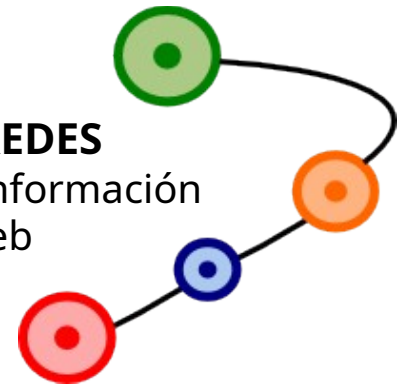




**Administración y Gestión de Redes**  
Lic. en Sistemas de Información

**Laboratorio de REDES**  
Recuperación de Información  
y Estudios de la Web



# Infraestructura de Data Centers y Cloud

**Normas de Infraestructura de Data Centers**  
**Virtualización**  
**Software Defined Networks (SDNs)**

Equipo docente:

Fernando Lorge (florge@unlu.edu.ar)

Santiago Ricci (sricci@unlu.edu.ar)

Alejandro Iglesias (aaiglesias@unlu.edu.ar)

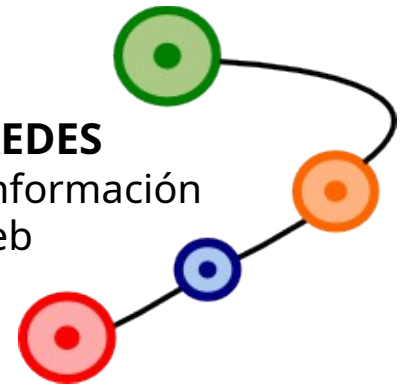
Mauro Meloni (maurom@unlu.edu.ar)

Patricio Torres (ptorres@unlu.edu.ar)



**Administración y Gestión de Redes**  
Lic. en Sistemas de Información

**Laboratorio de REDES**  
Recuperación de Información  
y Estudios de la Web



# Virtualización

Definición y Clasificación  
Características  
Técnicas de Virtualización  
Definiciones formales

## Virtualización

- Abstracción de los recursos de computación
- Abstracción → Aislación → Optimización de Recursos
- Desde el punto de vista de S.O.: memoria, dispositivos, archivos ...
- System Virtualization: Abstracción de una computadora completa, incluyendo memoria, CPU y periféricos.
- Hoy día la virtualización es capaz de proveer un entorno virtual para la ejecución de aplicaciones (o VMs), almacenamiento, memoria, red...



# Virtualización

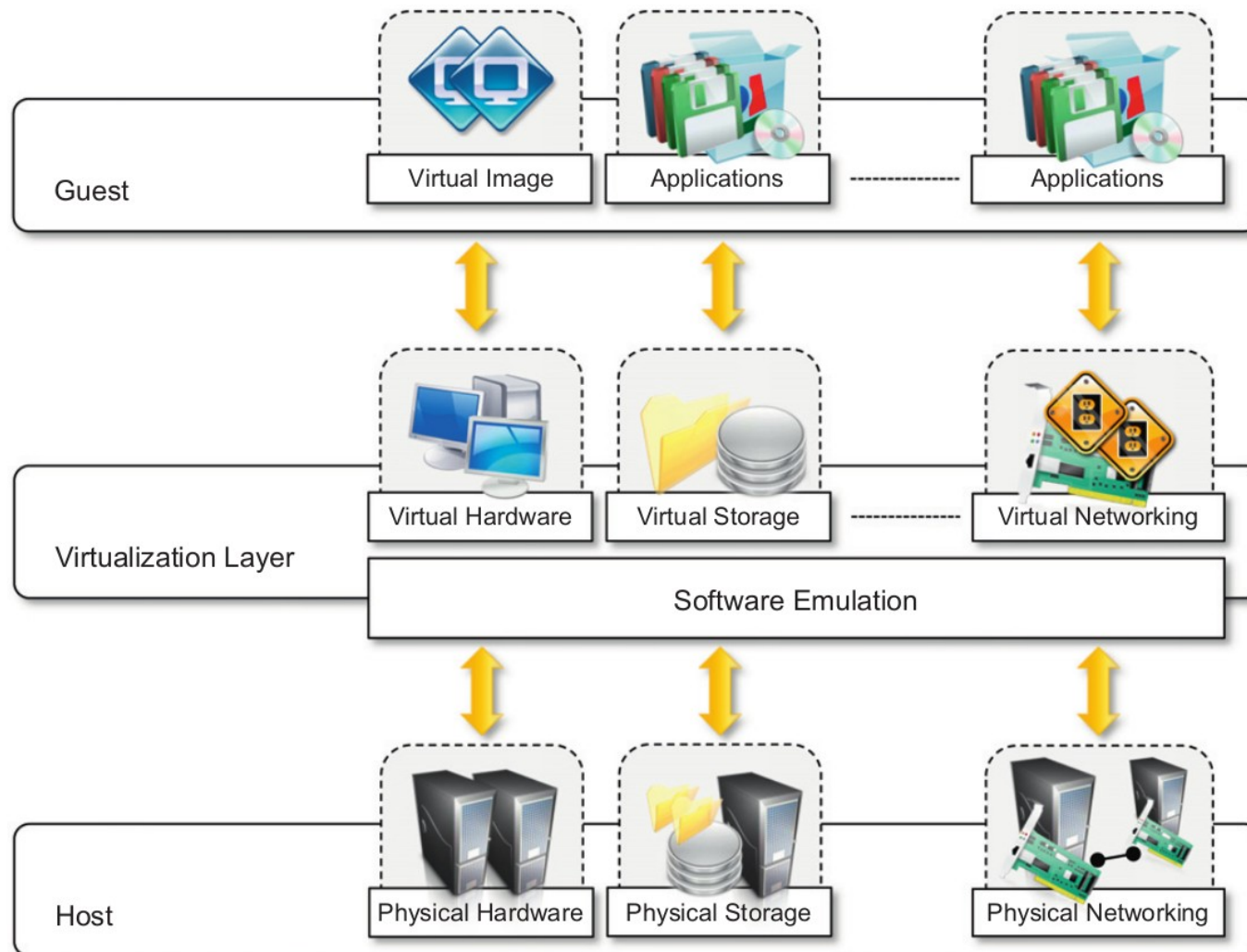
## Tipos de Virtualización

- Máquina (Computer System Virtualization)
- Almacenamiento (Storage Virtualization) – Por ej., SANs.
- Red (Network Virtualization) – Externa (VLANs), Interna (consecuencia de la virtualización de sistema: NAT, TUN/TAP, etc.)
- Escritorio (Desktop Virtualization) - VNC, RDP, X Server.
- Servidor de Aplicaciones (Application Server Virtualization)



