

Design + Implementation + Administration

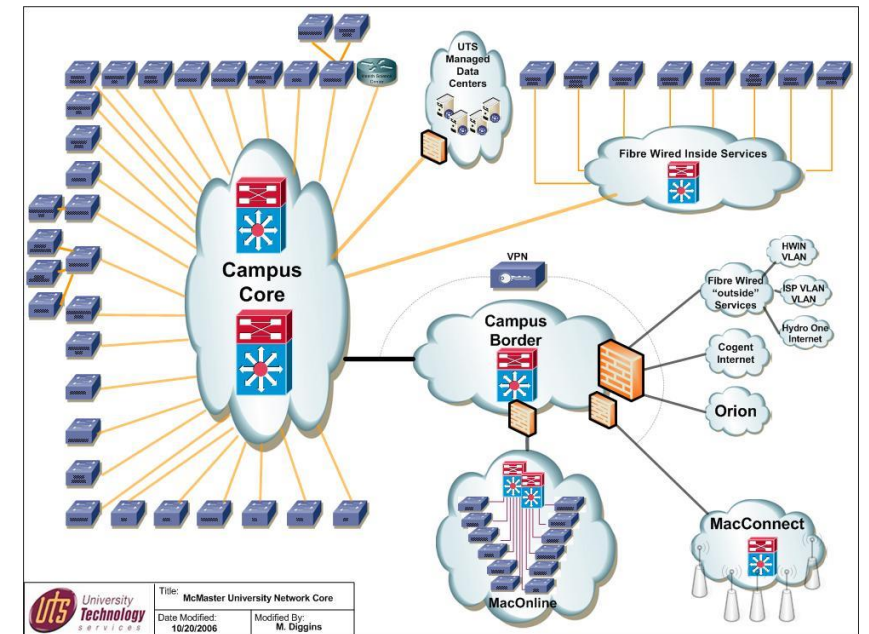
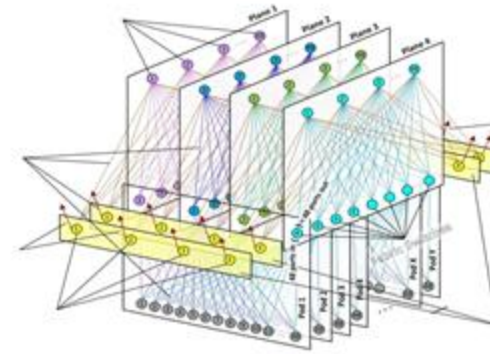


Large Systems: Design + Implementation

2024-2025

➤ Week2-L3: Virtualization- Part 3

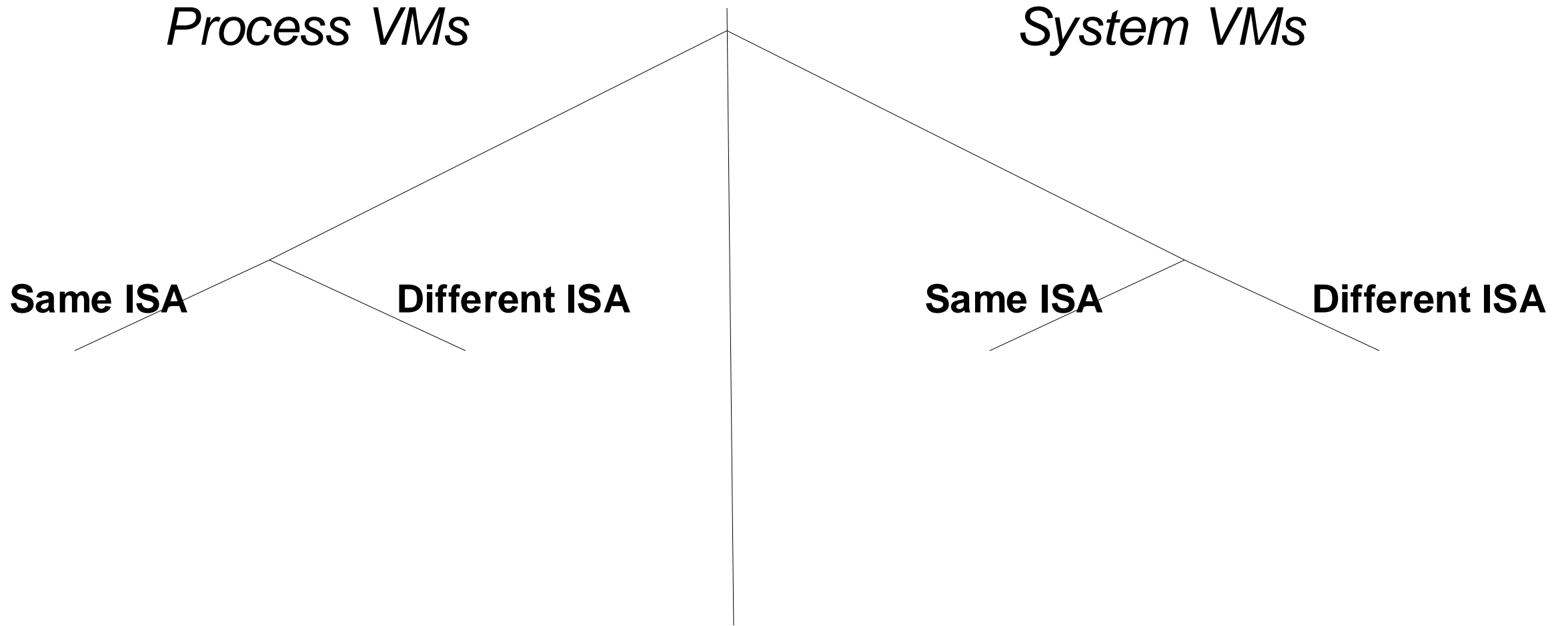
Shashikant Ilager
shashikantilager.com



Recap

- Difference **Process VM** and **System VM** in terms of the interface virtualized?

