Introduction to Xen

Kun Suo

UCCS

Schedule

- Monday: Introduction
- Tuesday: Xen
- Wednesday: Xen
- Thursday: Docker
- Friday: Conclusion/homework solution

Topic Today

- What is Xen ?
- Xen history
- Xen design philosophy
- How Xen virtualize resources ?

What is Xen ?

What is Xen ?

Xen is a hypervisor using a microkernel design, providing services that allow multiple computer operating systems to execute on the same computer hardware concurrently.

—— Wikipedia

Xen history

University of Cambridge

Xen History

2002, Xen starts from University of Cambridge

2003, Xen and the art of virtualization is published on SOSP 03

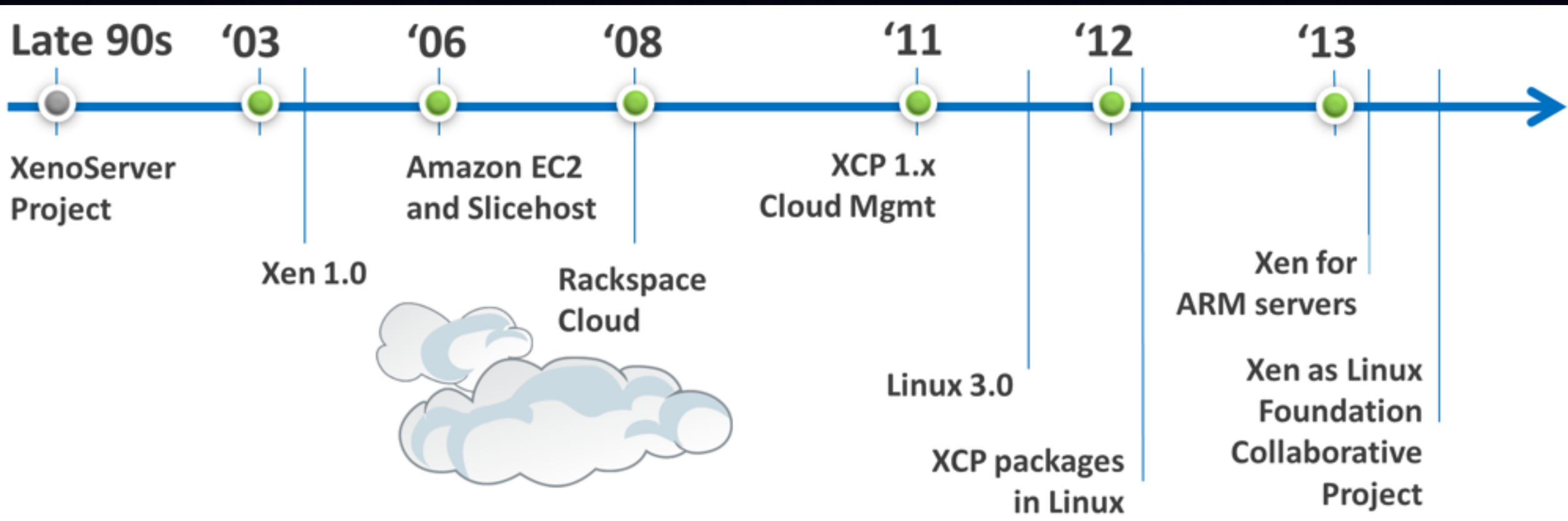
2004.04, Xen 1.0 is released

2004.11, Xen 2.0 is released

2005.12, Xen 3.0 is released

2010, Xen 4.0 is released

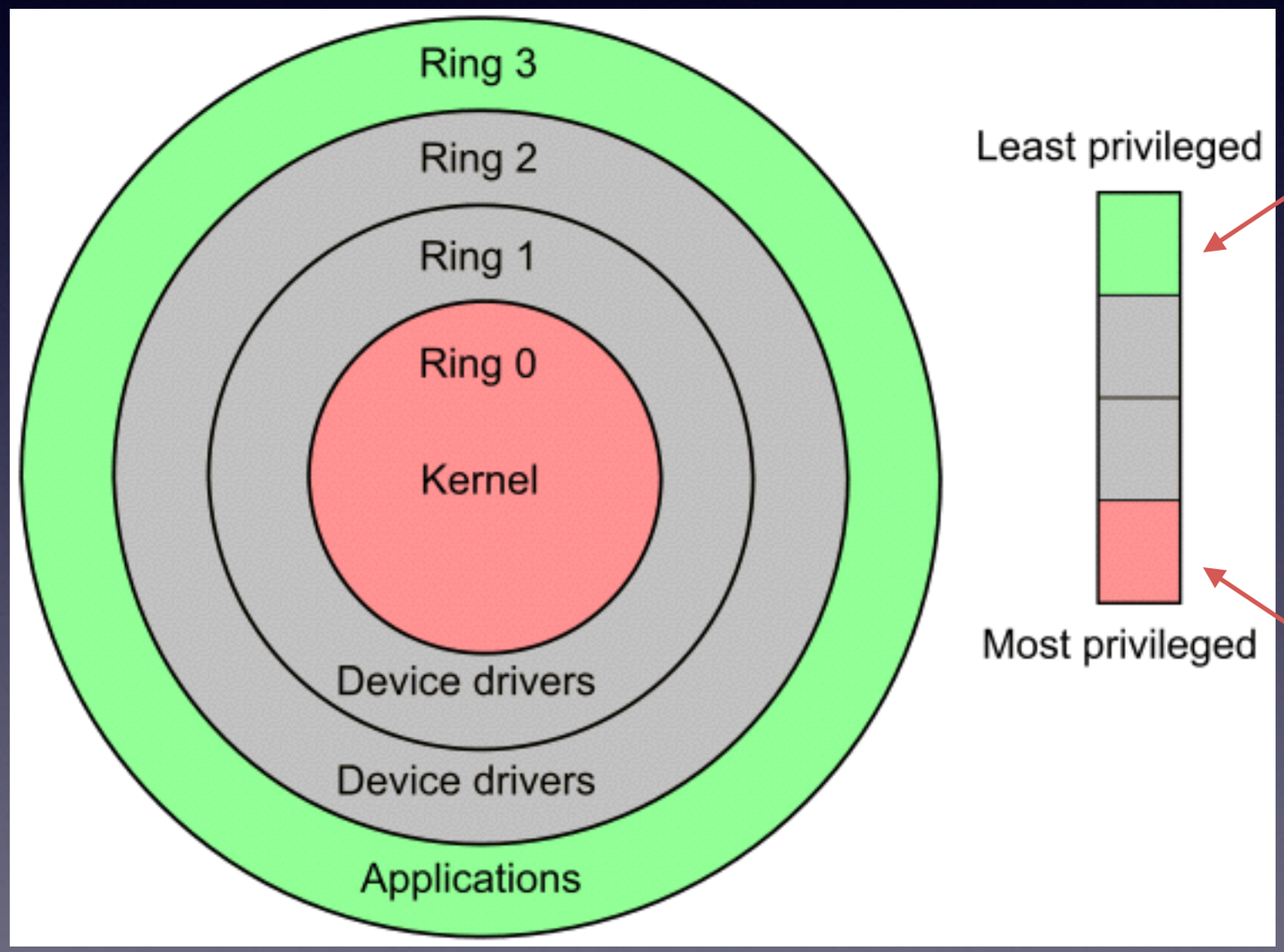
Xen History

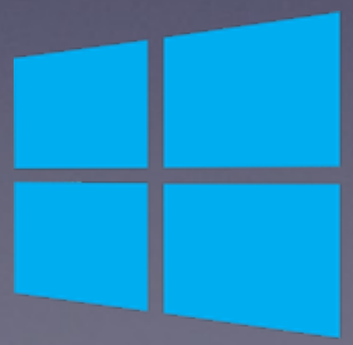

Late 90s
'03
'06
'08
'11
'12
'13
XenoServer Project
Xen 1.0
Amazon EC2 and Slicehost
Rackspace Cloud
XCP 1.x Cloud Mgmt
Linux 3.0
XCP packages in Linux
Xen for ARM servers
Xen as Linux Foundation Collaborative Project