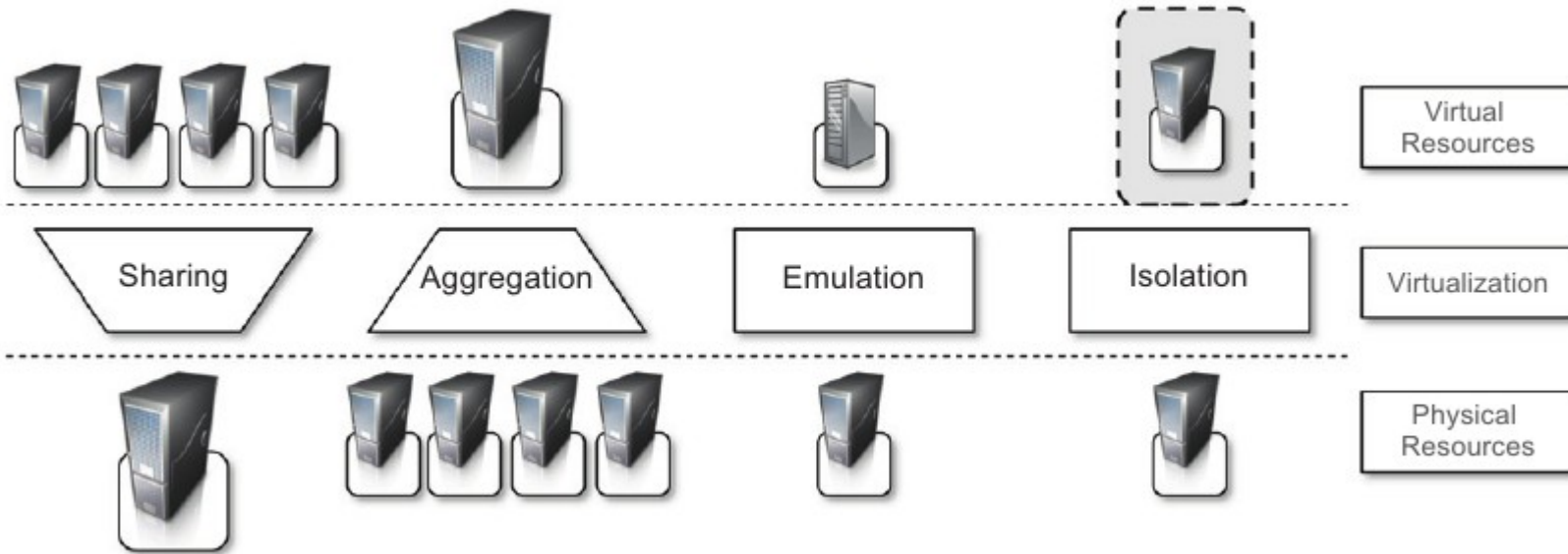
# Virtualización

## Componentes del Modelo de Referencia



# Virtualización

## Características implementables



- Compartimiento
- Agregación
- Emulación
- Aislamiento
- Performance Tuning → QoS → SLA



## Técnicas de virtualización

Proveen soporte para la ejecución de programas, ya sean:

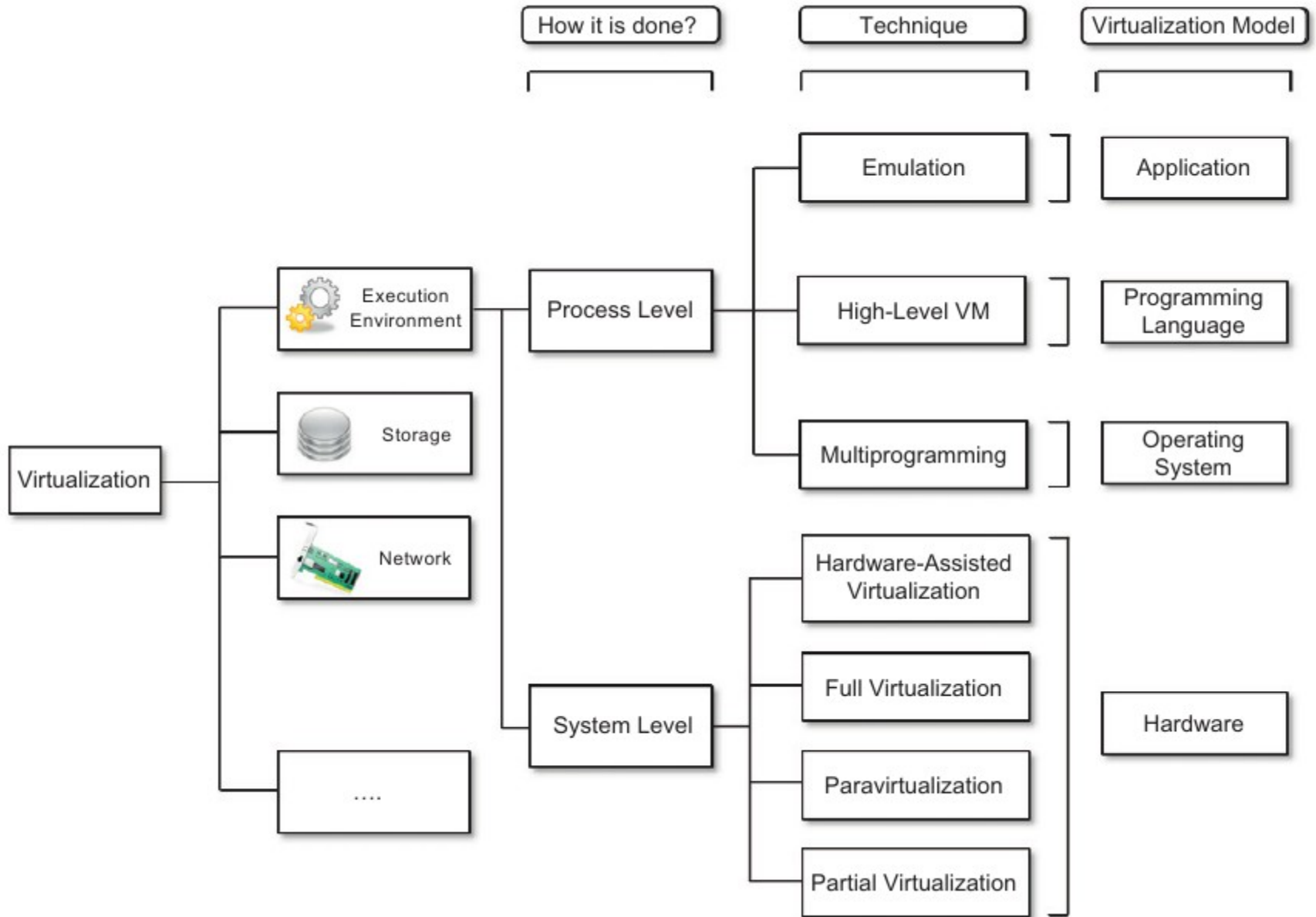
- S.O.
- ABI (Application Binary Interface)
- Programas compilados sobre un modelo de máquina abstracta
- Aplicaciones

Implementados:

- Directamente sobre el hardware
- Sobre el S.O.
- Mediante una aplicación
- Por librerías enlazadas estática o dinámicamente.



# Virtualización



## Virtualización de máquinas 'computadoras' (system virtualization)

- Introducida por IBM en sus mainframes en la década de 1960.
- Fines de 1990 VMWare en arquitecturas intel x386.
- En 2005 Intel y AMD incorporan a sus líneas de procesadores x86 extensiones específicas para dar soporte a virtualización (Intel VT, AMD-V).