Taxonomy

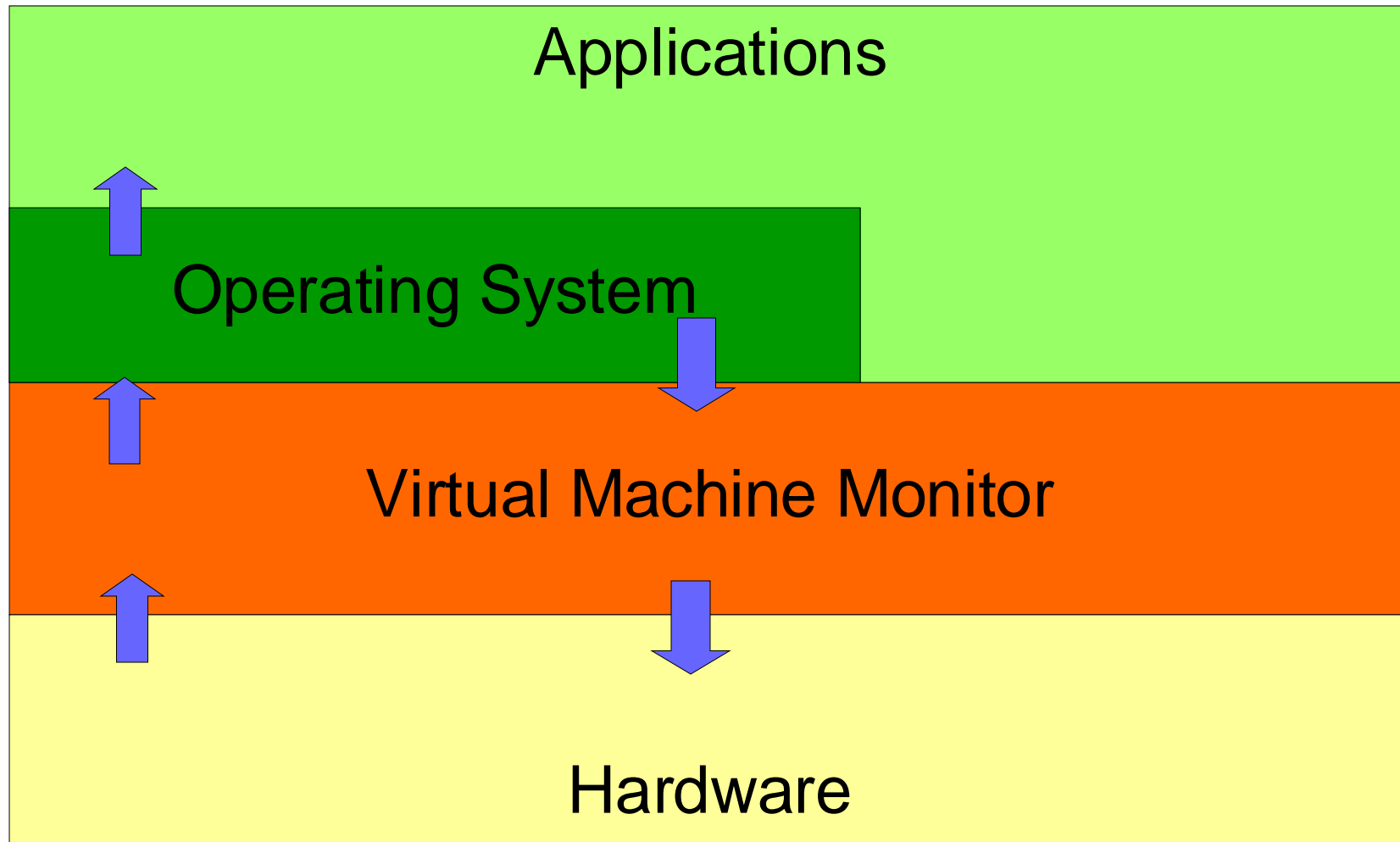


Java, Xen and Android emulation

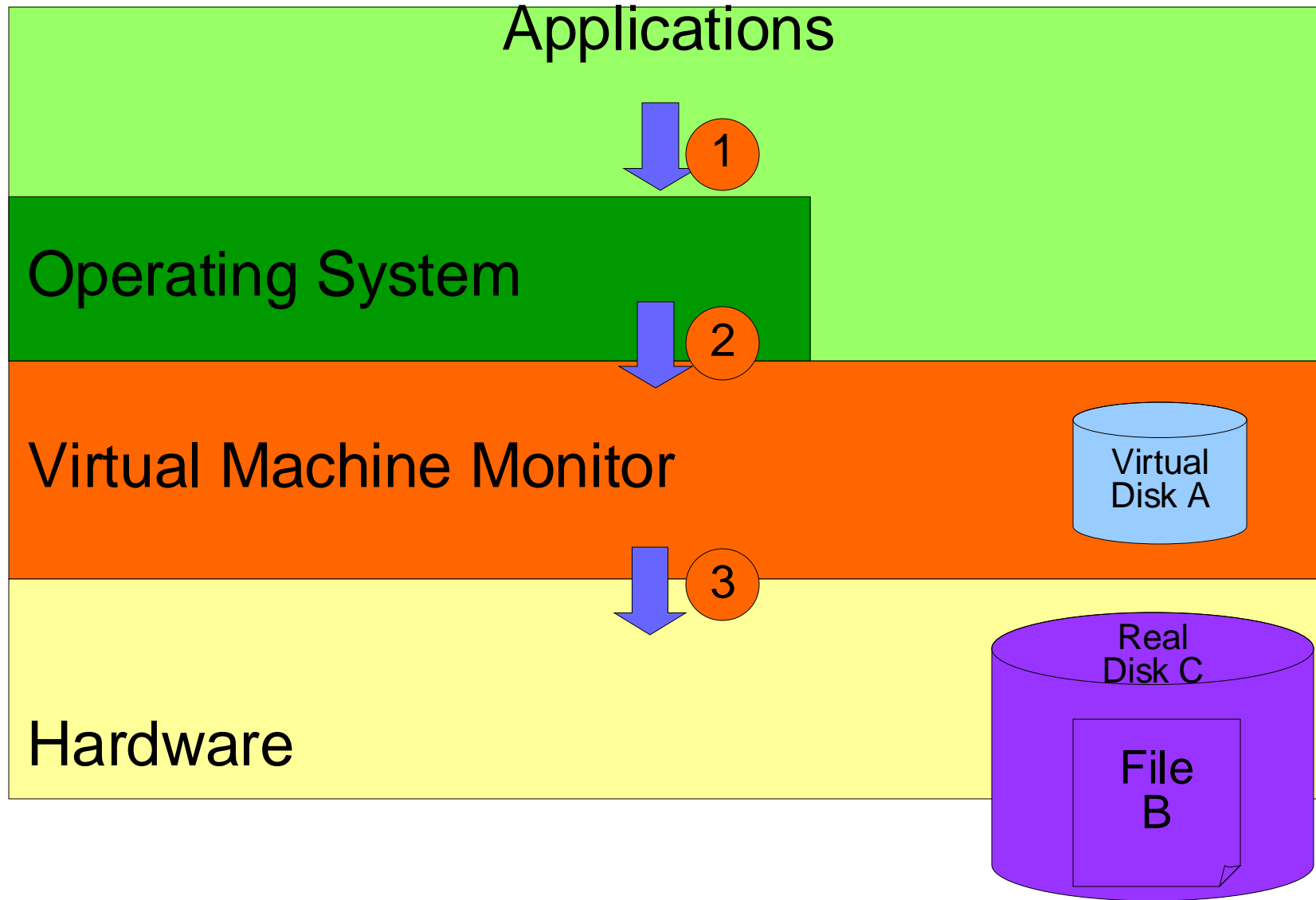
Recap

- What is Intel VT-x?
- What problem does it solve?