Xen design philosophy

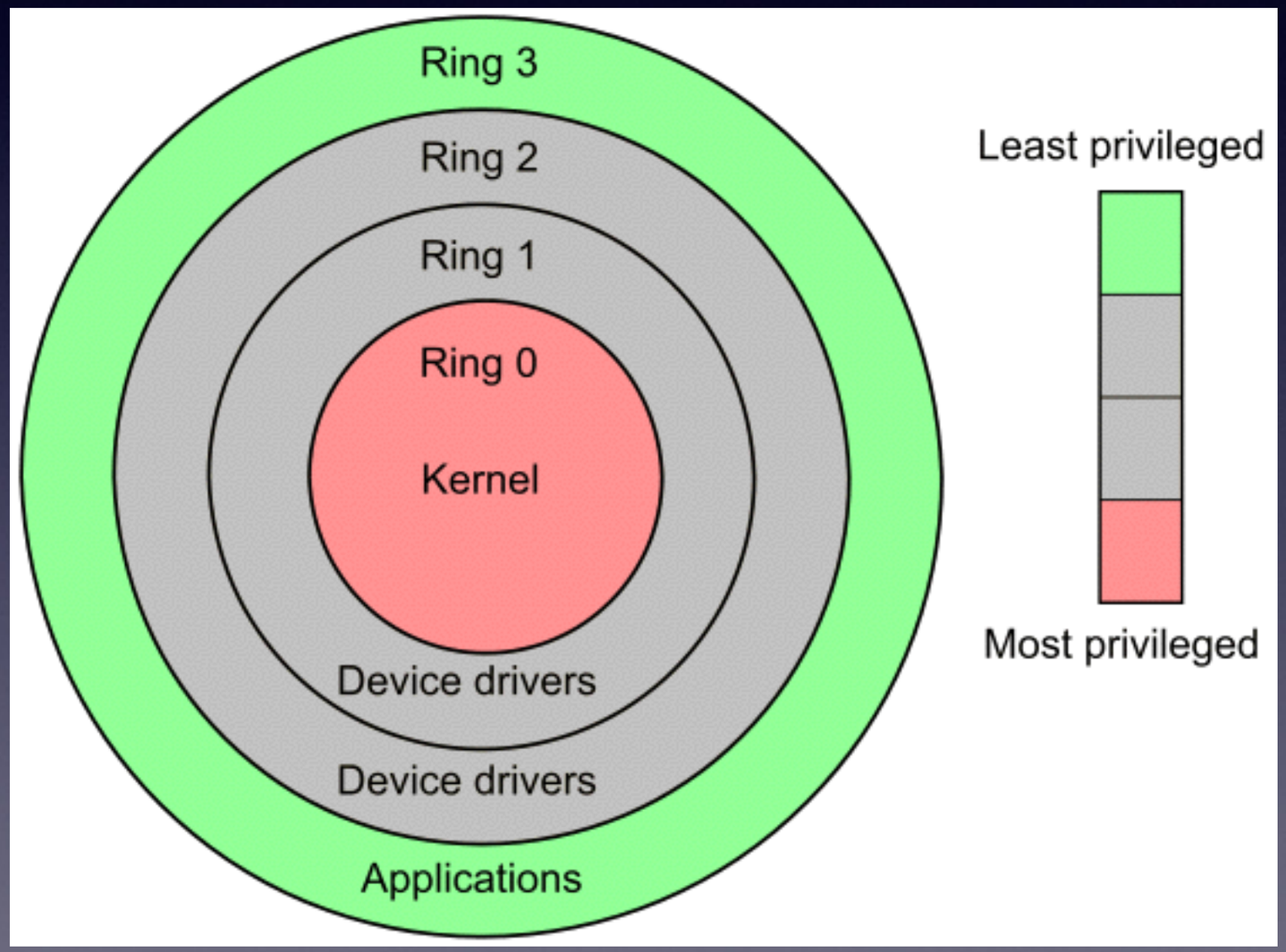
CPU Ring

为了防止app的错误代码使机器崩溃，现在大多数的cpu都划分了2种或者多种状态。某些指令只能在高级别下执行





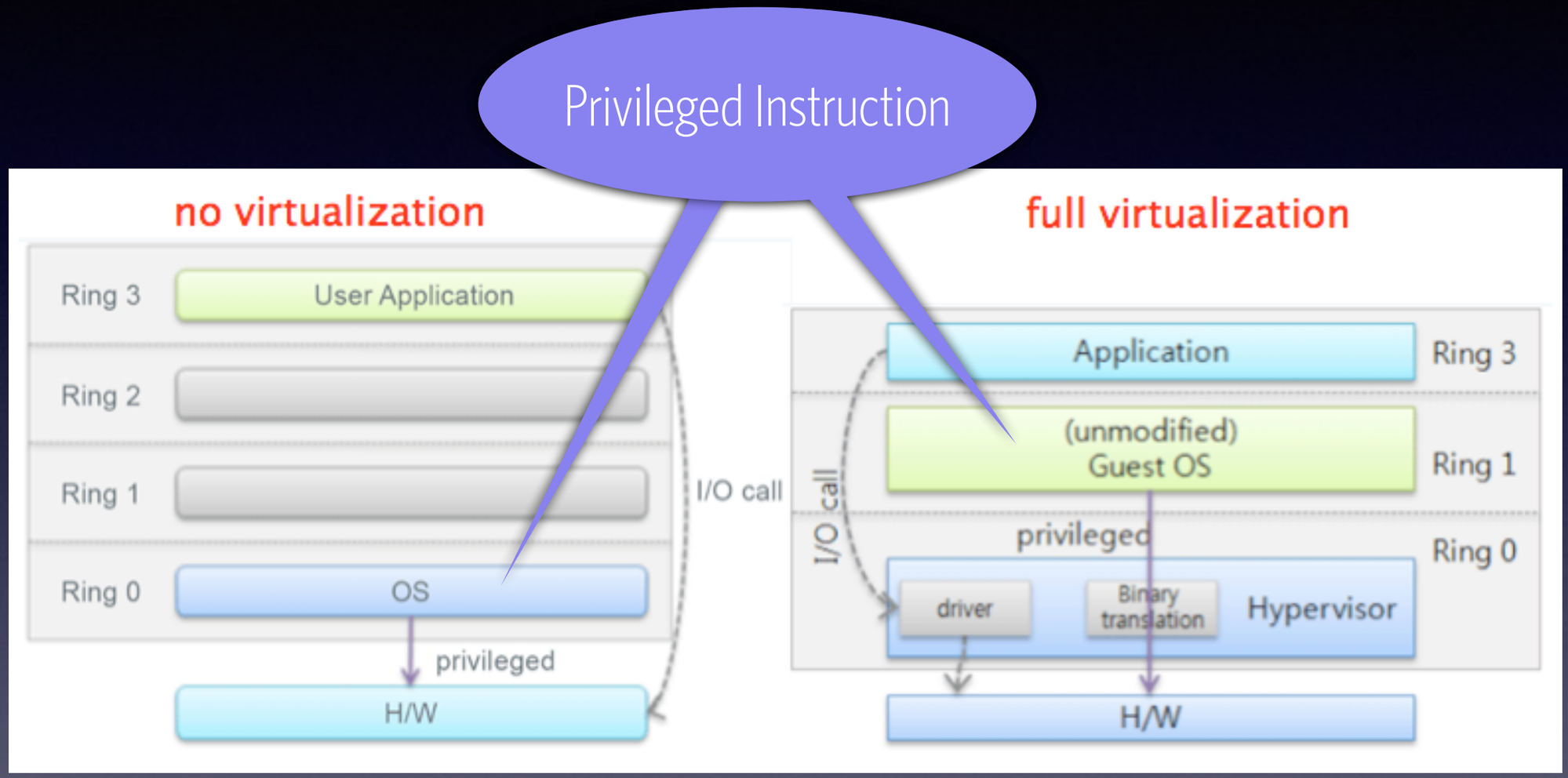
CPU Ring



Non-Privileged Instruction: can run in all rings;

