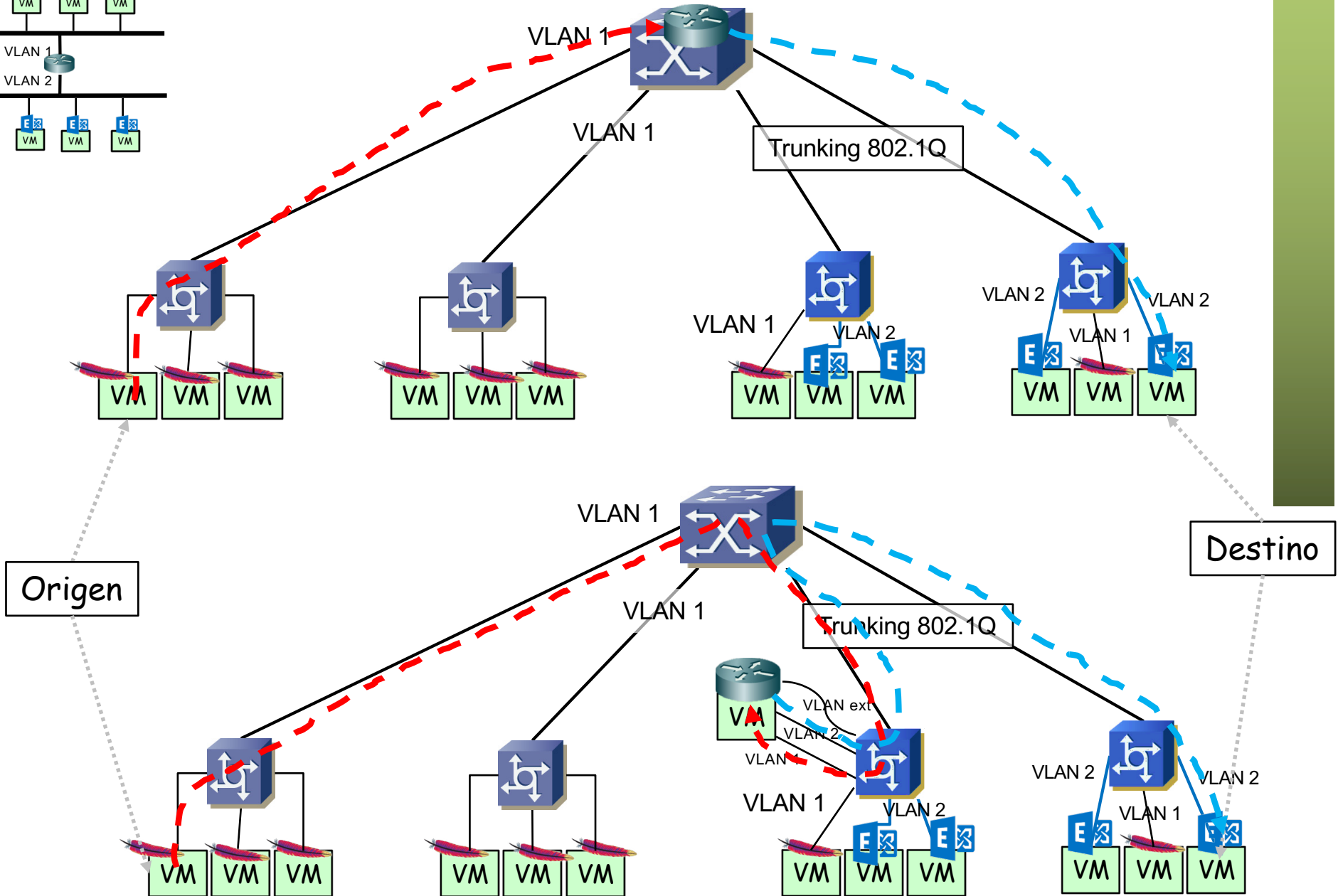
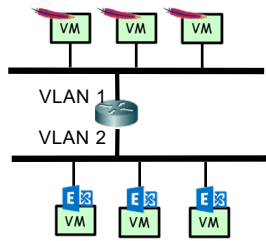
# Networking

- Pero no nos olvidemos de algo fundamental
- Físicamente tenemos: 4 servidores
- Que con las VMs y vSwitches se comportan como todo eso