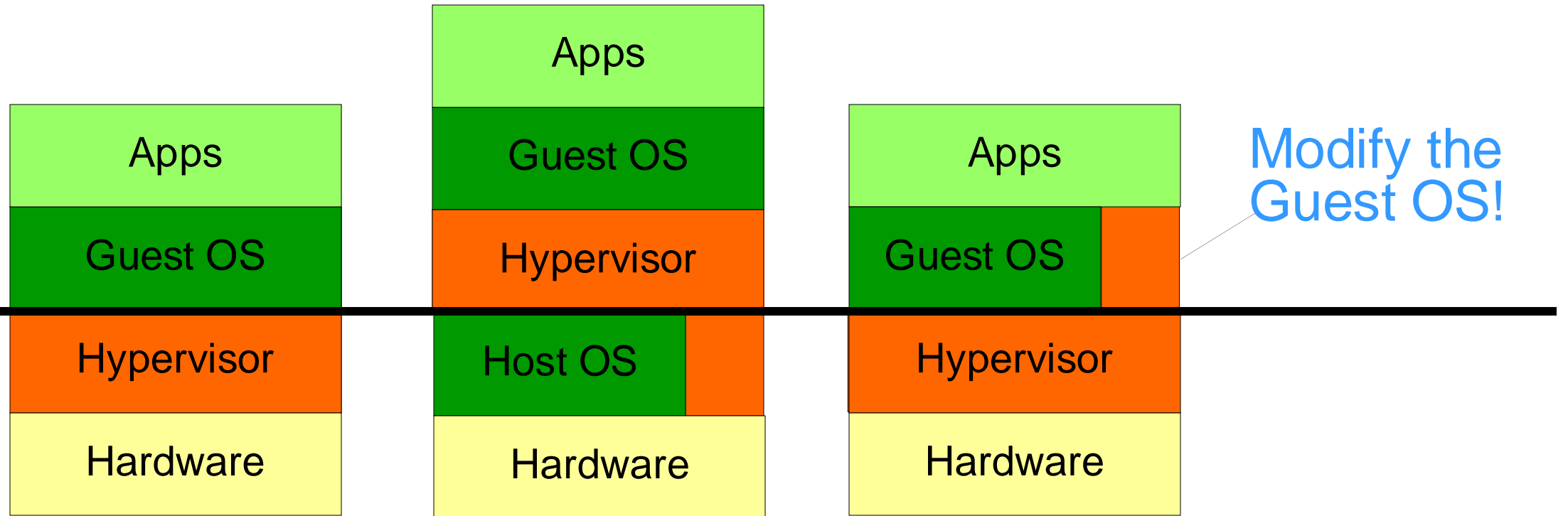
Recap: App Scheduling



Recap: Example- Reading from disk



Native, Hosted, Paravirtualized VMs

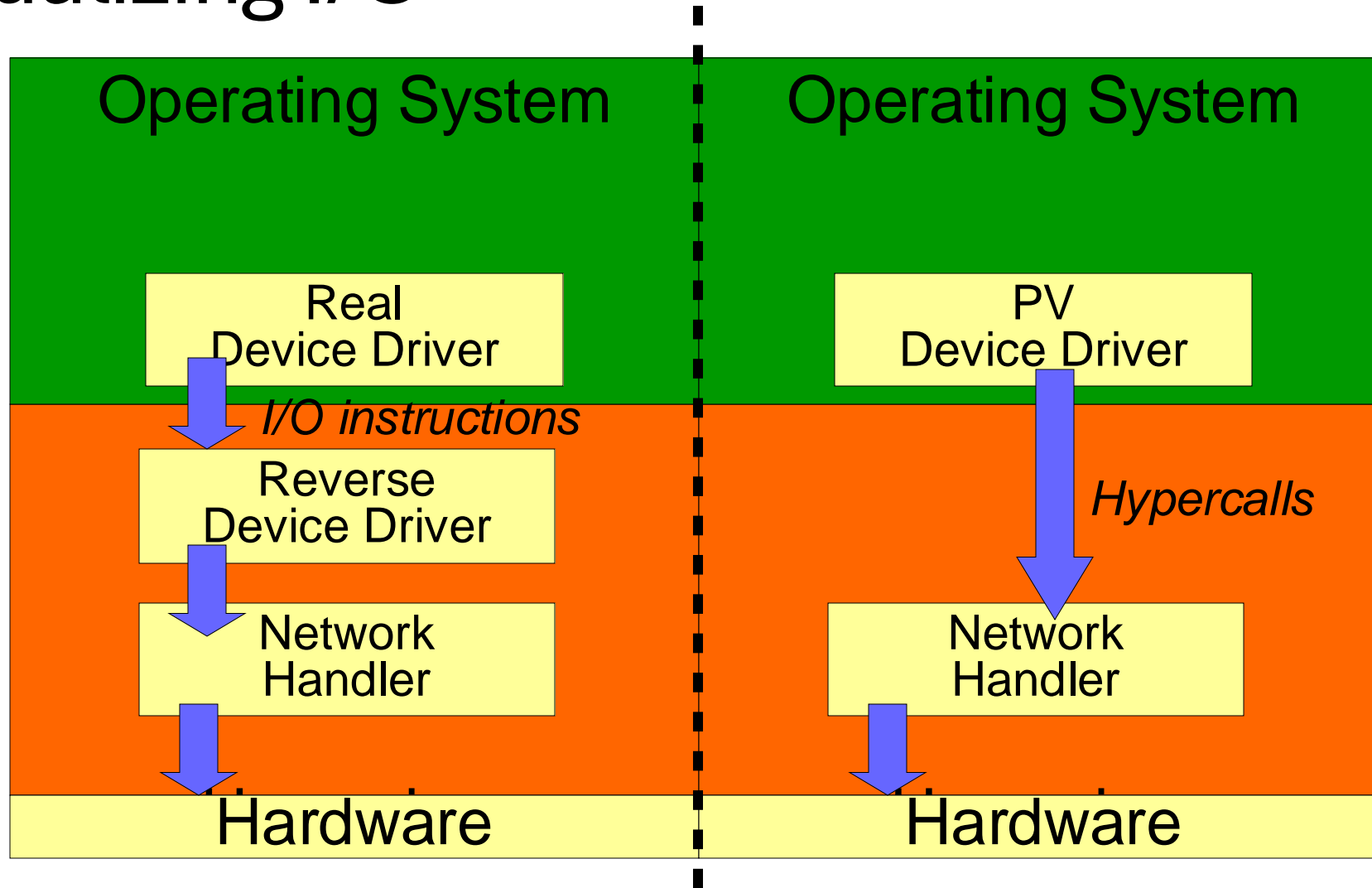


Paravirtualization

- System VMs can be faster when Guest OS can be modified for virtualization
- Showcased in Xen Project, also used in KVM
- Modified
 - Linux
 - Windows XP
- Near native performances



Paravirtualizing I/O



Guest OS knows it is not running on real hardware

Xen Evolution

- **Problems:**
 - only open-source OSes can be modified
 - Xen implementation tricks not on x86-64
- **New approach:** Start from Full virtualization with Hardware Support (e.g. VT-x)
- Apply Paravirtualization in areas where speed can be gained:
 - Disk and network I/O
 - Interrupts and timers
 - Emulated motherboard, legacy boot
 - Privileged instructions, page tables

Xen Mode: HVM

Poor Performance
 Scope for Improvement
 Optimal Performance

P = Paravirtualized
 VS = Software Virtualized (QEMU)
 VH = Hardware Virtualized