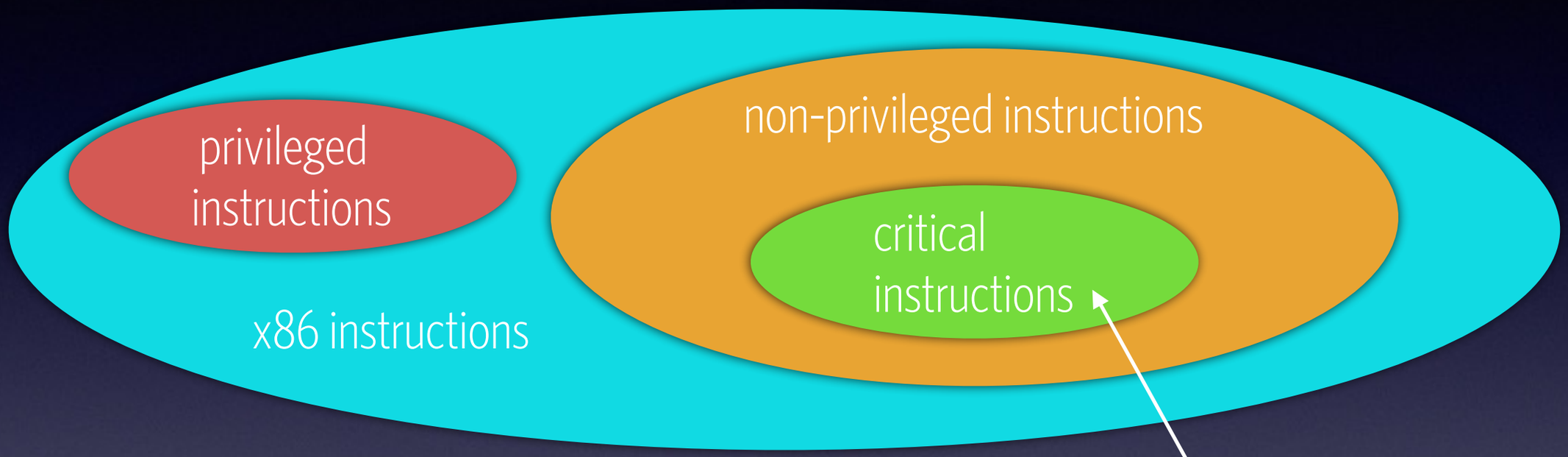
Privileged Instruction: only run in ring 0; trap in other rings;

When brings in Virtualization



When privileged instructions run, it will cause Trap and caught by hypervisor and run by hypervisor

When brings in Virtualization



only used by OS but not privileged instructions won't cause Trap and caught by hypervisor

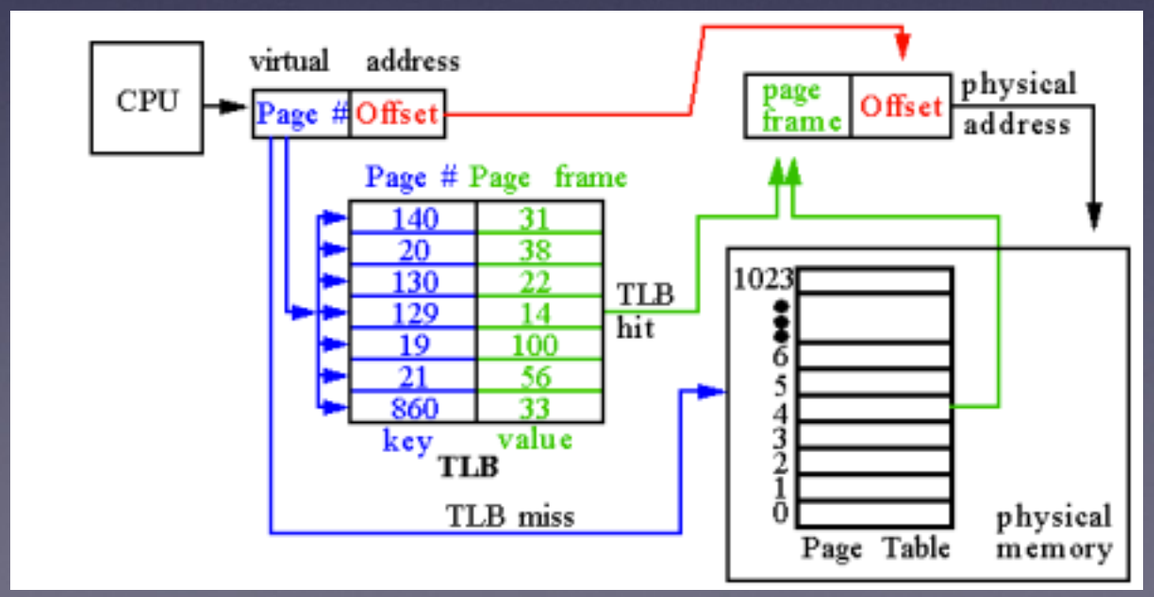
This sounds cool, but...

When privileged instructions run, it will cause Trap and caught by hypervisor and run by hypervisor

And memory in Virtualization ...

(TLB) translation lookaside buffer

Virtual address
—> Physical address
—> read/write data



Virtual address
—> 1st level page table
... ...
—> xx level page table
—> Physical address
—> read/write data

Principle of Locality
first see TLB, if miss do sth else

And memory in Virtualization ...

The TLB will be flush when context-switch happens.

The flush of TLB has a huge impact on machine performance

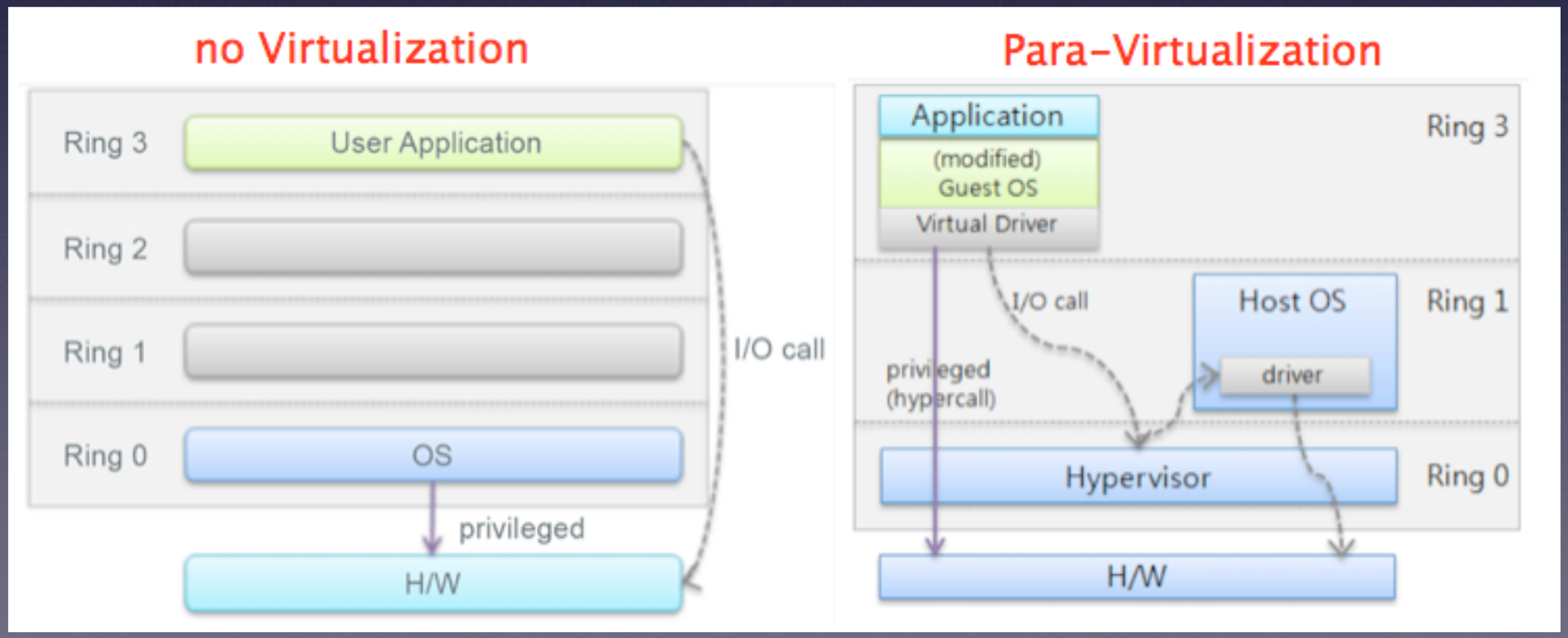
in some architectures	in x86 architectures
• software managed TLB • tagged TLB	:-(