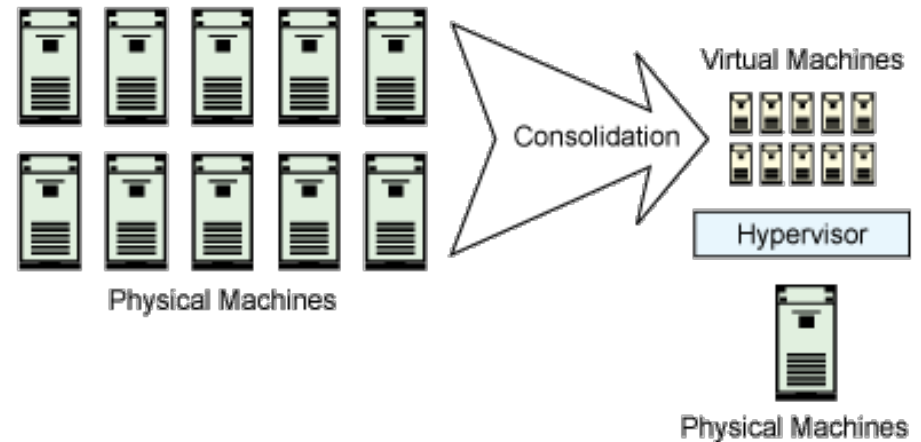


# Virtualización

## Virtualización

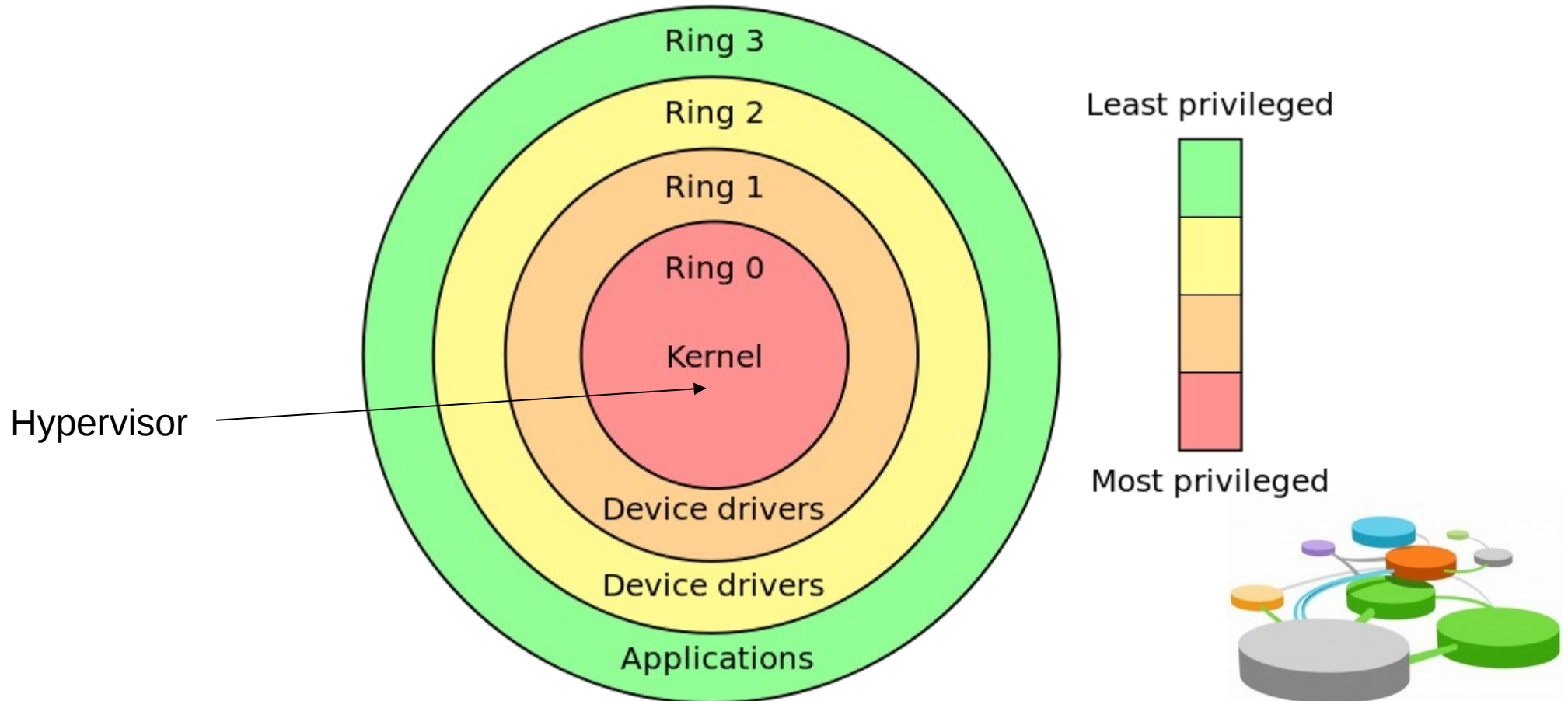
En alza en los últimos años debido a:

- Incremento en la capacidad de cómputo.
- Hardware y software infrautilizado.
- Necesidad de espacio adicional.
- Iniciativas “verdes”
- Costos administrativos



## ISA - Soporte de hardware

Las plataformas de hardware generalmente poseen conjuntos de instrucciones dividido en clases que definen quién puede utilizarlas:



## ISA - Soporte de hardware

### Arquitecturas:

- IBM System/370, System/390 y mainframes ZSeries
- X86: Intel VT, AMD-V 2005
- POWER: ISA v 2.03 2006 (ISA v 2.06 2009 full virt for embedded)
- ARM (Cortex A15 2010)
- SPARC UltraSPARC Architecture 2005



# Virtualización

## Técnicas de Virtualización de Plataforma o Sistema:

### 1) Emulación

- Bucle Leer, Interpretar/Traducir, Ejecutar.

### 2) Full Virtualization

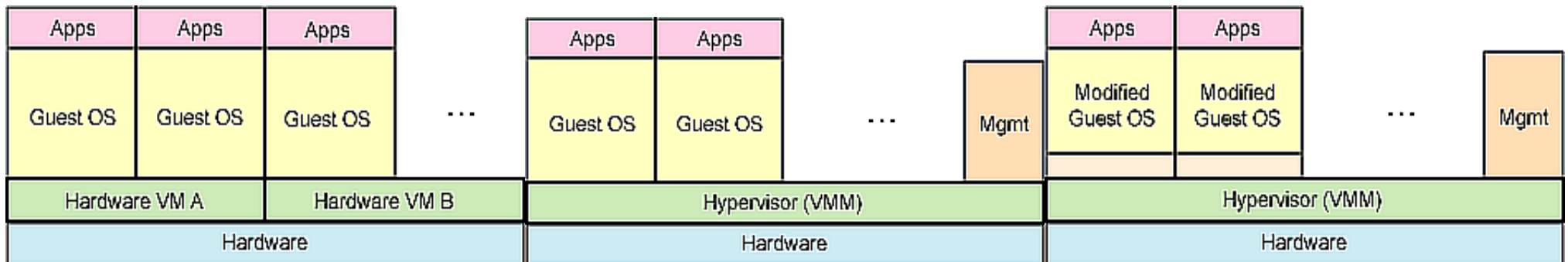
- “Traducción Binaria” realizada por un Hypervisor

### 3) Paravirtualización

- El Hypervisor permite compartir el hardware, el guest “lo sabe”