Xen Mode: PV

Poor Performance
 Scope for Improvement
 Optimal Performance

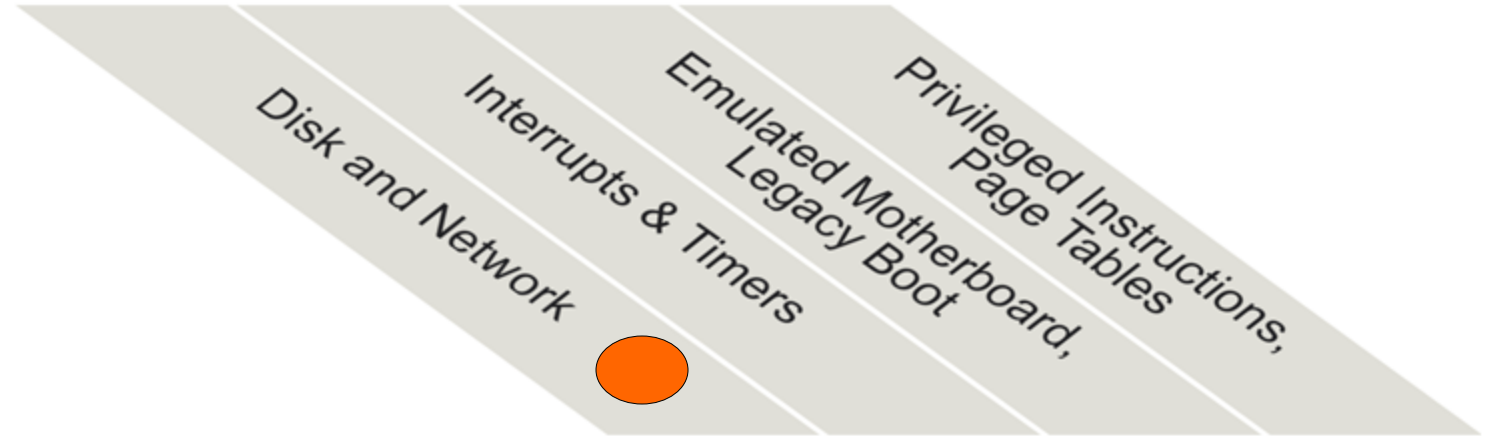
P = Paravirtualized
 VS = Software Virtualized (QEMU)
 VH = Hardware Virtualized

			<div>Privileged Instructions, Page Tables</div> <div>Emulated Motherboard, Legacy Boot</div> <div>Interrupts & Timers</div> <div>Disk and Network</div>			
Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
PV	PV		P	P	P	P

Xen Mode: HVM + PV Drivers

Poor Performance
 Scope for Improvement
 Optimal Performance

P = Paravirtualized
 VS = Software Virtualized (QEMU)
 VH = Hardware Virtualized

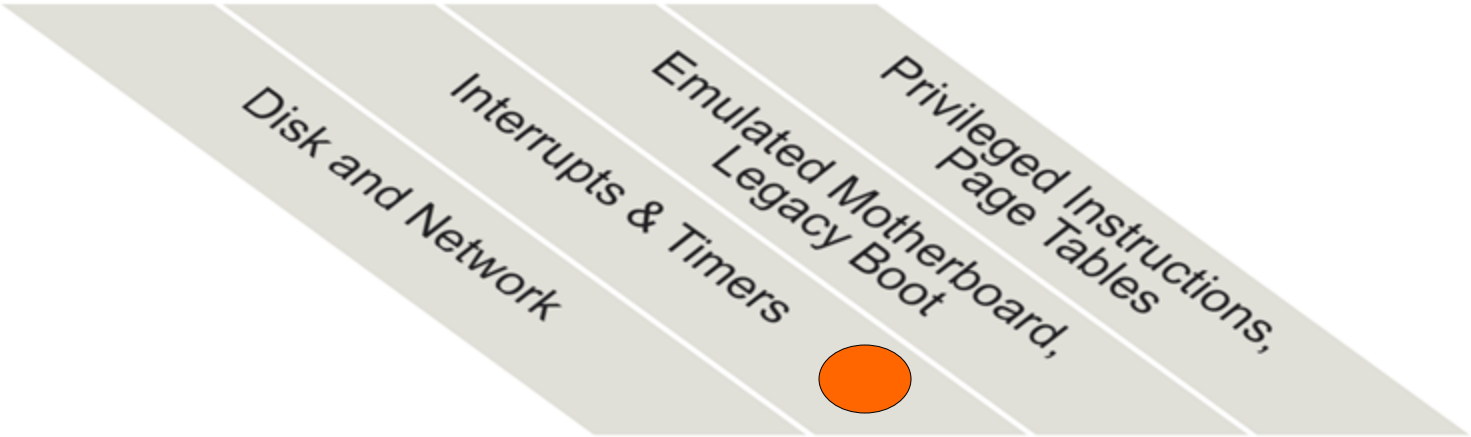


Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	P	VS	VS	VH
PV	PV		P	P	P	P

Xen Mode: PVHVM Drivers

- Poor Performance
- Scope for Improvement
- Optimal Performance

P = Paravirtualized
VS = Software Virtualized (QEMU)
VH = Hardware Virtualized



Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	P	VS	VS	VH
PVHVM	HVM	PVHVM Drivers	P	P	VS	VH
PV	PV		P	P	P	P