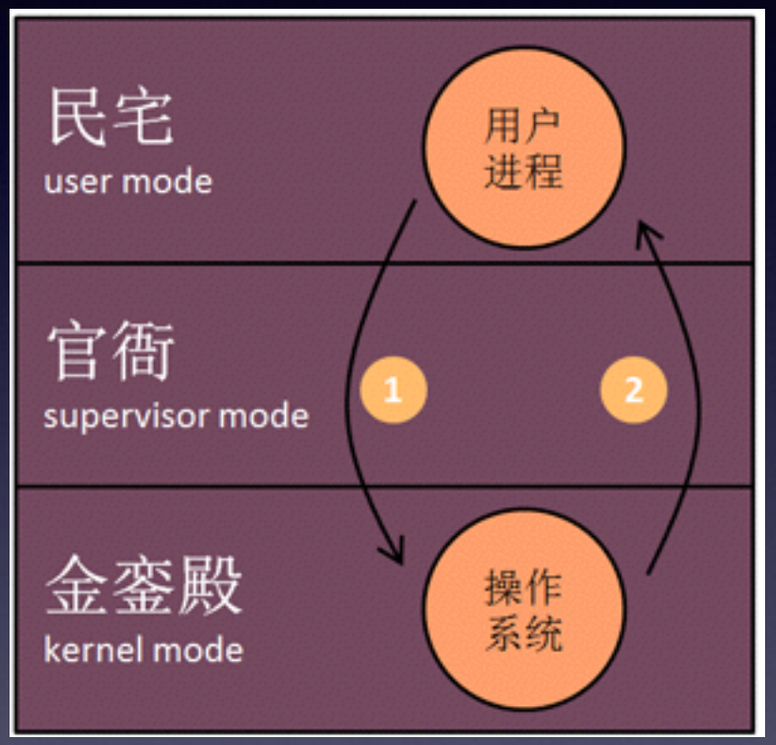
many context switches between hypervisor and guest OS in virtualization

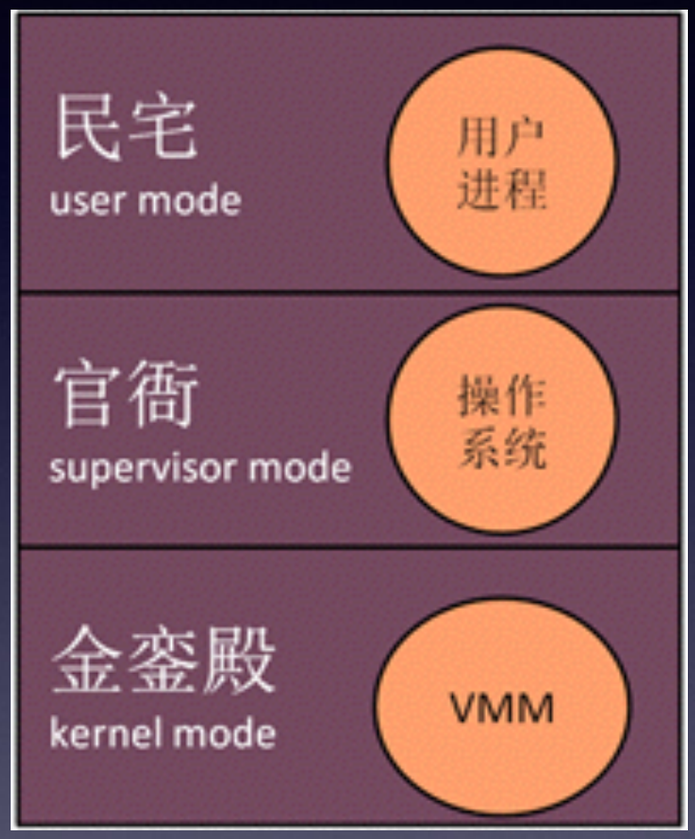
T – T

How Xen deal with these issues ?

Hypercall: modified guest OS kernel, let guest OS visit some virtual resources(like registers); guest OS kernel knows itself running in Ring 1 and hypervisor executes those critical instructions on behalf of guest OS

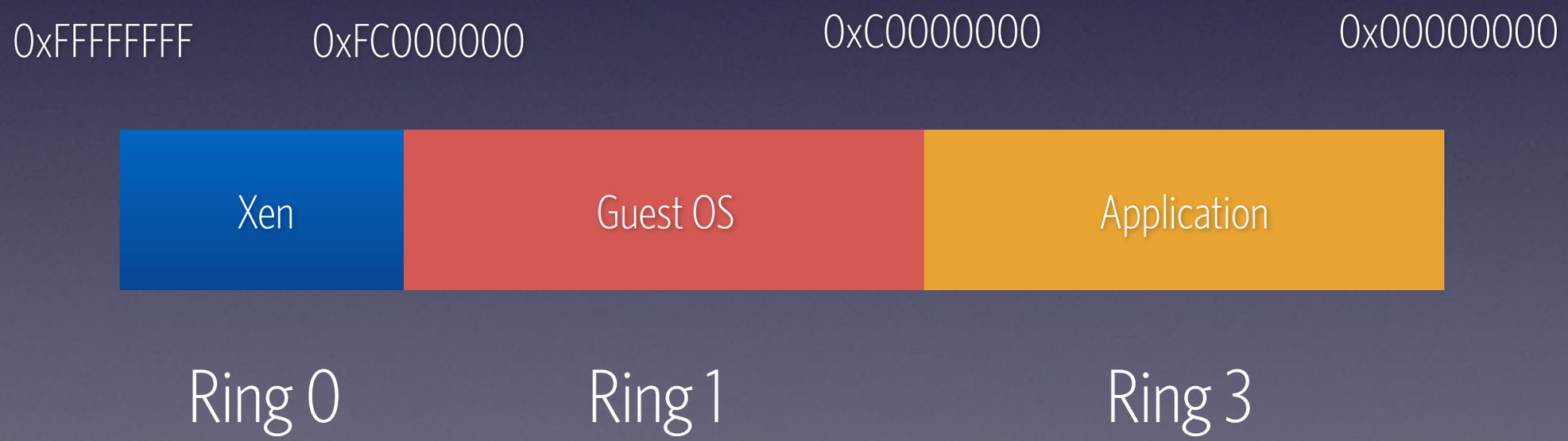



民宅
user mode
用户
进程
官衙
supervisor mode
1
2
金銮殿
kernel mode
操作
系统


民宅
user mode
用户
进程
官衙
supervisor mode
操作
系统
金銮殿
kernel mode
VMM

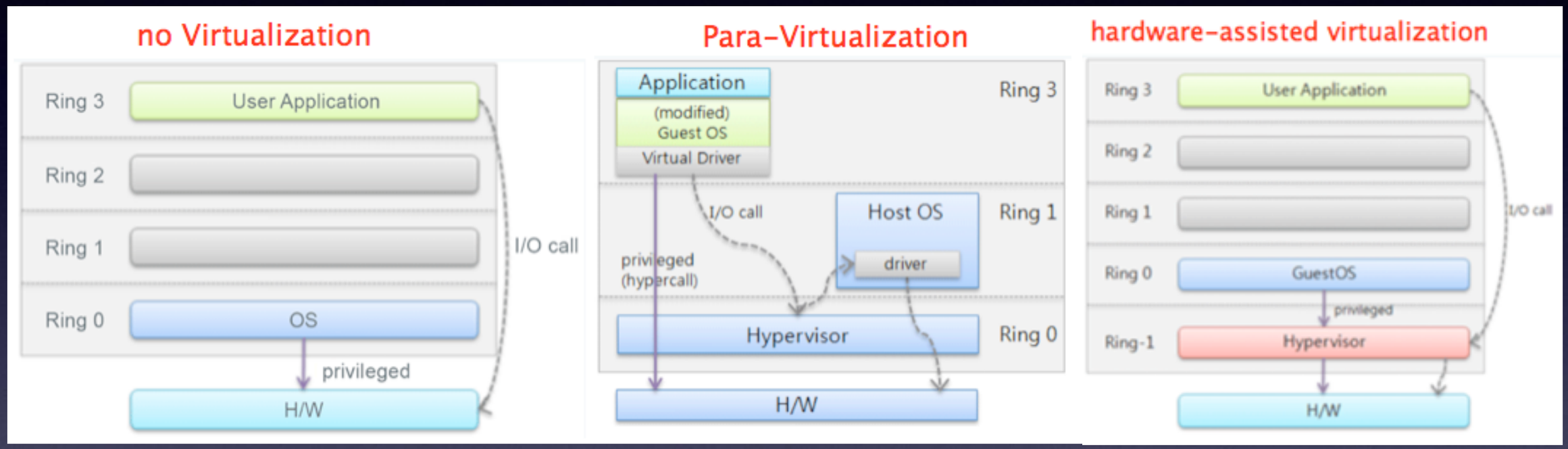
How Xen deal with these issues ?