# Networking: Caminos



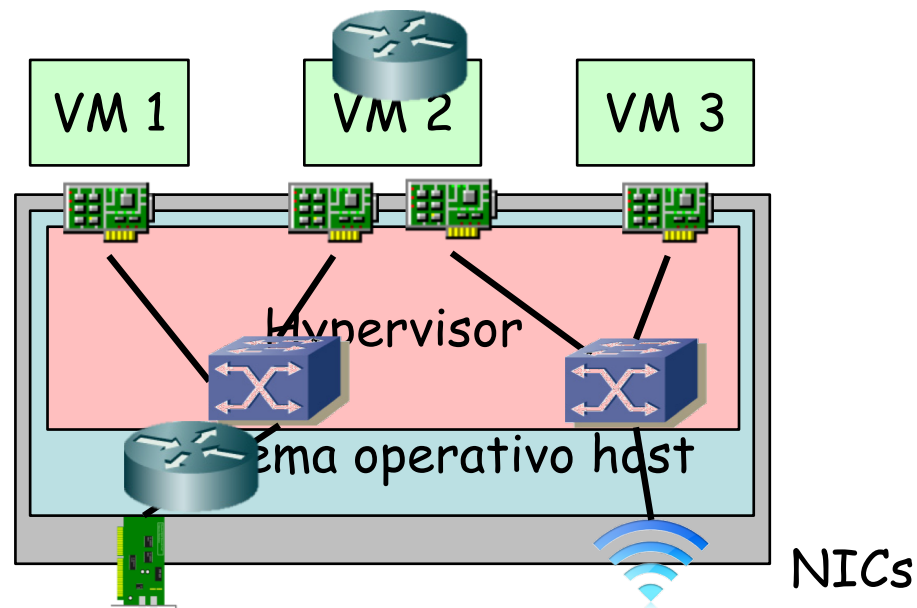
# Networking

- ¿Podemos hacer algo más que routers con esas VMs?
- Firewalls
- Antivirus en red
- Inspectores de contenido
- Balanceadores y publicadores
- Caches
- Puentes (sí, en vez de un vSwitch en el hypervisor estaría como una VM)
- Cualquier cosa que en el fondo sea software en un sistema operativo
- Por otro lado la funcionalidad de router podría ser llevada a cabo por el Kernel de un host en lugar de por una VM

# Networking en escenario *hosted*

# Escenario *hosted*

- Es utilizado principalmente en soluciones de escritorio
- En estos casos suele haber 1 ó 2 NICs (Ethernet + WiFi)
- La NIC WiFi se virtualiza haciéndola parecer una Ethernet
- De cara al exterior es frecuente que el host haga de router o de NAT
- Una VM puede actuar como un equipo de conmutación capa 3



# Beneficios de la virtualización

- Independencia del hardware
- Consolidación
  - Ahorro en hardware para correr los servicios (y espacio)
  - Ahorro en consumo eléctrico
  - Ahorro en refrigeración
- Sencilla separación de entornos de desarrollo, pruebas y producción
- Sencilla creación, backup y replicación
  - Facilita la migración a otro hardware
  - Creación de instantáneas y retorno a ellas
- Instalaciones menos atadas al hardware pues requieren drivers para el hardware virtualizado
- Permite mantener software (sistemas operativos) antiguos sobre hardware moderno (aunque no tengan drivers)

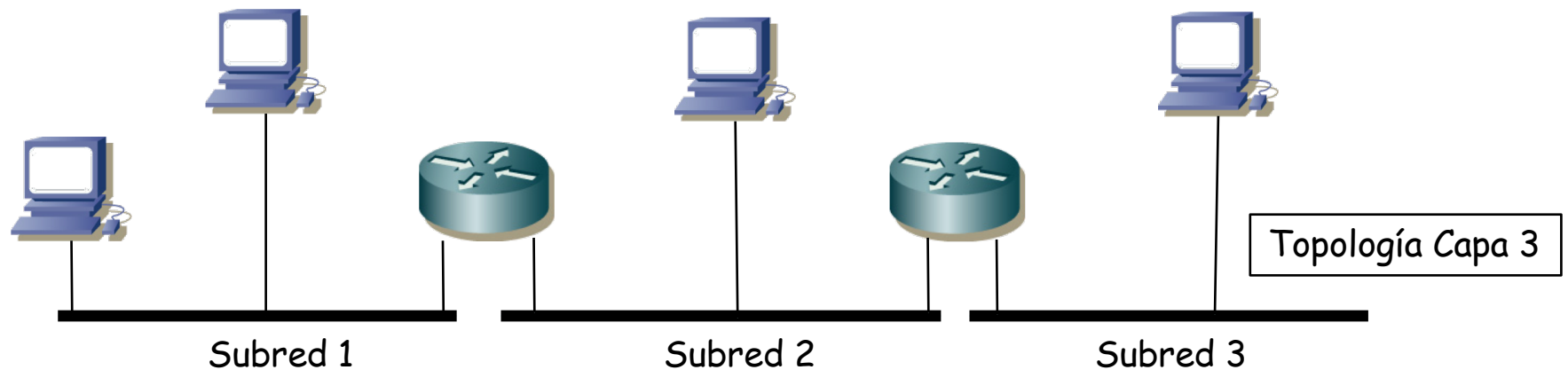
# Desventajas de la virtualización

- Pérdida de rendimiento
  - Con aplicaciones con alta carga, que hacen un uso intensivo del hardware, puede no ser rentable
  - Hay que dimensionar la capacidad para la combinación de carga de VMs
- Compatibilidad con el hardware
  - Podemos contar con hardware especializado para el que no exista drivers en el hypervisor
- Un fallo hardware tiene efecto en múltiples VMs
- Depuración del sistema global más compleja, mayor acomplamiento
- Nuevas herramientas de gestión, nuevas habilidades requeridas al personal de IT