Xen Mode: PVH

	Poor Performance
	Scope for Improvement
	Optimal Performance

P = Paravirtualized

VS = Software Virtualized (QEMU)

VH = Hardware Virtualized

			<div> <div>Privileged Instructions, Page Tables</div> <div>Emulated Motherboard, Legacy Boot</div> <div>Interrupts & Timers</div> <div>Disk and Network</div> </div>			
Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	P	VS	VS	VH
PVHVM	HVM	PVHVM Drivers	P	P	VS	VH
PVH	PV	pvh=1	P	P	P	VH
PV	PV		P	P	P	P
			<div>Windows</div> <div>Linux, BSDs, ...</div>			

KVM

	Poor Performance
	Scope for Improvement
	Optimal Performance

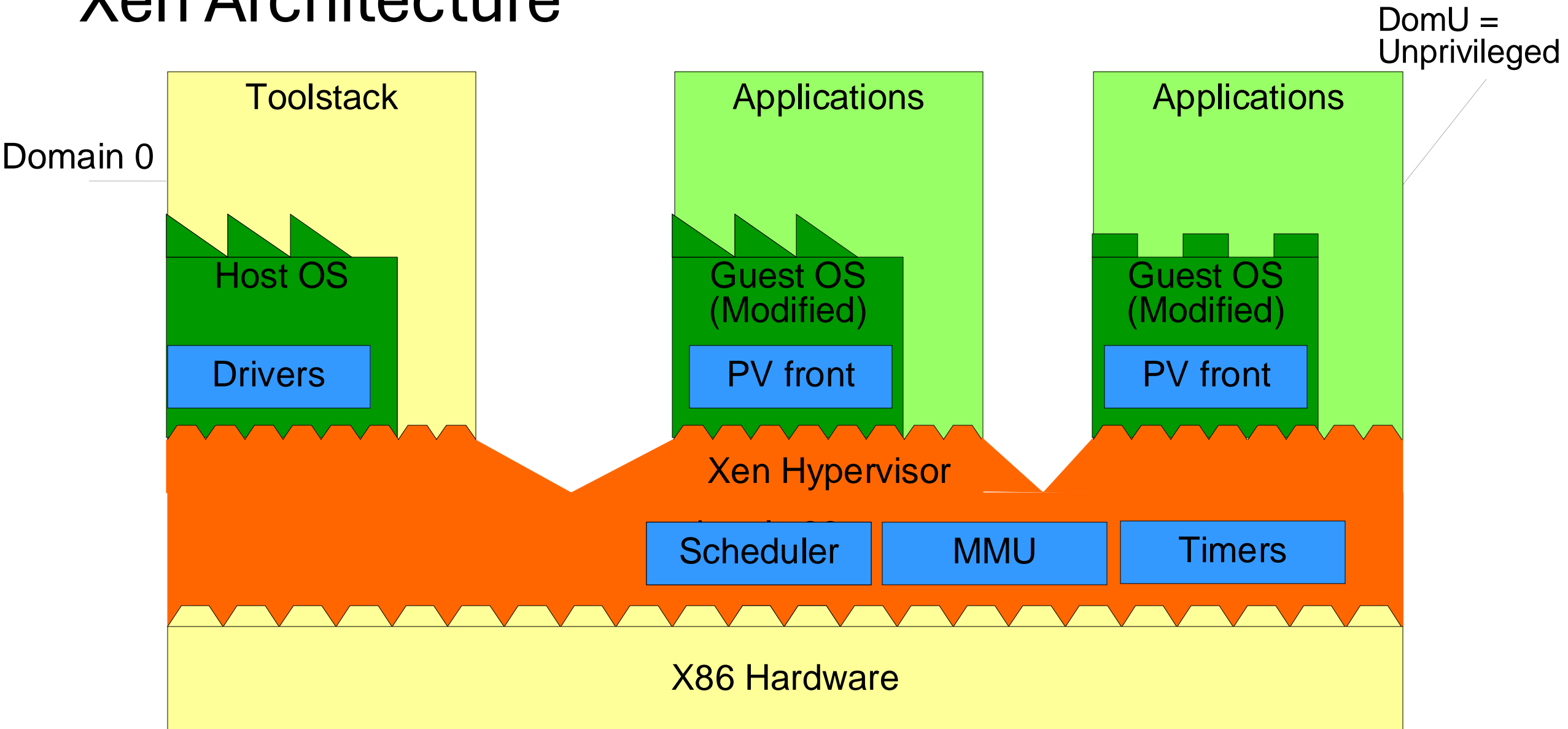
P = Paravirtualized
 VS = Software Virtualized (QEMU)
 VH = Hardware Virtualized

			<div>Privileged Instructions, Page Tables</div> <div>Emulated Motherboard, Legacy Boot</div> <div>Interrupts & Timers</div> <div>Disk and Network</div>			
Shortcut	Mode	With				
HVM / Fully Virtualized	HVM		VS	VS	VS	VH
HVM + PV drivers	HVM	PV Drivers	P	VS	VS	VH
KVM	HVM		P	VS	VS P	VH
PVHVM	HVM	PVHVM Drivers	P	P	VS	VH
PVH	PV	pvh=1	P	P	P	VH
PV	PV		P	P	P	P