Merge Xen and Guest OS address space. Therefore, context switch does not need when Xen executes hypercall. Also, Xen can access guest OS memory page without memory mapping. In addition, TLB does not flush too often and performance can be guaranteed.



How Xen virtualized resource ?

How Xen virtualized CPU ?

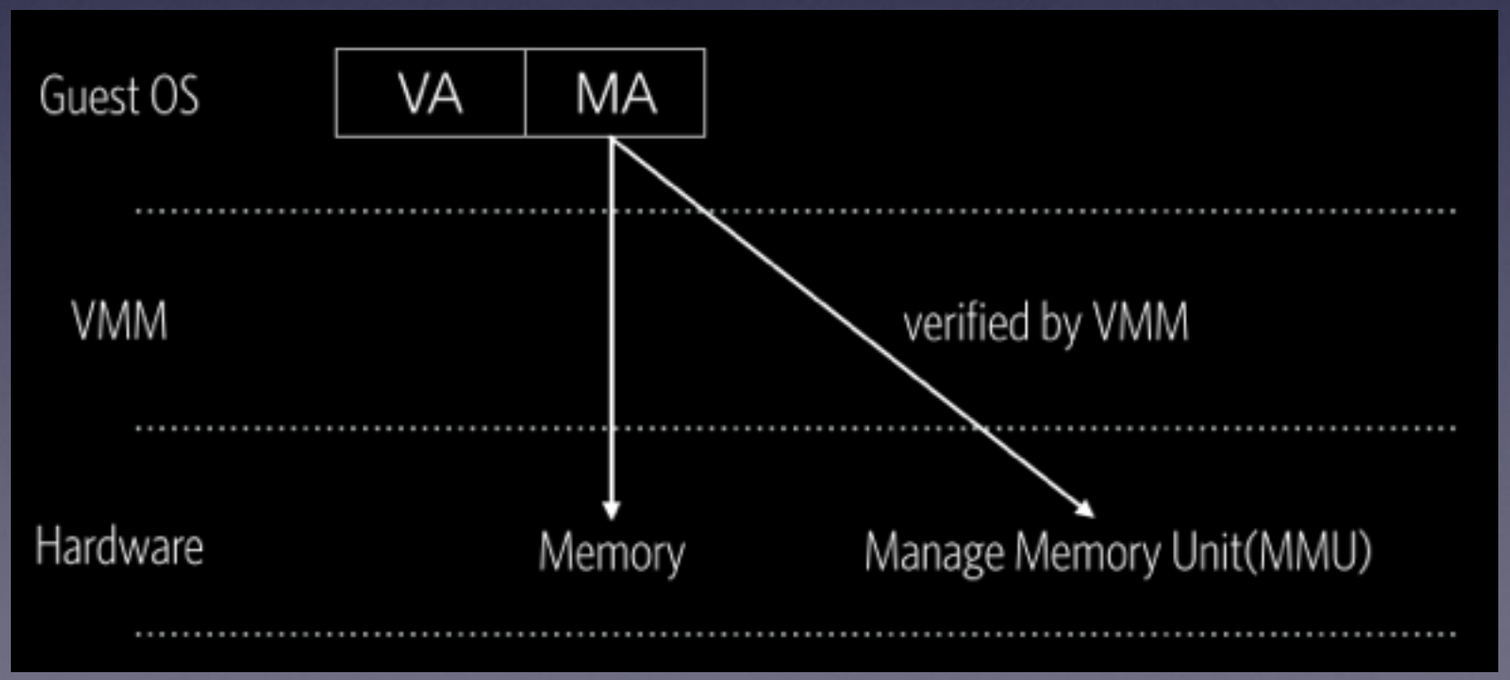


- Para-virtualization: hypervisor needs assists from OS to simulate x86 privilege instructions; OS needs to be modified; also called OS-assisted virtualization

- Hardware-assisted virtualization: hypervisor needs assists from hardware to simulate hardware resources; e.g.: Intel-VT, AMD-V

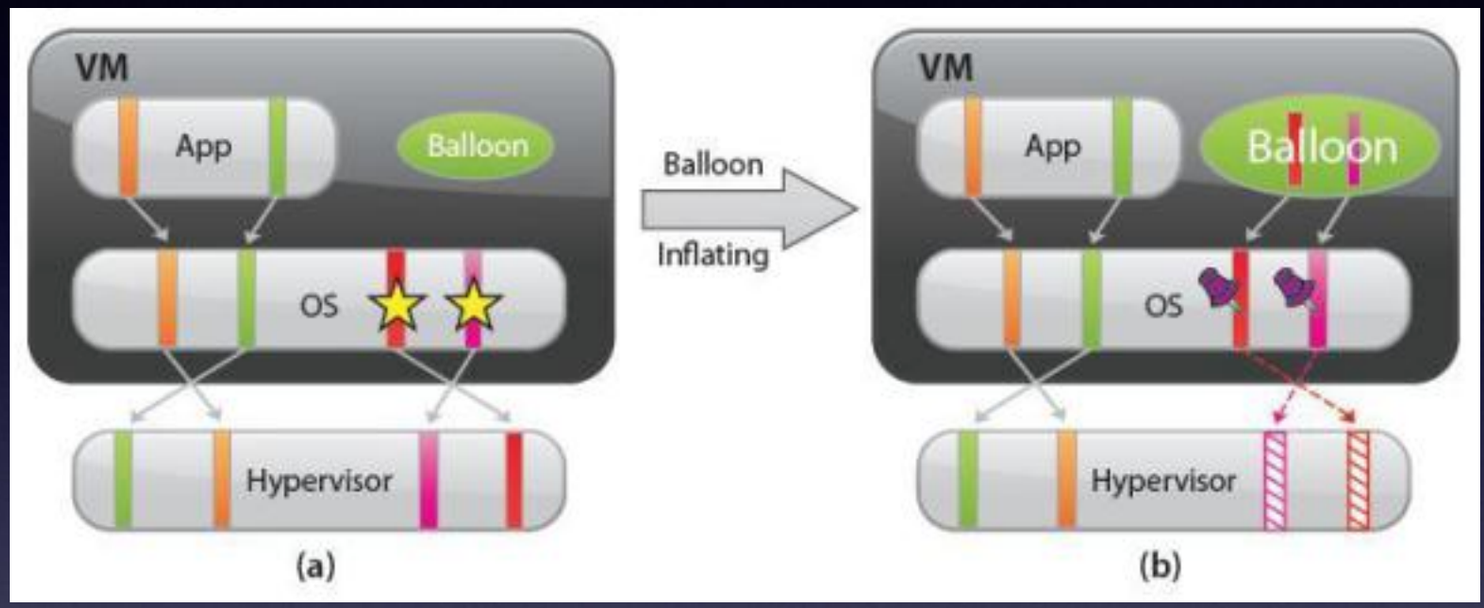
How Xen virtualized memory ?

- Memory Segmentation: make Xen and guest OS share one memory address space and simplify memory management
- Memory Paging: Xen can isolate different domain memory and make each domain access its own page table



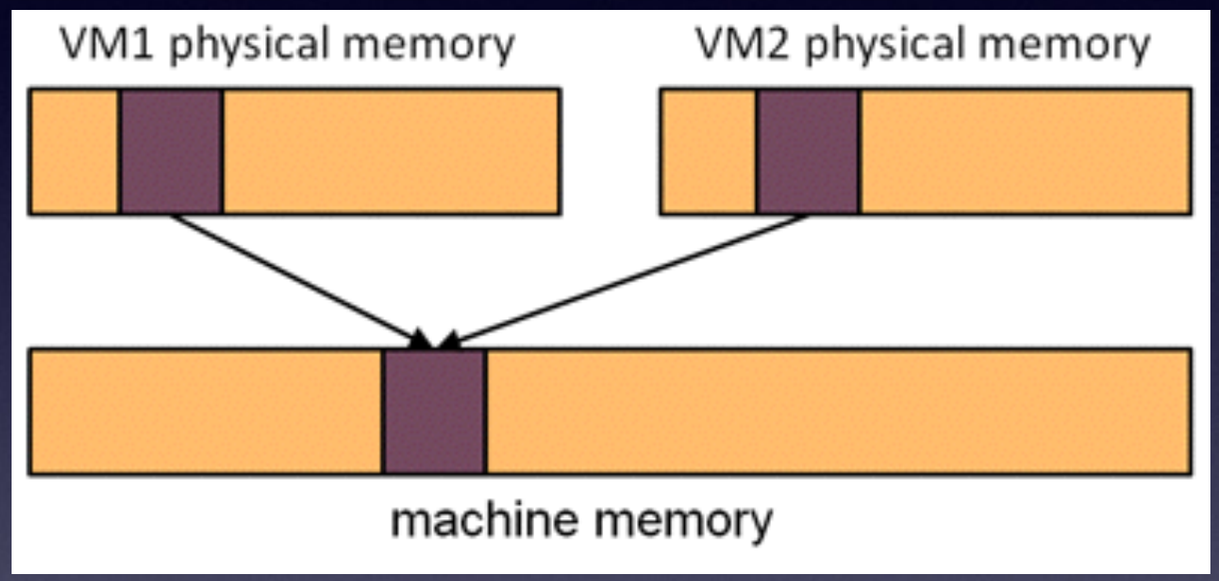
- MMU Para virtualization

How Xen virtualized memory ?