### 4) HVM – Hardware Asisted Virtualization

- El hardware provee soporte en el CPU para ejecutar un hypervisor



1) Emulación

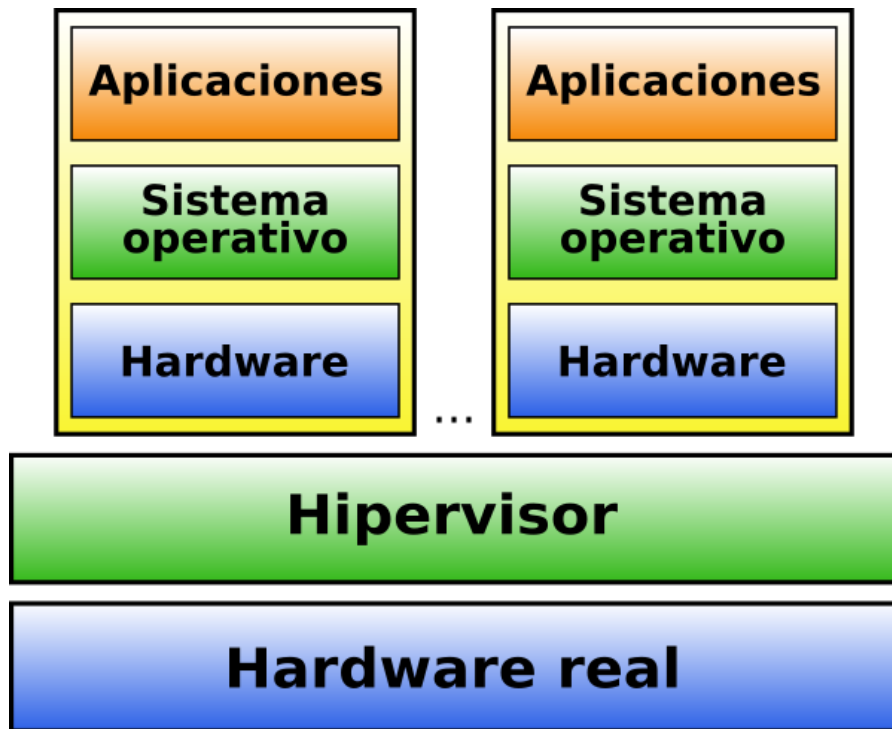
2,4) Full Virtualization, HVM

3) Paravirtualization

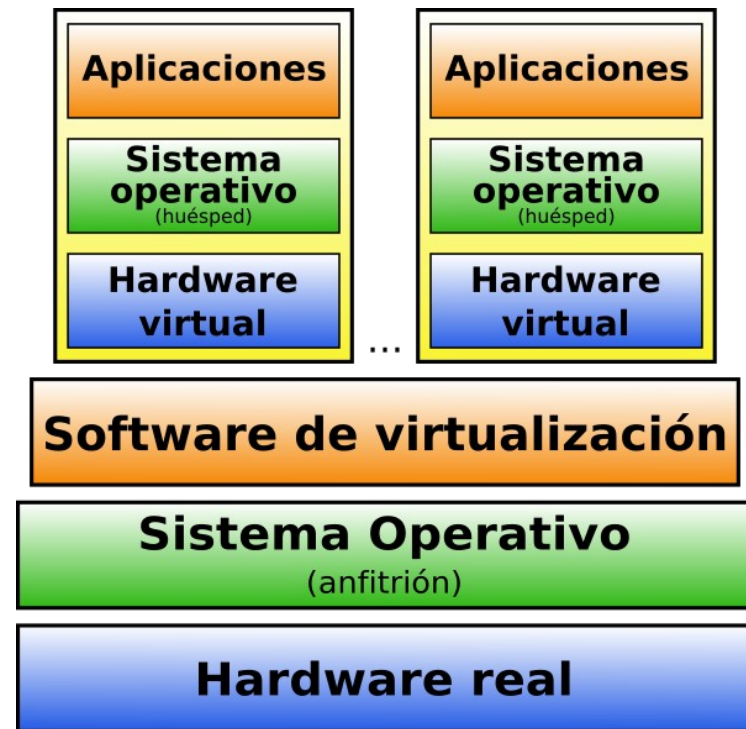
# Virtualización

## Tipos de Hipervisor

1. Tipo I - Bare-Metal, unhosted, native
2. Tipo II - Hosted



Tipo I - Bare-Metal



Tipo II - Hosted

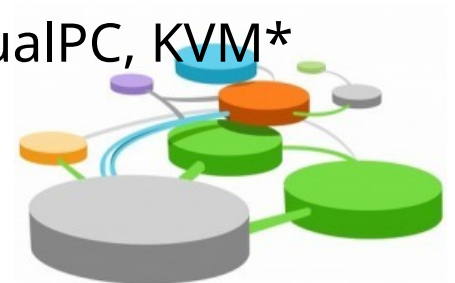
## Hypervisores Type I (native, bare-metal)

- Se ejecutan directamente sobre el hardware.
- Interactúan directamente con la interfaz ISA (instruction Set Architecture) del hardware.
- Deben implementar manejadores de dispositivos y capacidades de administración del hardware.
- Ejemplos: Xen, VMWare ESXi, Citrix, MS Hyper-V Server

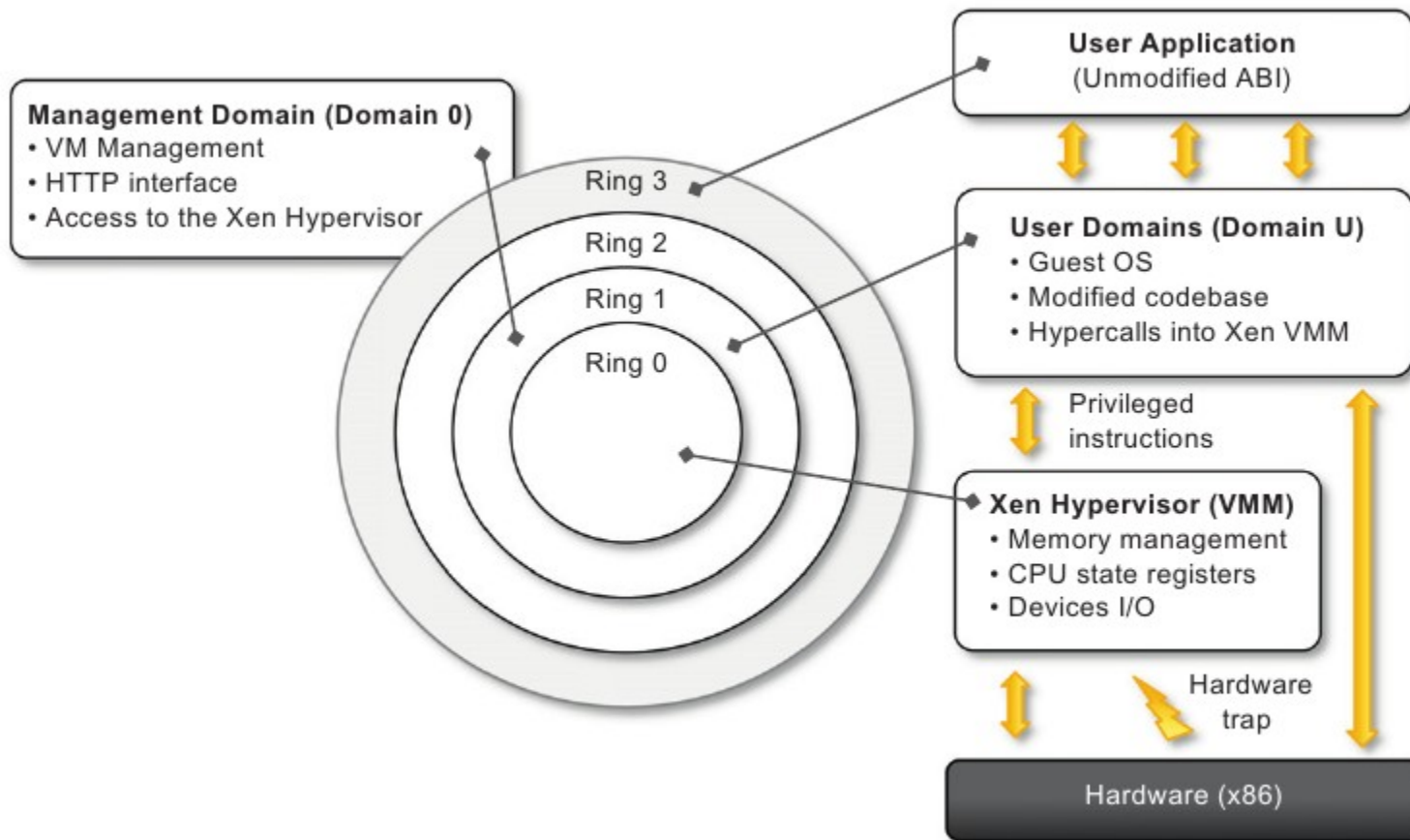


## Hypervisores Type II (Hosted)

- Se ejecutan sobre un S.O. convencional.
- Aprovechan los manejadores de dispositivos y capacidades de administración del hardware.
- Interactúan con los S.O. a través de la ABI (Application Binary Interface) y emulan la ISA del hardware virtual para las VMs.
- Ejemplos: VMWare Workstation, Parallels, VirtualBox, VirtualPC, KVM\*