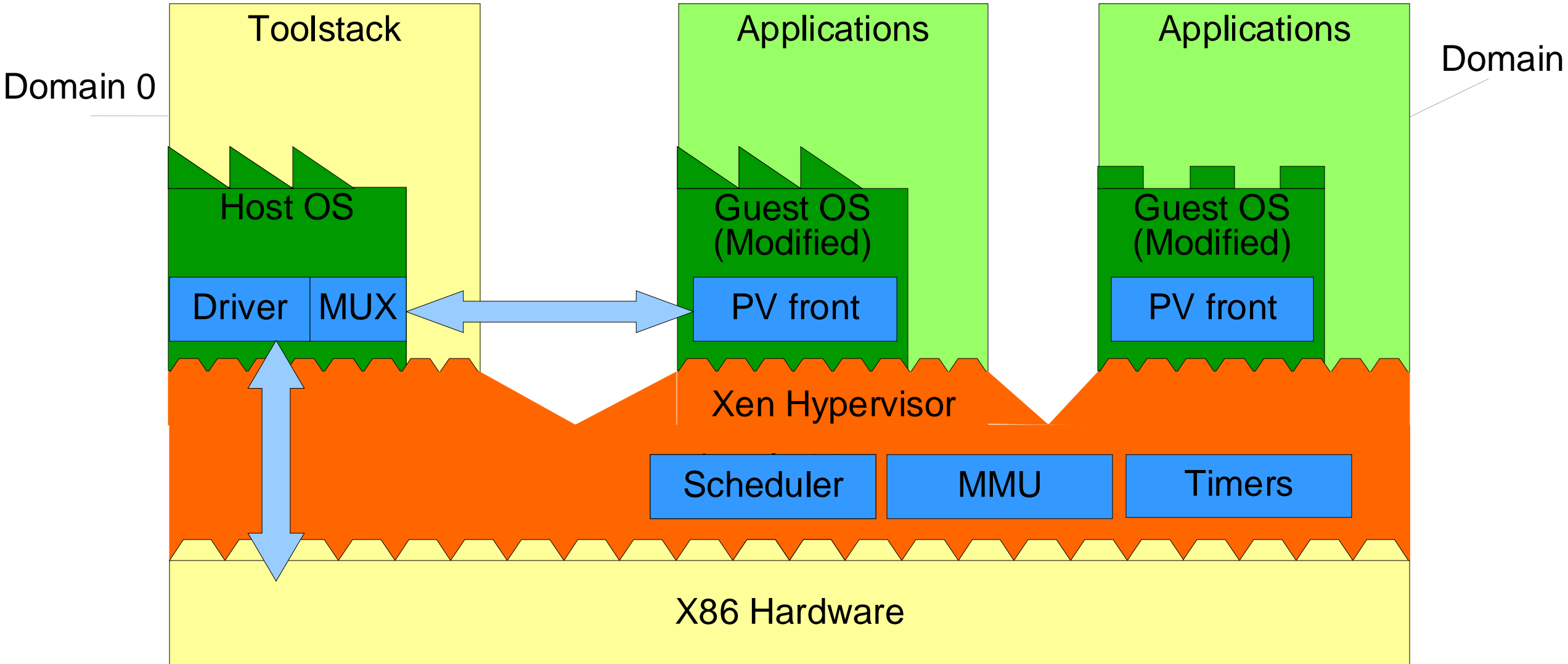
KVM Architecture

- The kernel is both OS and hypervisor
- The guest kernel uses QEMU to emulate hardware
- Guest + QEMU talk to kvm.ko kernel module