- Balloon driver: allocate and release domain memory
- Balloon inflate: Xen release domain memory from VM
- Balloon shrink: Xen allocate domain memory for VM

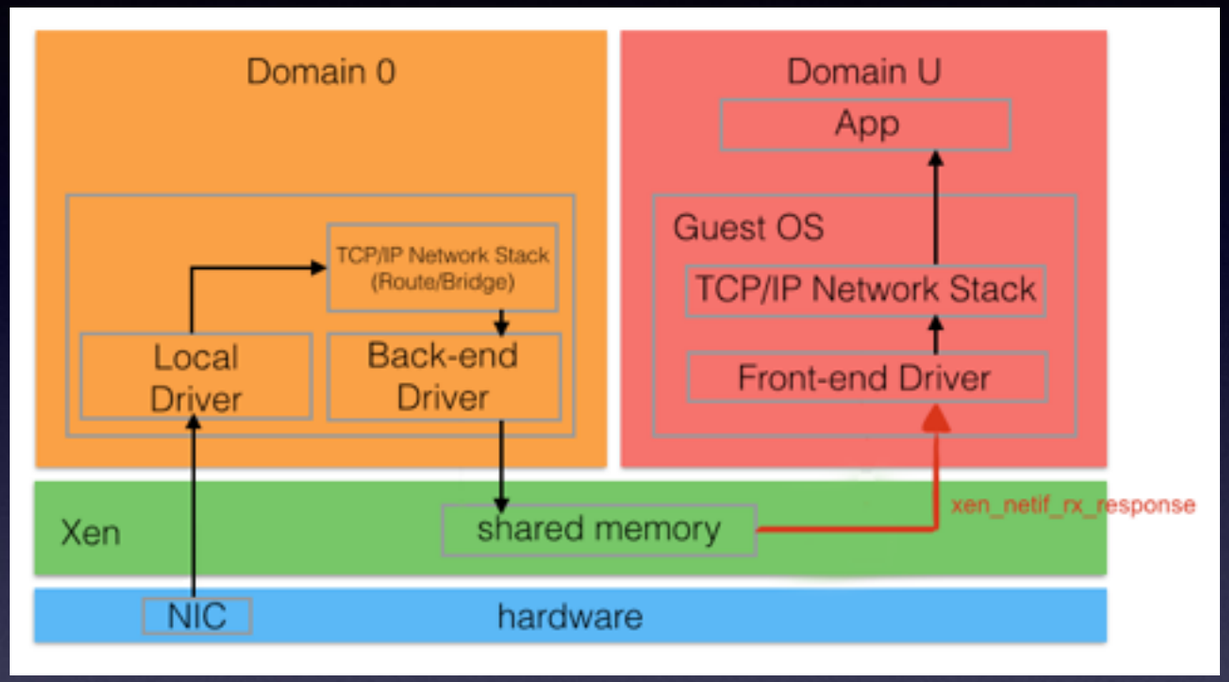
How Xen virtualized memory ?



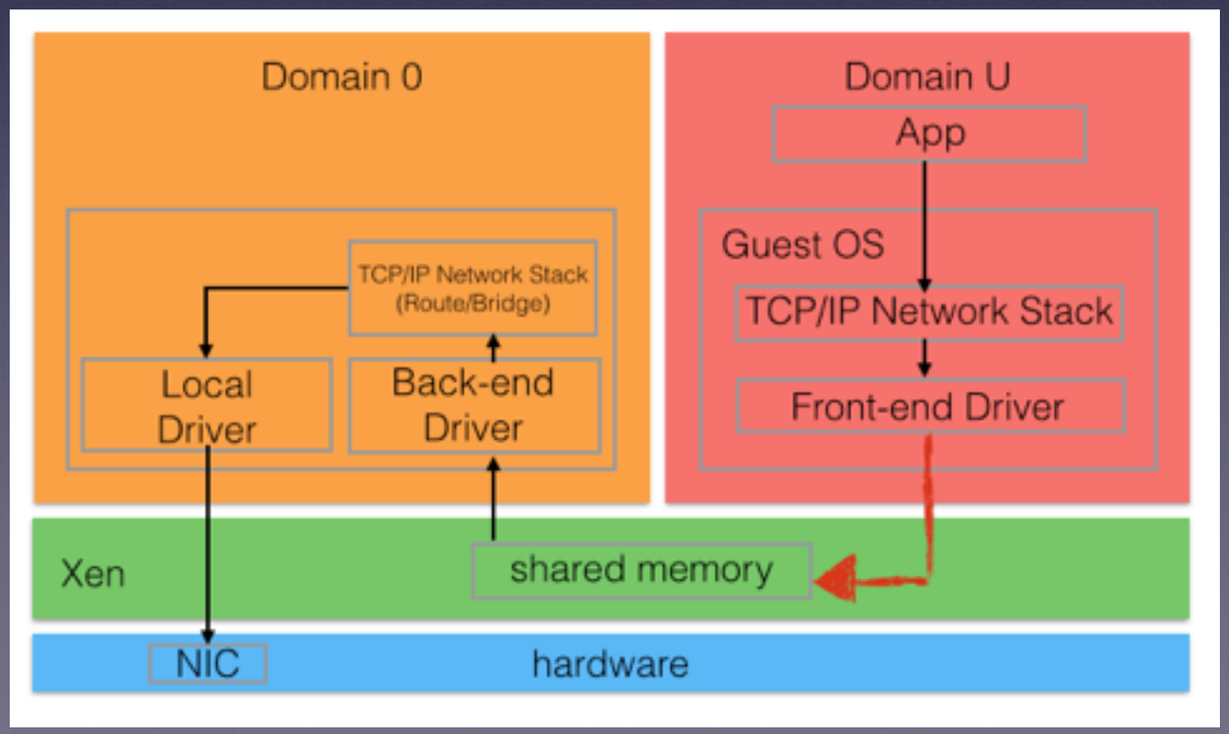
- How 1GB memory used for 2GB ?

How Xen virtualized I/O ?