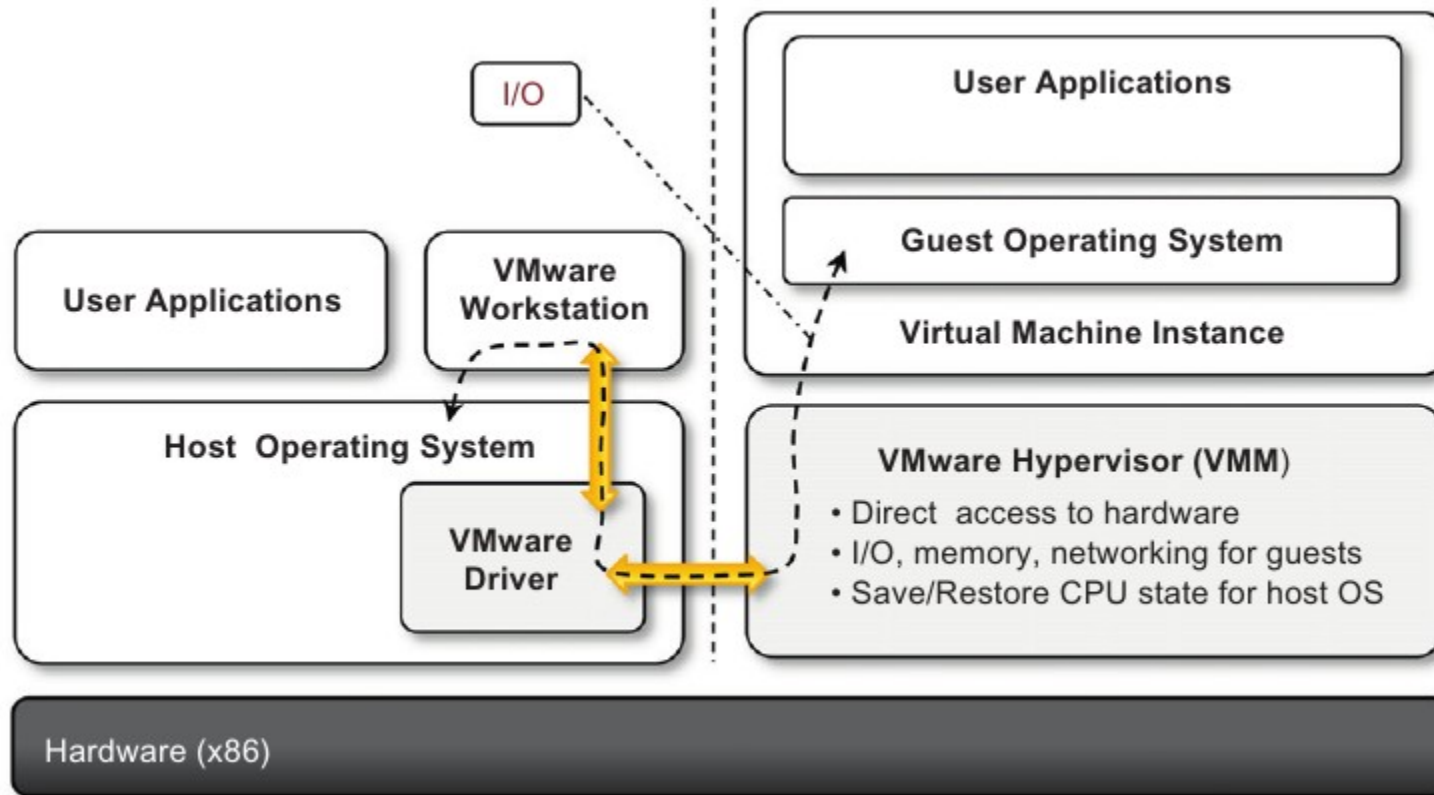


## Arquitectura XEN (caso paravirtual)



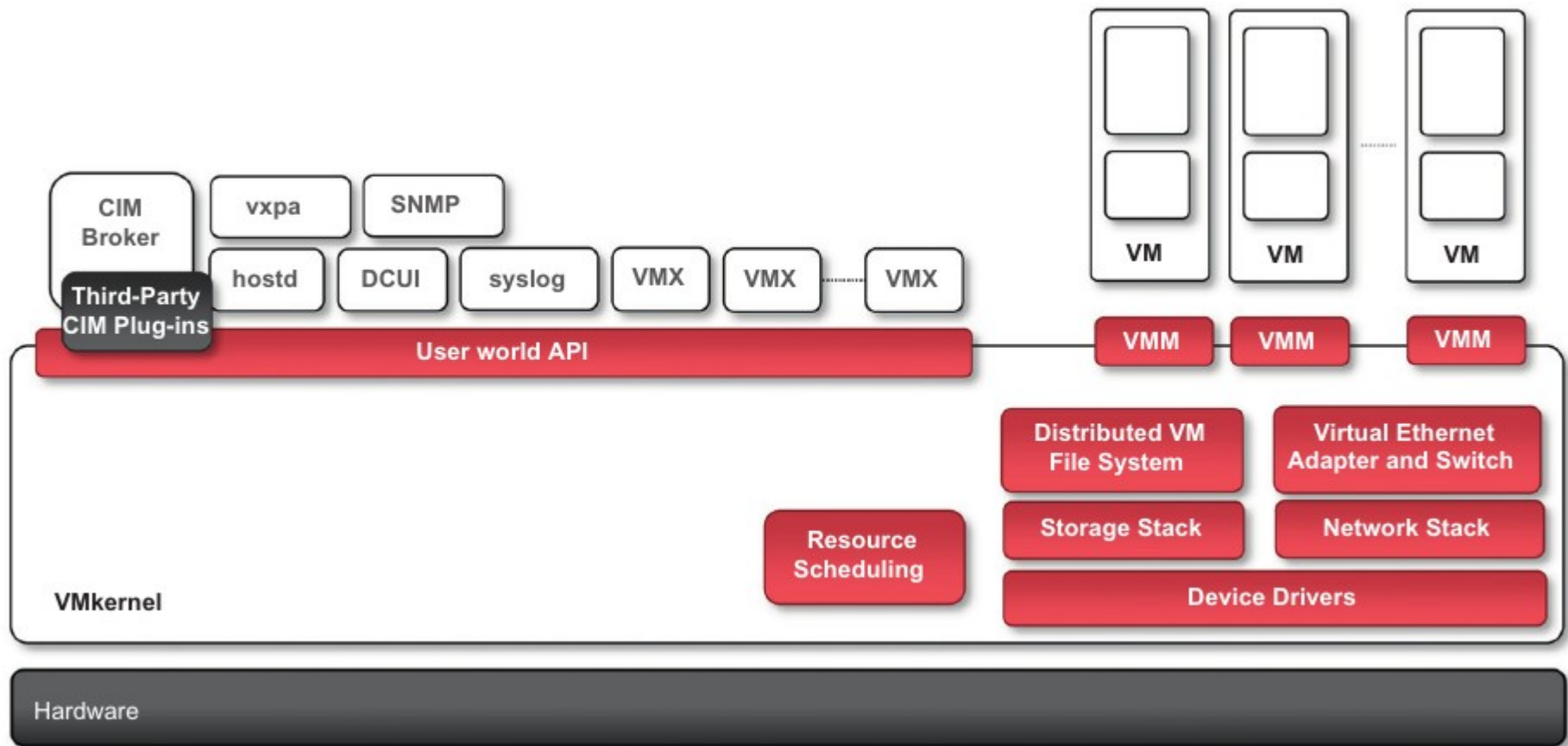
# Virtualización

## Arquitectura VMWare Workstation



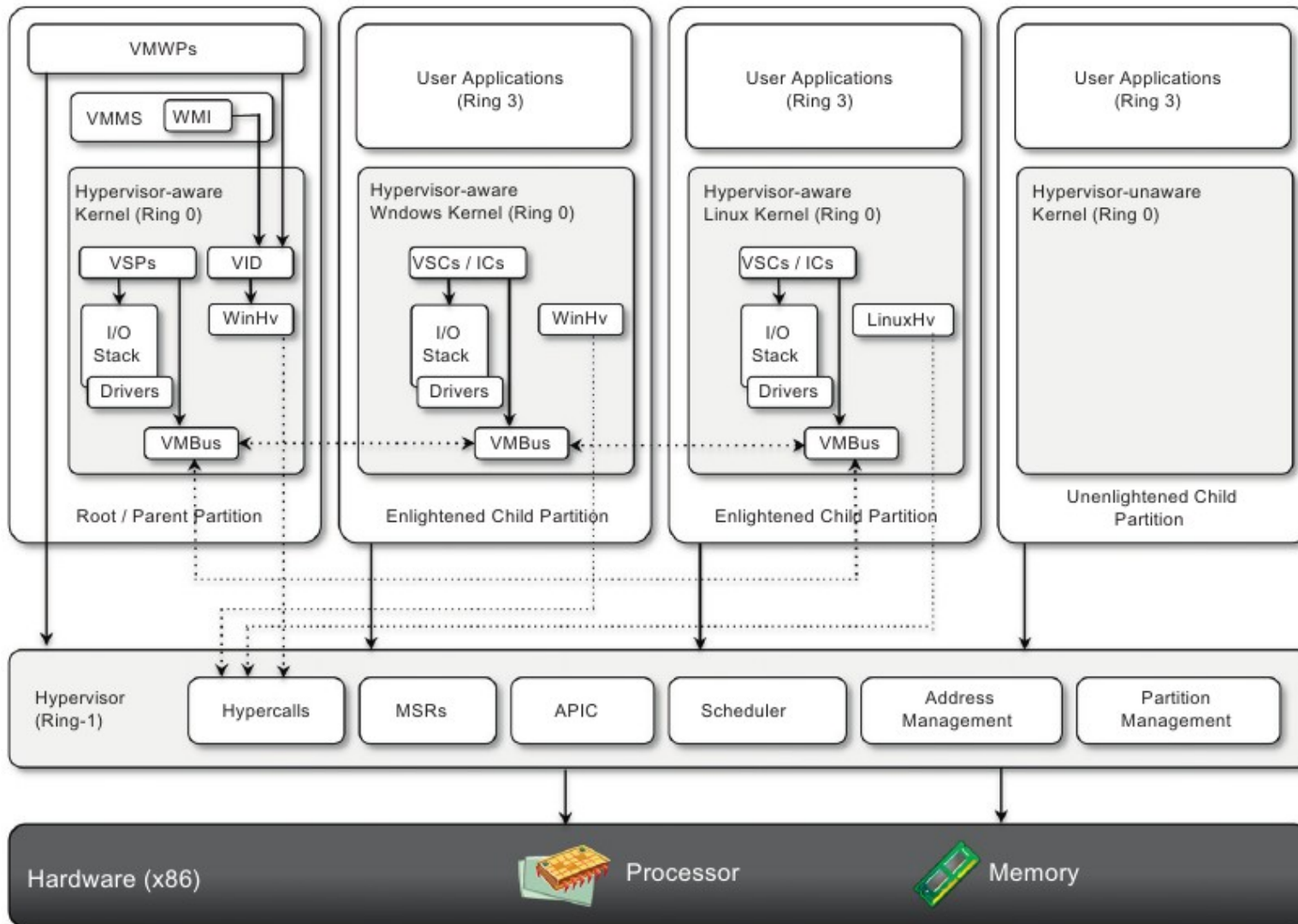
# Virtualización

## Arquitectura VMWare ESXi Server



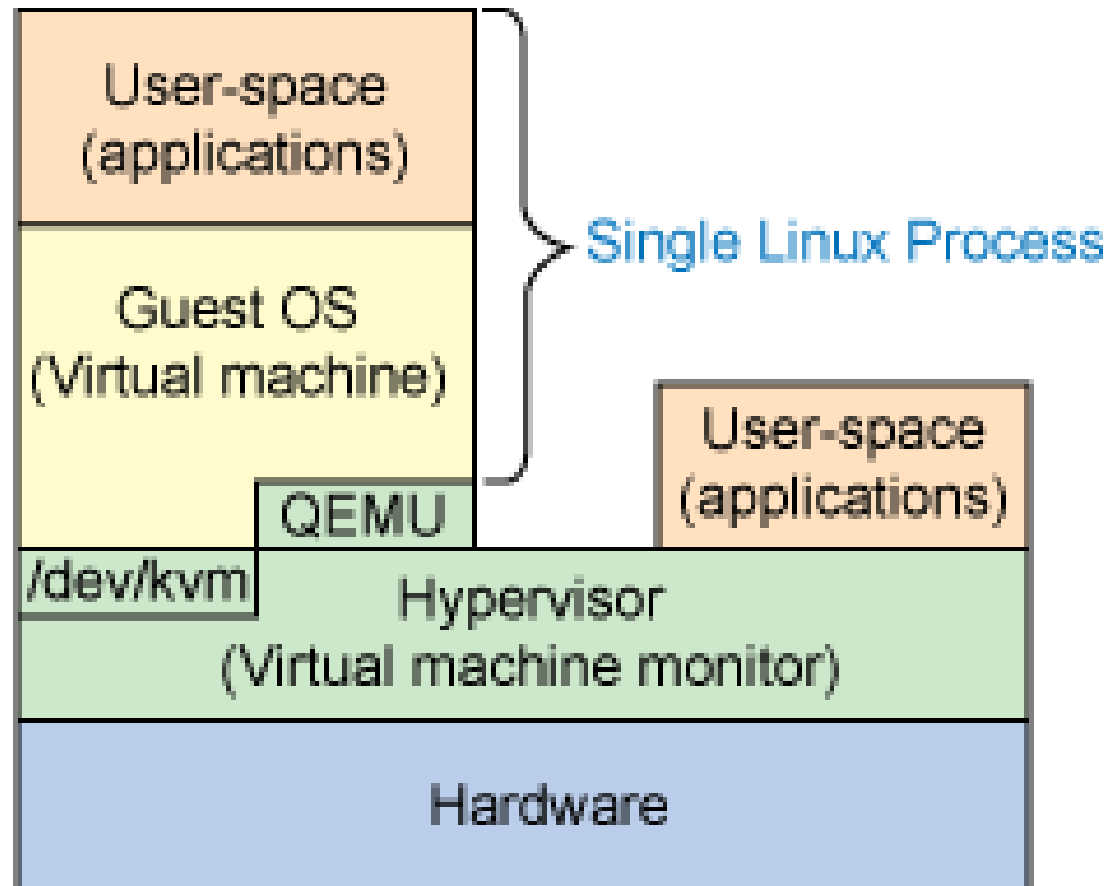
# Virtualización

## Arquitectura Microsoft Hyper-V



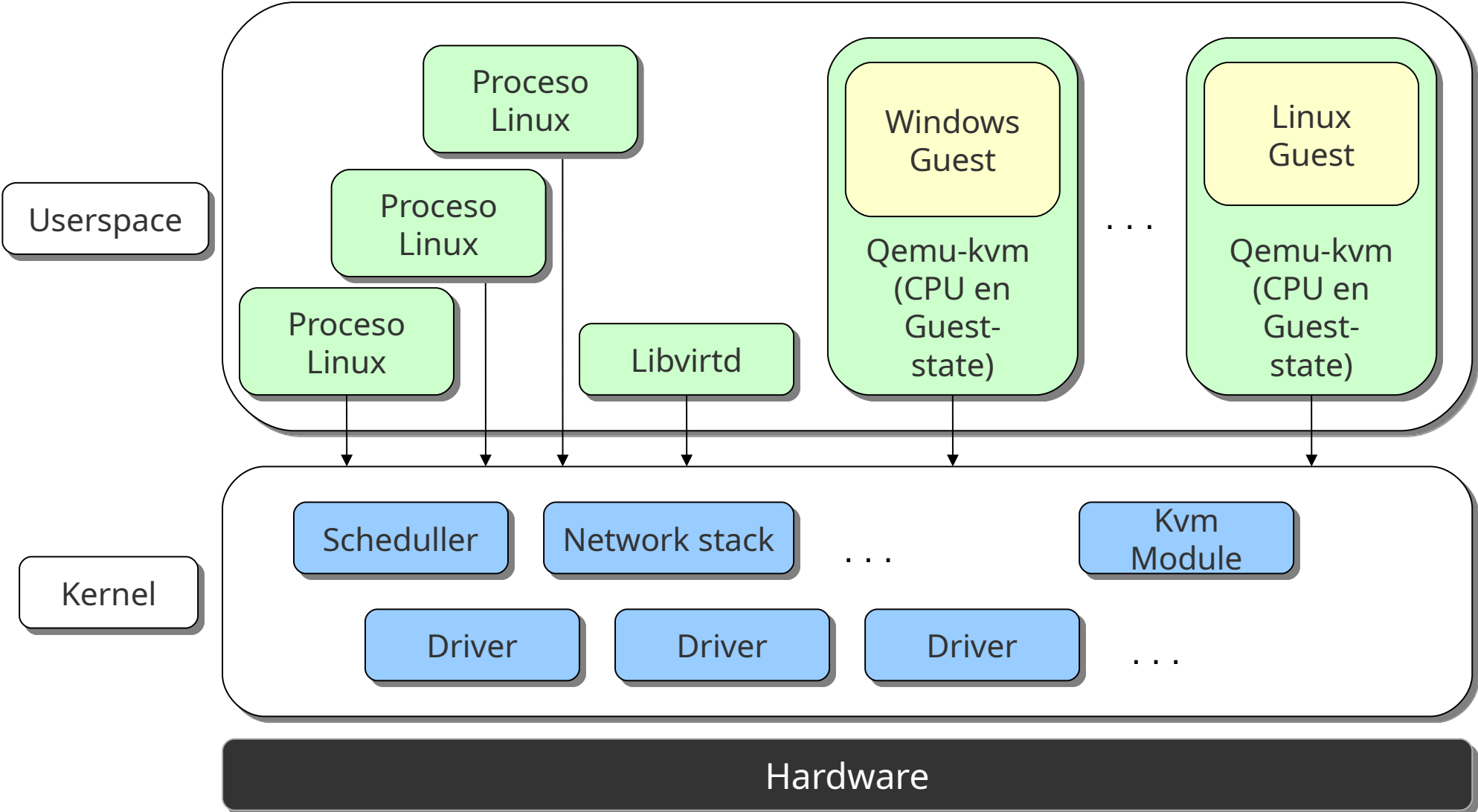
# Virtualización

## Arquitectura KVM



# Virtualización

## Arquitectura KVM



# Virtualización

## AWS EC2 Virtualization Types

|  |                        |
|--|------------------------|
|  | Bare-metal performance |
|  | Near-metal performance |
|  | Optimized performance  |
|  | Poor performance       |

Importance

Most → Least

|     | # | Tech | Type                | With                       |    |    |    |    |    |    |
|-----|---|------|---------------------|----------------------------|----|----|----|----|----|----|