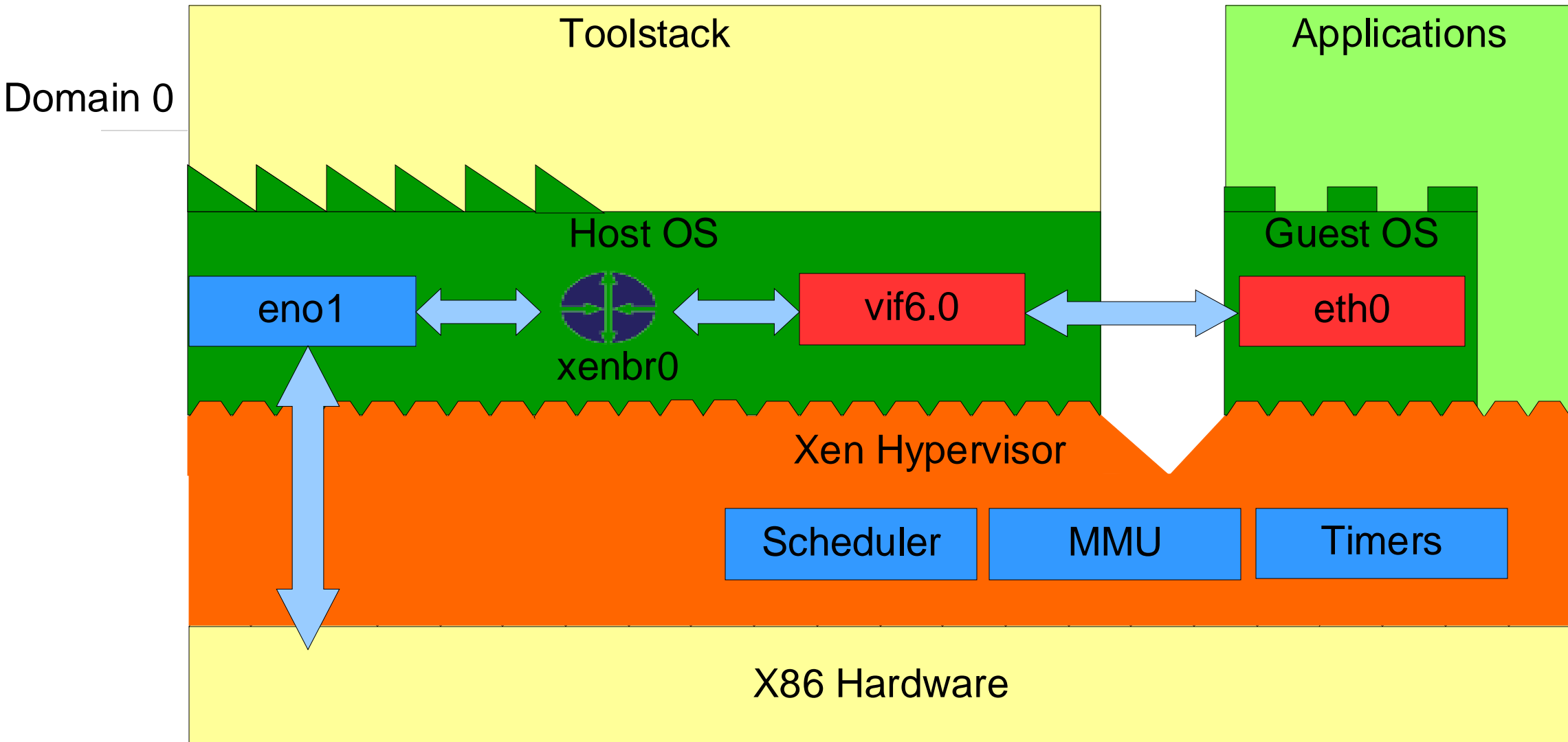
Xen Architecture



Xen Architecture



Xen Virtual Devices

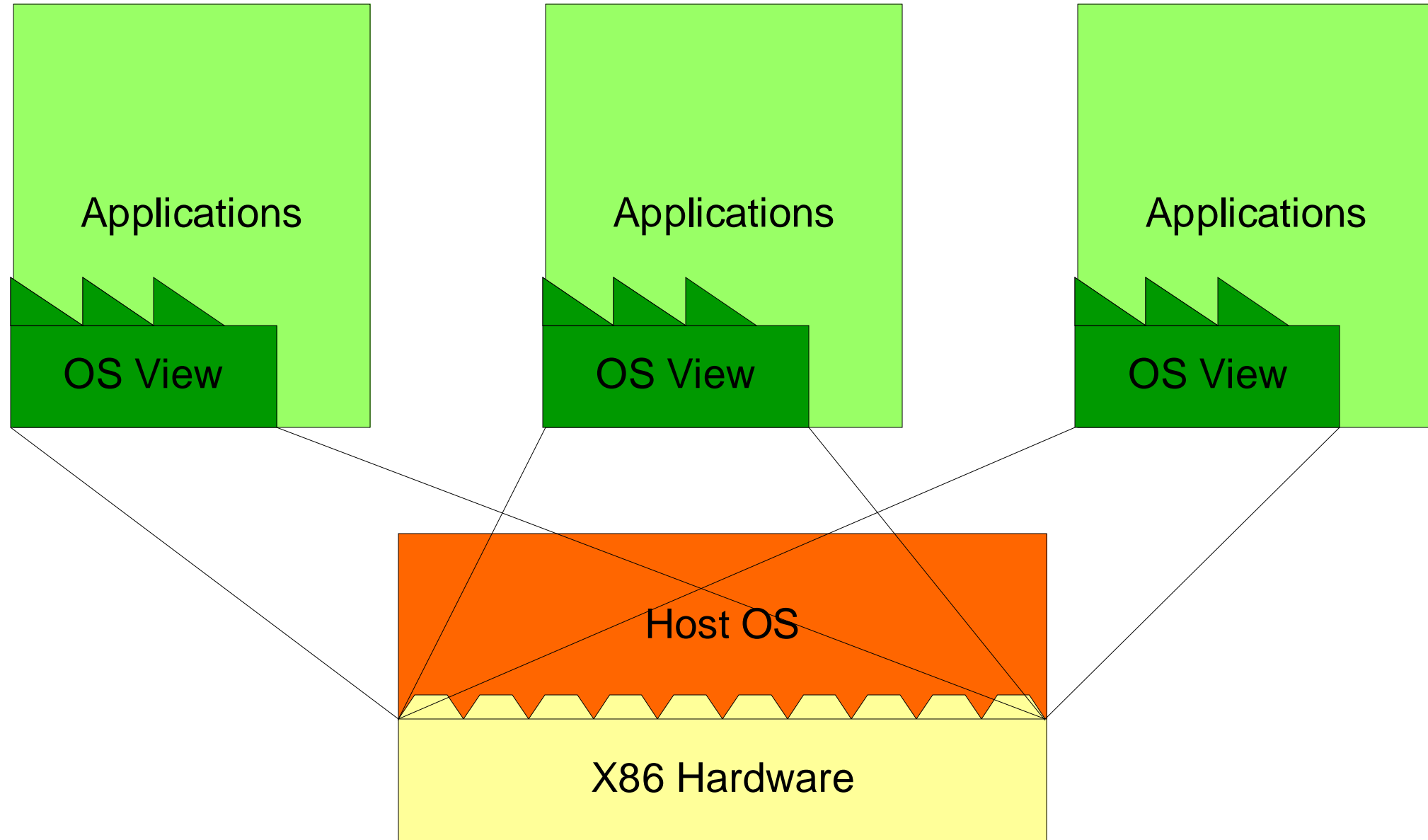


Operating-System Level Virtualization

- In between System VM and Process VM
- **Not** System VM:
 - Cannot choose OS
- **Not** Process VM:
 - Multiple processes, not isolated
- As if multiple instances of the same OS are running on the same machine
 - Example: **Linux Containers**
 - cf. Docker