- Receive packet



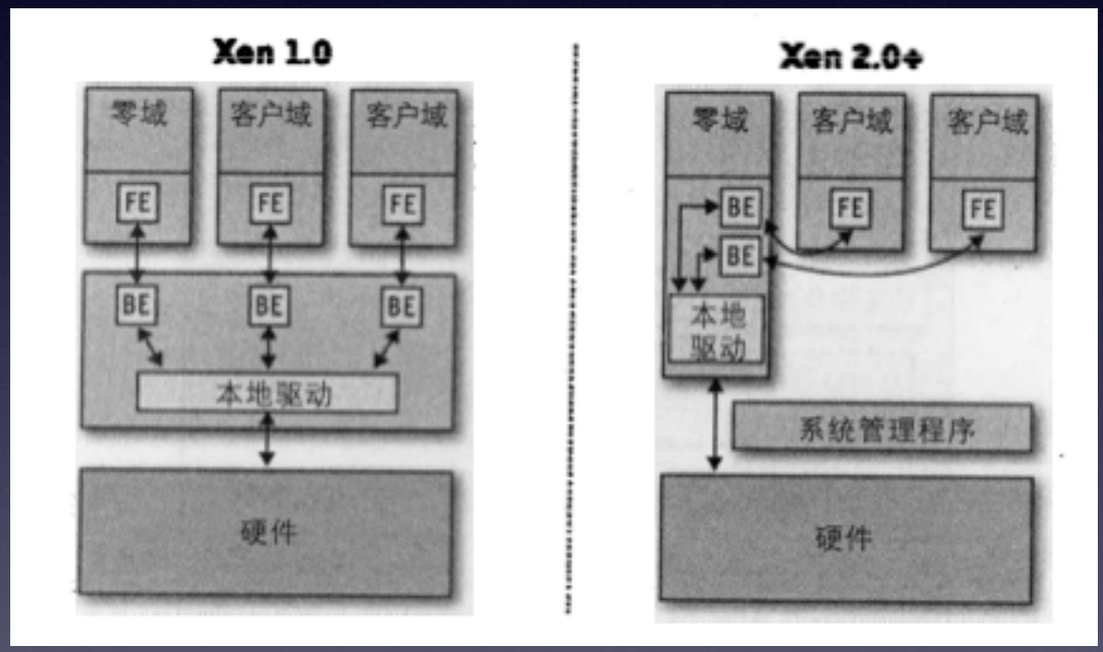
- Send packet



How Xen virtuali

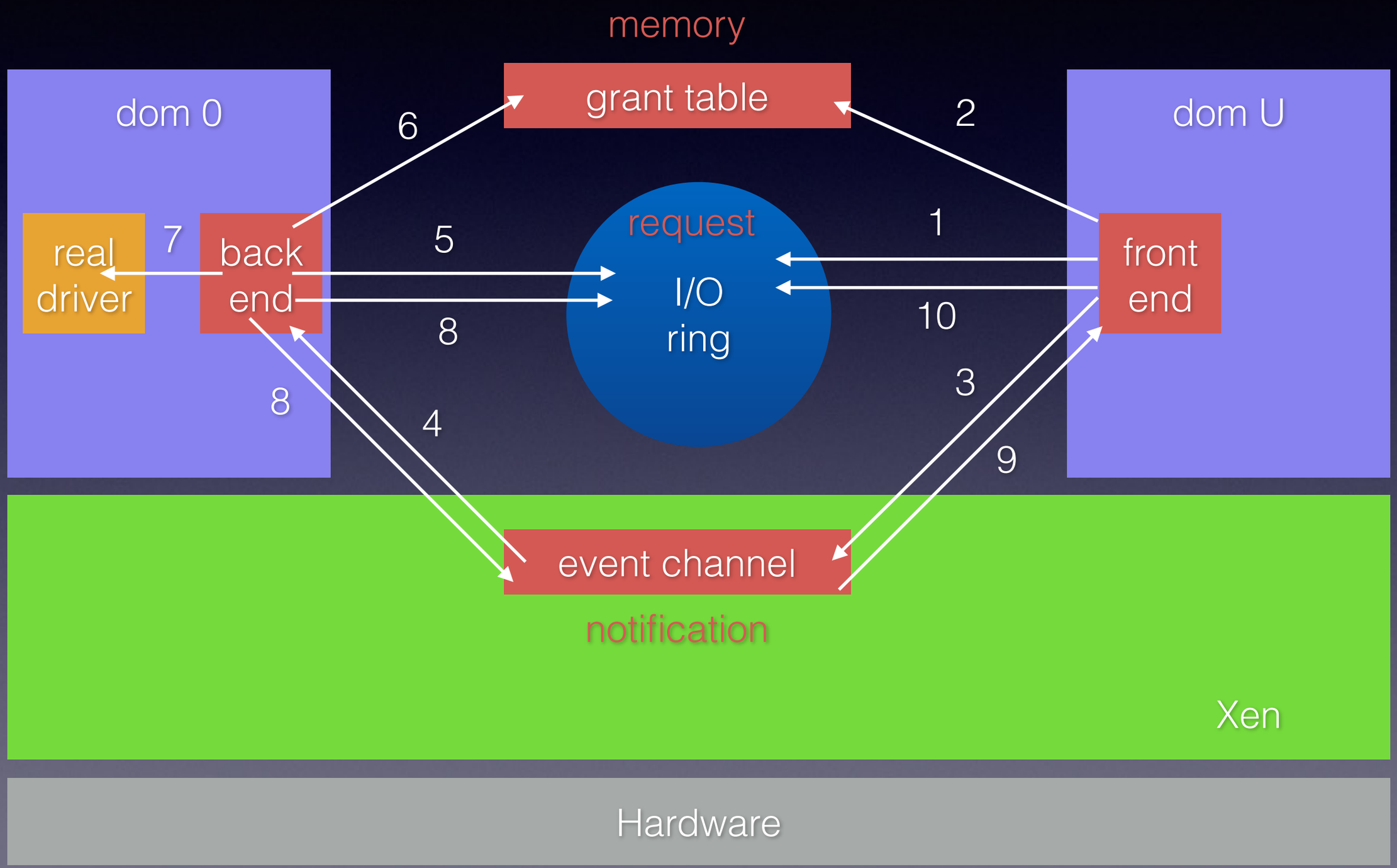
《架构之美：第七

In the following picture, you can see the difference between Xen 1.0 and 2.0. In Xen 1.0, the virtual backend is realized in hypervisor. All domains need to visit the hardware by backends. In Xen 2.0, the architecture is simplified. Domain 0 can access hardware by local drivers and all backends of Domain U are moved into Domain 0. Since Xen 2.0, frontends and backends communicate with each other through event channels which make the communication between domains much efficient and safer.

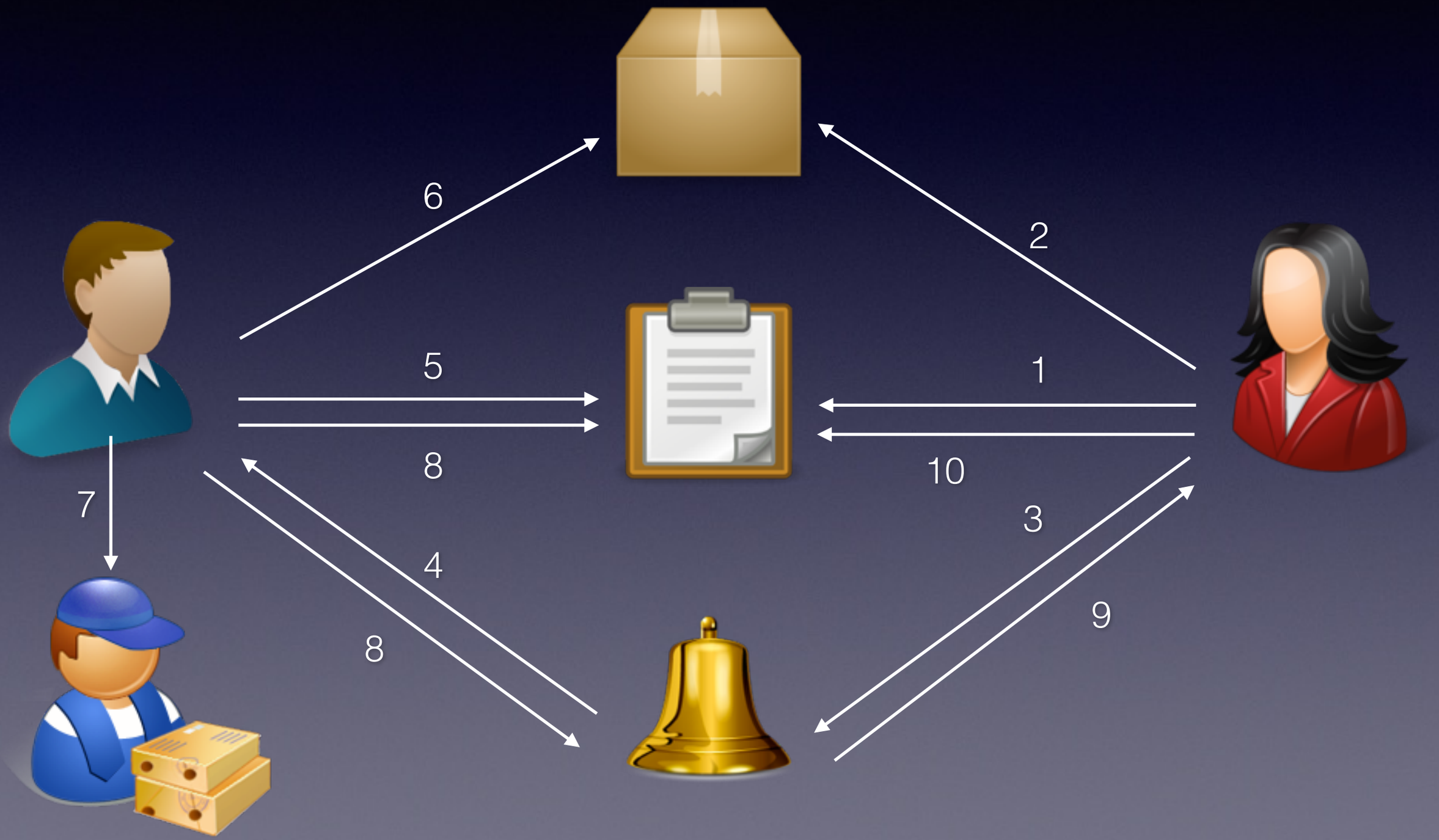


In Xen 1.0, all VMs are using virtual device to access hardware and hypervisor is responsible for transferring all these visits to the real hardware. Therefore, hypervisor will include all frontend Linux hardware drivers and backend virtual drivers. It simplify the virtual machine design, but increase the complexity and kick the responsibility of developing new drivers to Xen develop.