|     | 1 | VM   | Fully Emulated      |                            | VS | VS | VS | VS | VS | VS |
| Old | 2 | VM   | Xen PV 3.0          | PV drivers                 | P  | P  | P  | P  | VS | VS |
|     | 3 | VM   | Xen HVM 3.0         | PV drivers                 | VH | P  | P  | P  | VS | VS |
|     | 4 | VM   | Xen HVM 4.0.1       | PVHVM drivers              | VH | P  | P  | P  | P  | VS |
|     | 5 | VM   | Xen AWS 2013        | PVHVM + SR-IOV(net)        | VH | VH | P  | P  | P  | VS |
|     | 6 | VM   | Xen AWS 2017        | PVHVM + SR-IOV(net, stor.) | VH | VH | VH | P  | P  | VS |
|     | 7 | VM   | AWS Nitro 2017      |                            | VH | VH | VH | VH | VH | VS |
| New | 8 | HW   | AWS Bare Metal 2017 |                            | H  | H  | H  | H  | H  | H  |
|     |   |      | Bare Metal          |                            | H  | H  | H  | H  | H  | H  |

VM: Virtual Machine. HW: Hardware.

VS: Virt. in software. VH: Virt. in hardware. P: Paravirt. Not all combinations shown.

SR-IOV(net): ixgbe/ena driver. SR-IOV(storage): nvme driver.

Fuente:

<http://www.brendangregg.com/blog/2017-11-29/aws-ec2-virtualization-2017.html>

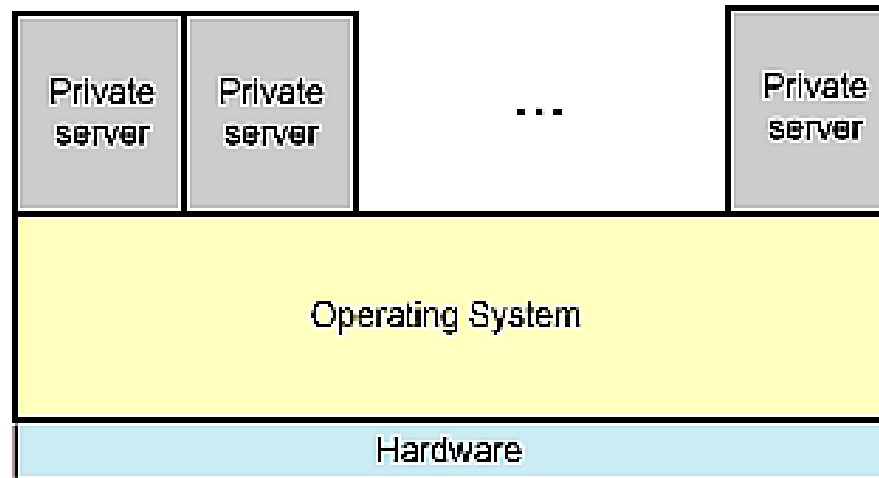
# Virtualización

## Virtualización a nivel de S.O.

(Containers, Zones, Jails, Virtual Private Servers...)

- **Guests comparten kernel con el anfitrión.**
- **Orientados a aislar servicios, no plataformas completas**
- **Basada en aislamiento de “user-spaces” - Separación de espacios de nombres.**

- FreeBSD Jails
- HP-UX Containers
- Solaris Zones
- Linux Vserver
- OpenVZ
- Virtuozzo
- LXC
- Docker
- VMWare ThinApp

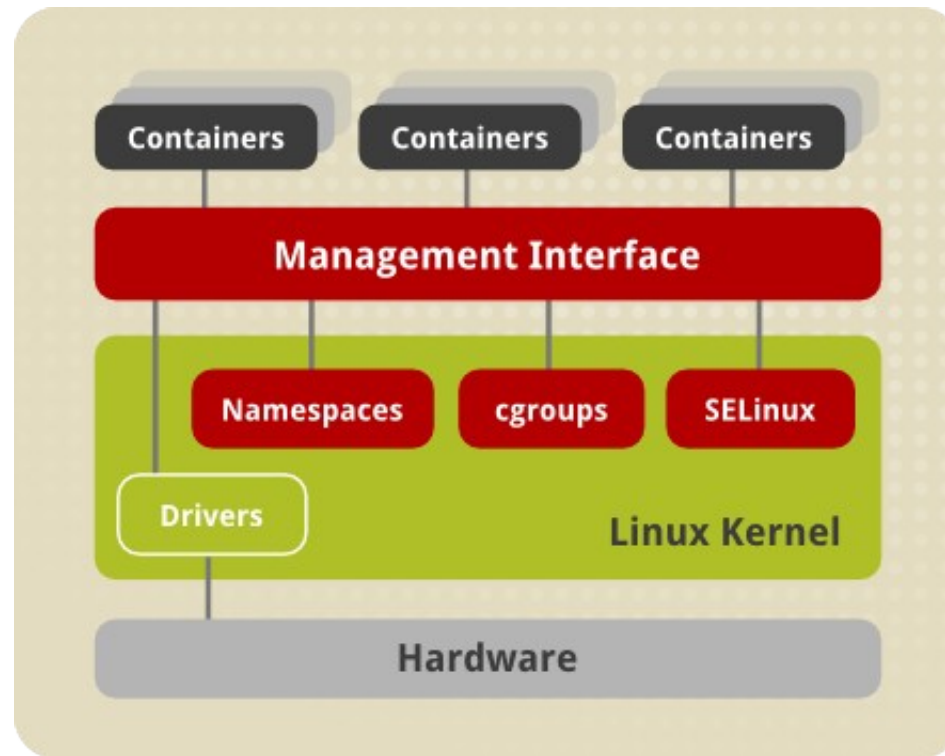


# Virtualización

## Virtualización a nivel de S.O. Linux Containers

### ● Namespaces:

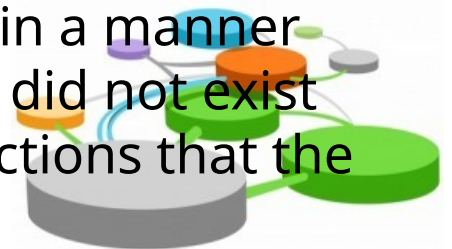
- Mount
- UTS
- IPC
- Process
- Network
- User



## Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures" Popek, G. J.; Goldberg, R. P. (1974)

- **The efficiency property.** All innocuous instructions are executed by the hardware directly, with no intervention at all on the part of the control program.
- **The resource control property.** It must be impossible for that arbitrary program to affect the system resources, i.e. memory, available to it; the allocator of the control program is to be invoked upon any attempt.
- **The equivalence property.** Any program K executing with a control program resident, with two possible exceptions, performs in a manner indistinguishable from the case when the control program did not exist and K had whatever freedom of access to privileged instructions that the programmer had intended.



## Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures" Popek, G. J.; Goldberg, R. P. (1974)

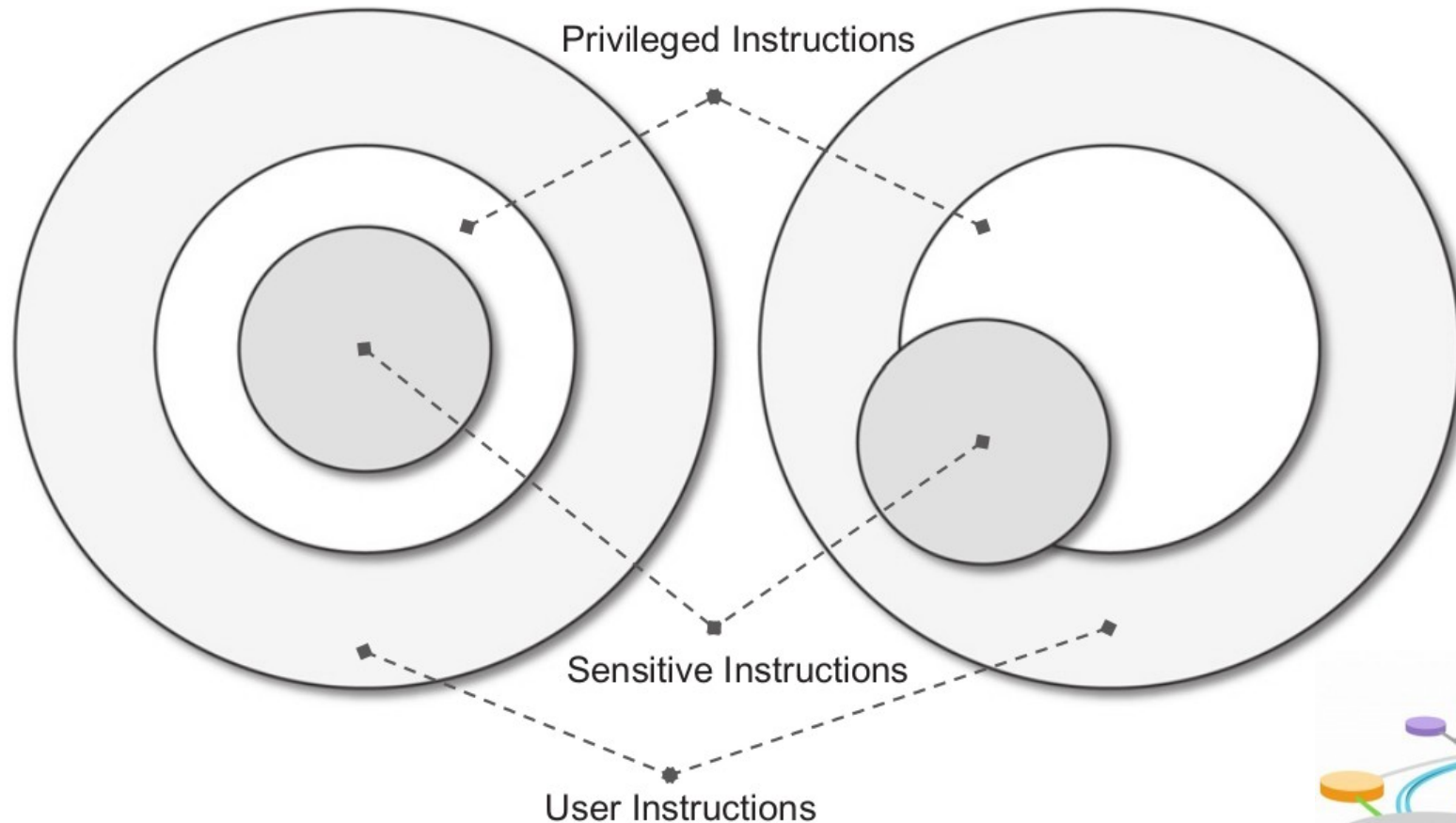
- **Instrucciones privilegiadas** (Privileged instructions)
  - Generan un trap si CPU en modo usuario, no en supervisor.
- **Instrucciones sensibles** (Sensitive instructions)
  - Control sensitive: Cambian la configuración/recursos del sistema (por ej., memoria)
  - Behavior sensitive: Dependen específicamente del hw para el que fue hecho



# Virtualización

## Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures" Popek, G. J.; Goldberg, R. P. (1974)



**¿Cuál CPU es virtualizable?**



## Propiedades de las VM

"Formal Requirements for Virtualizable Third Generation Architectures" Popek, G. J.; Goldberg, R. P. (1974)

- **THEOREM 1:** For any conventional third generation computer, a virtual machine monitor may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions.
- **THEOREM 2:** A conventional third generation computer is recursively virtualizable if it is: (a) virtualizable, and (b) a VMM without any timing dependencies can be constructed for it.
- **THEOREM 3:** A hybrid virtual machine monitor may be constructed for any conventional third generation machine in which the set of user sensitive instructions are a subset of the set of privileged instructions.

# Referencias

Mastering Cloud Computing, Foundations and Applications Programming. Rajkumar Buyya, Christian Vecchiola and S. Thamarai Selvi. Chapter 3: Virtualization. Morgan Kaufmann. 2013.

Real World Multicore Embedded Systems. Edited by: Bryon Moyer. Chapter 7: System Virtualization in Multicore Systems. David Kleidermacher. Newnes. 2013.

Cloud Computing, Theory and Practice. Dan C. Marinescu. Chapter 5 – Cloud Resource Virtualization. Morgan Kaufmann. 2013.

Formal requirements for virtualizable third generation architectures. Popek, G. J.; Goldberg, R. P. . Communications of the ACM. 1974

# Referencias

Virtual Linux

<https://web.archive.org/web/20120214062442/http://www.ibm.com/developerworks/linux/library/l-linuxvirt/>

Anatomy of a Linux hypervisor

<https://www.ibm.com/developerworks/linux/library/l-hypervisor/>

Virtio: An I/O virtualization framework for Linux

<https://www.ibm.com/developerworks/linux/library/l-virtio/>

Linux virtualization and PCI passthrough

<https://www.ibm.com/developerworks/linux/library/l-pci-passthrough/index.html>

Virtual networking in Linux

<https://www.ibm.com/developerworks/linux/library/l-virtual-networking/>

Anatomy of Security-Enhanced Linux (SELinux)

<https://www.ibm.com/developerworks/linux/library/l-selinux/>

Understanding Full Virtualization, Paravirtualization, and Hardware Assist - VMware White Paper

[https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware\\_paravirtualization.pdf](https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/VMware_paravirtualization.pdf)