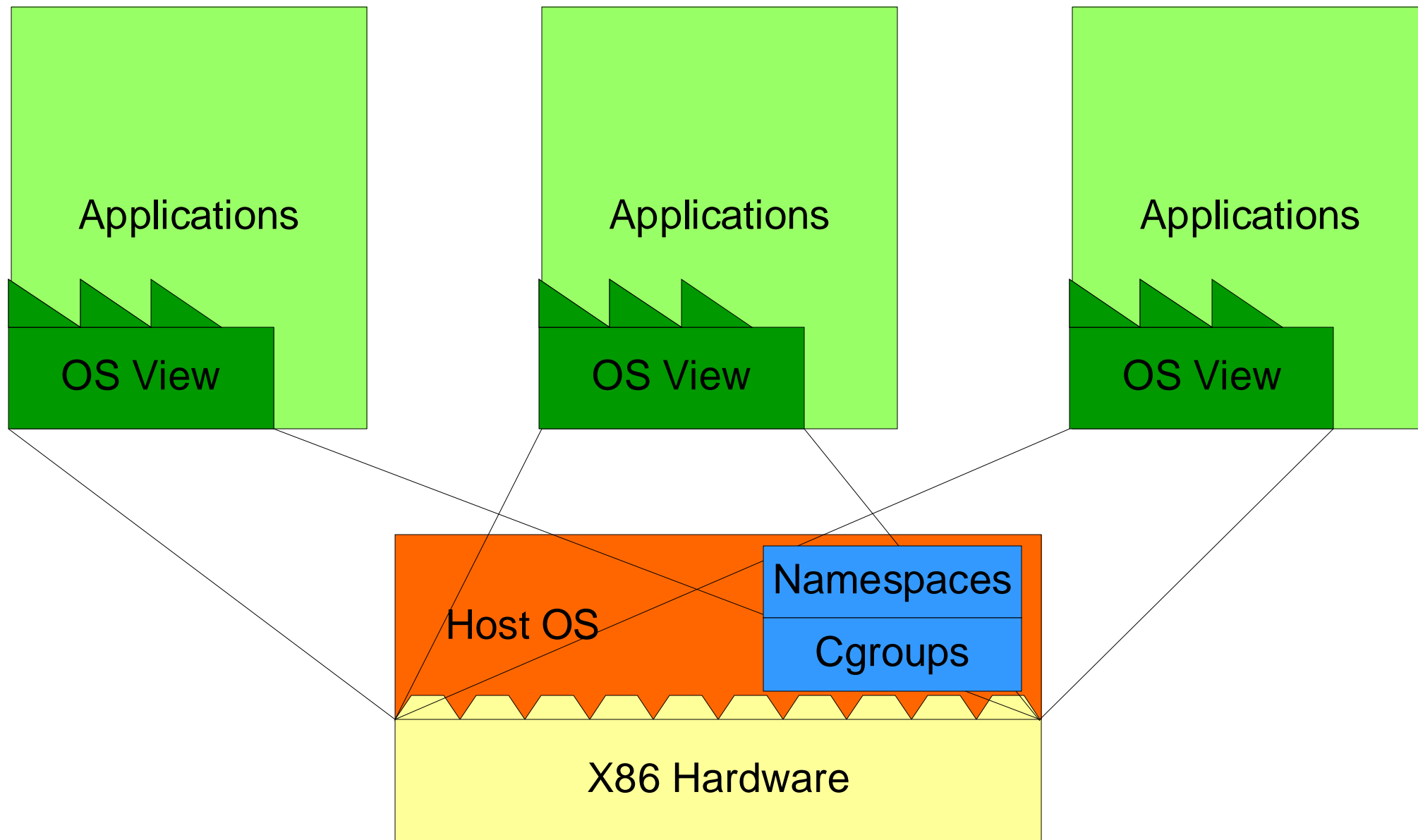
Linux Containers



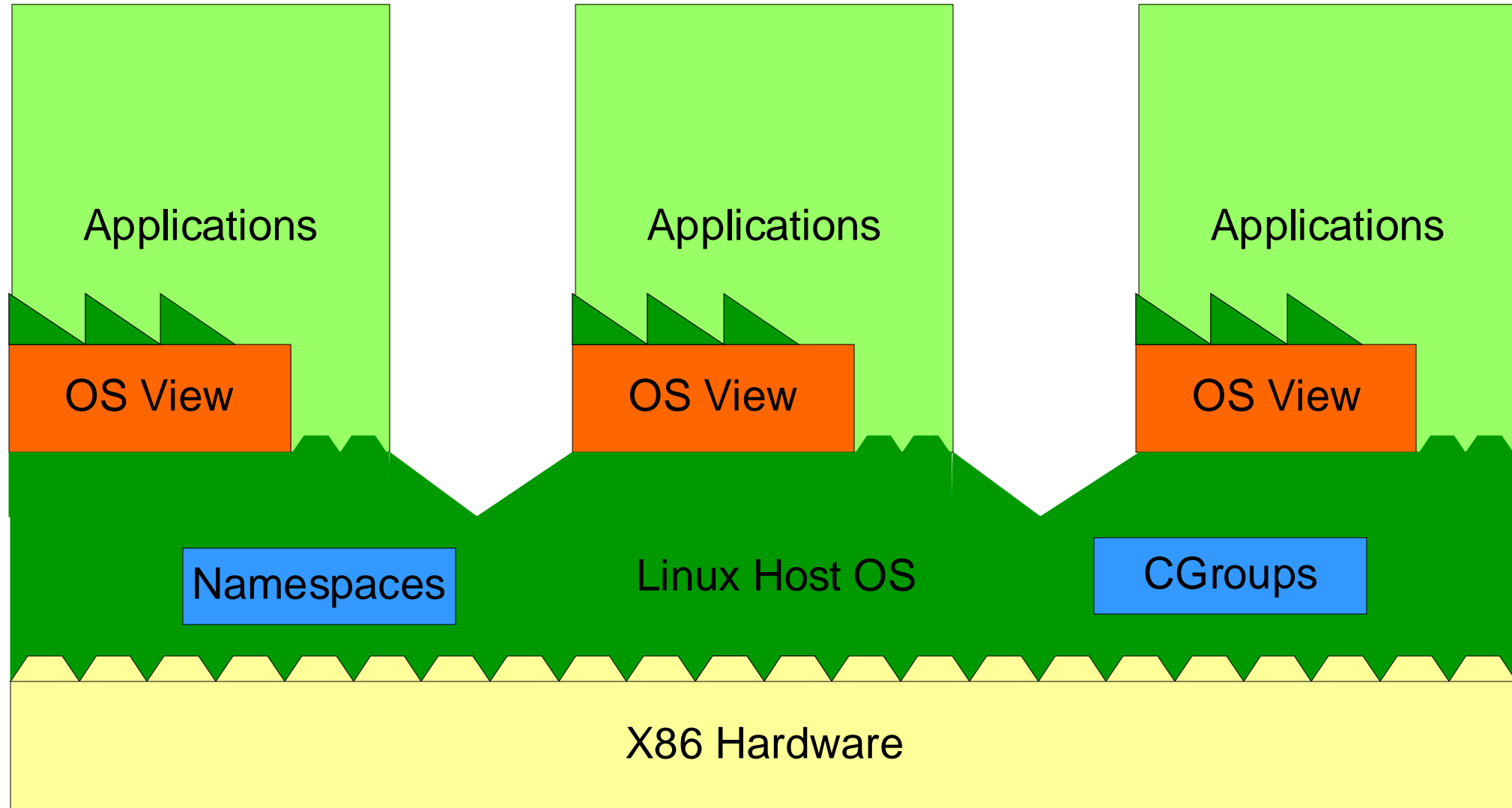
Containers

- Only **one** kernel is installed, and the hardware is not virtualized.
- Rather, the operating system is virtualized, providing processes within a container with **the impression** that they are the **only** processes on the system.
- One or more containers can be created, and each can have its **own** applications, network addresses, user accounts, and so on.

Linux Containers



Linux Containers



Containers Continued...