How Xen virtualized I/O ?



How Xen virtualized I/O ?

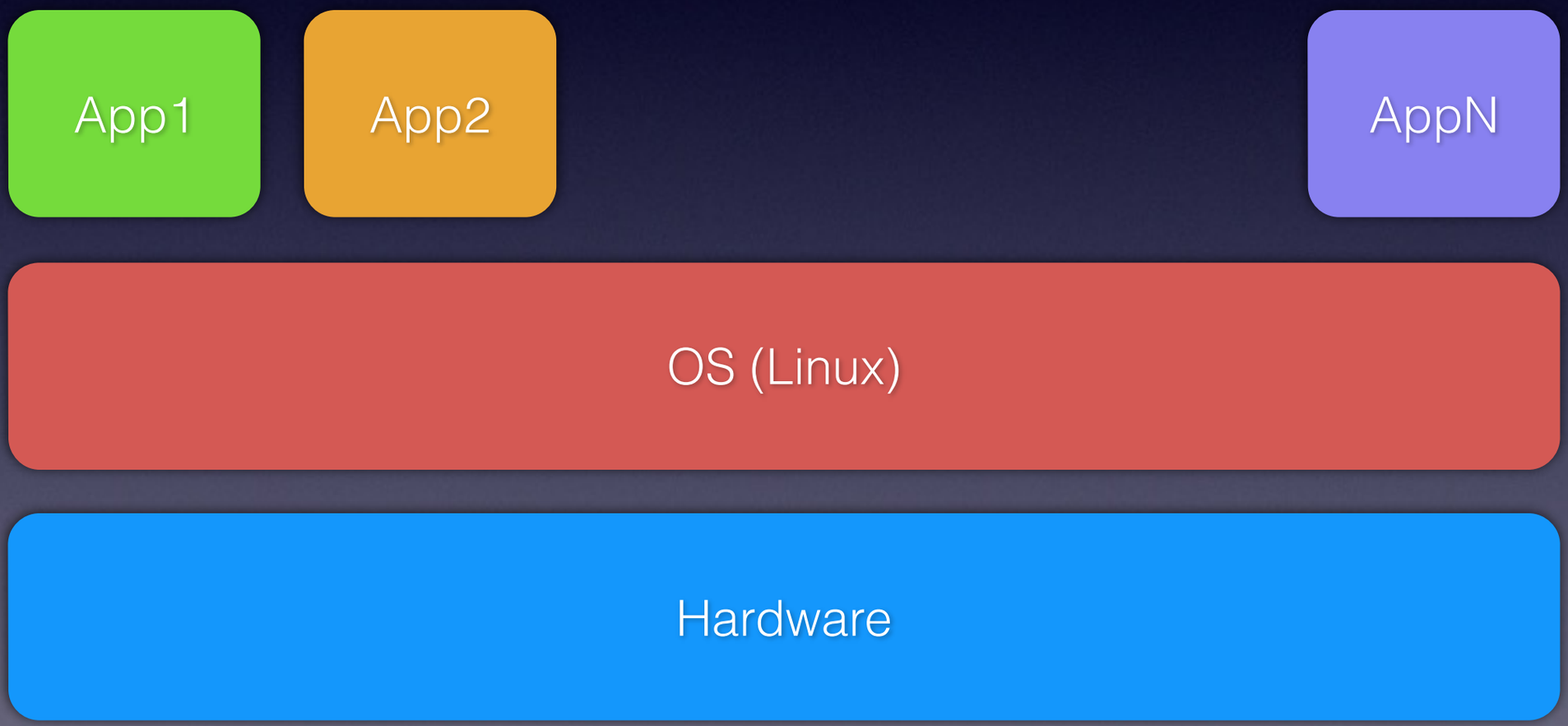


Topic Today

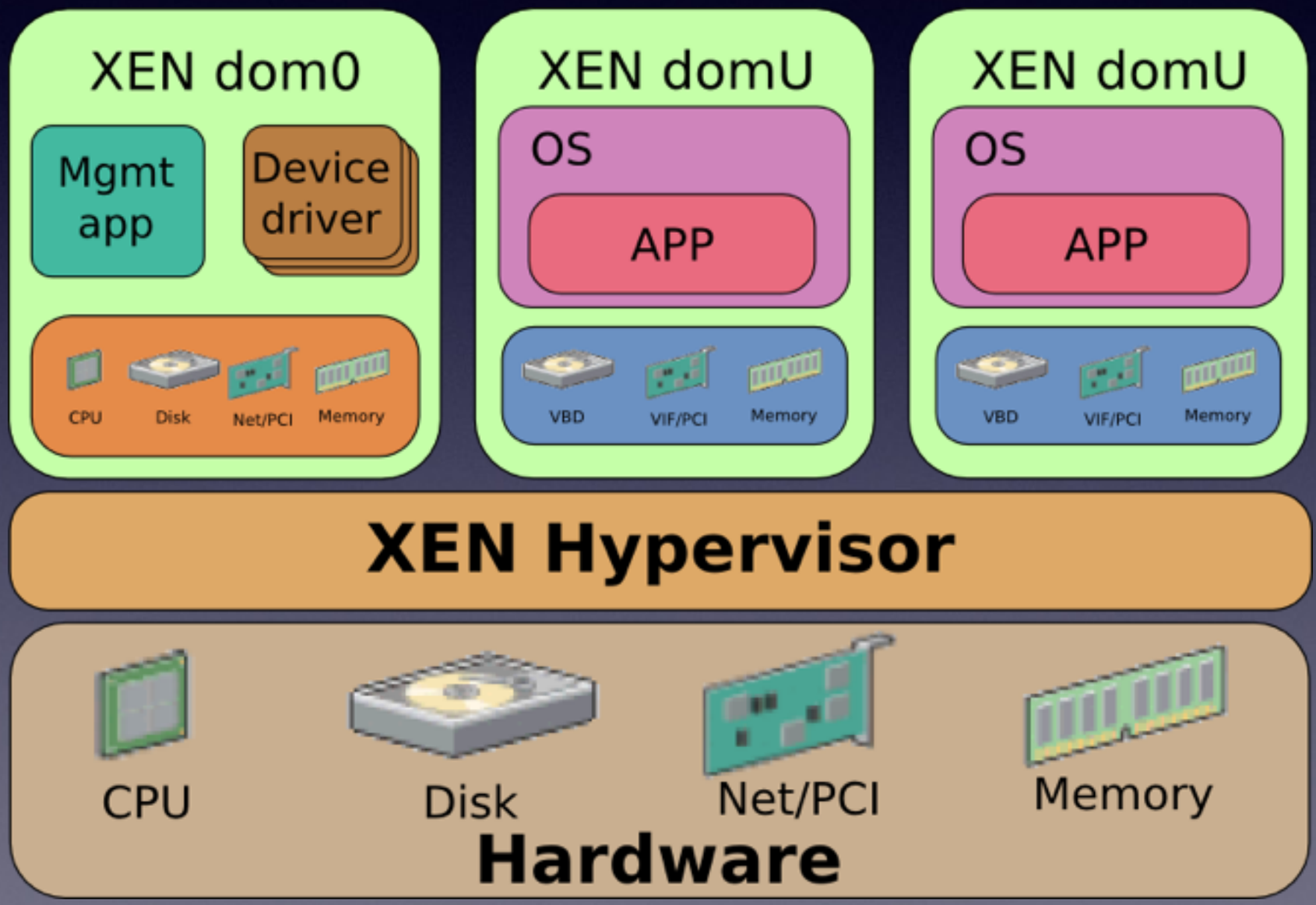
- What is Xen ?
- Xen history
- Xen design philosophy
- How Xen virtualize resources ?

Demo: how to install Xen

Pure Linux



After installed with Xen



Thank you