# Casos de prácticas

# Práctica

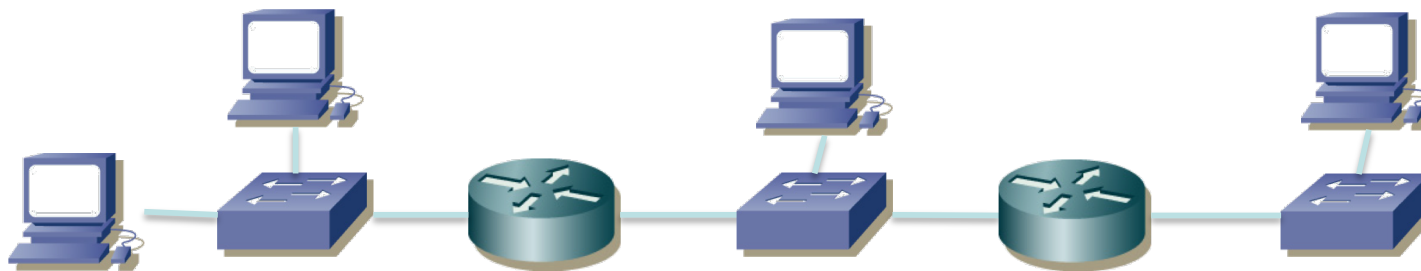
- Vamos a crear este escenario
- Hay 3 LANs Ethernet (3 subredes IP, una sobre cada LAN), dos routers interconectándolas y 4 PCs repartidos entre ellas





# Práctica

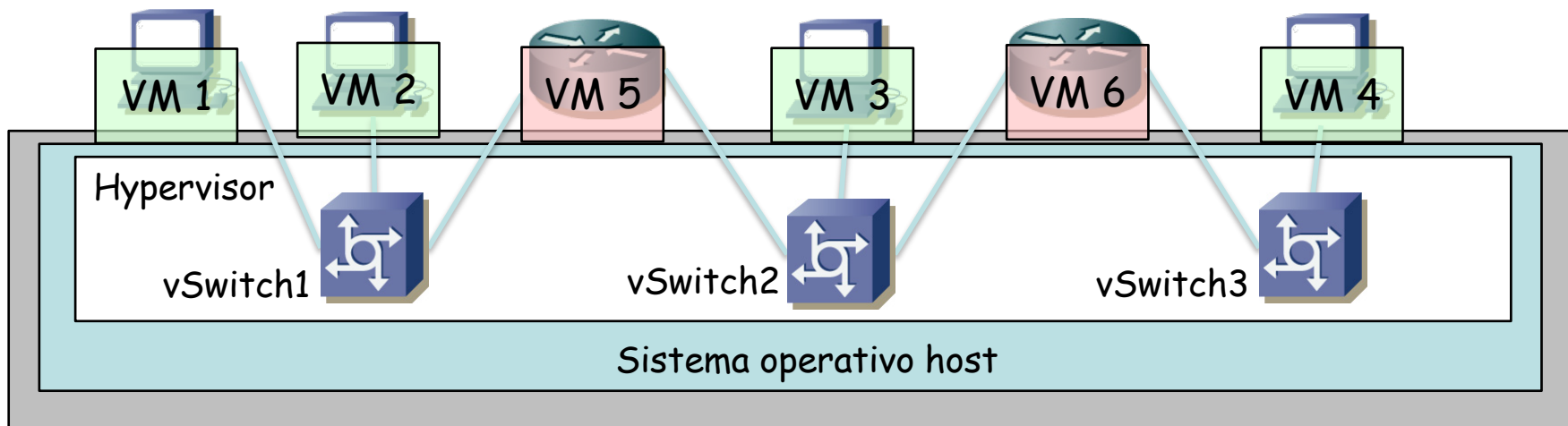
- Vamos a crear este escenario
- Hay 3 LANs Ethernet (3 subredes IP, una sobre cada LAN), dos routers interconectándolas y 4 PCs repartidos entre ellas
- Cada LAN creada con un Switch



Topología física

# Práctica

- Vamos a crear este escenario
- Hay 3 LANs Ethernet (3 subredes IP, una sobre cada LAN), dos routers interconectándolas y 4 PCs repartidos entre ellas
- Cada LAN creada con un Switch
- Sin embargo va a ser todo virtual:
  - Los PCs son VMs
  - Los routers también
  - Los conmutadores son vSwitches
  - Todo ello con un hypervisor tipo 2 (VirtualBox)



# Práctica

- Podemos también crear puentes con VMs

