Virtualization

GIRS 2015

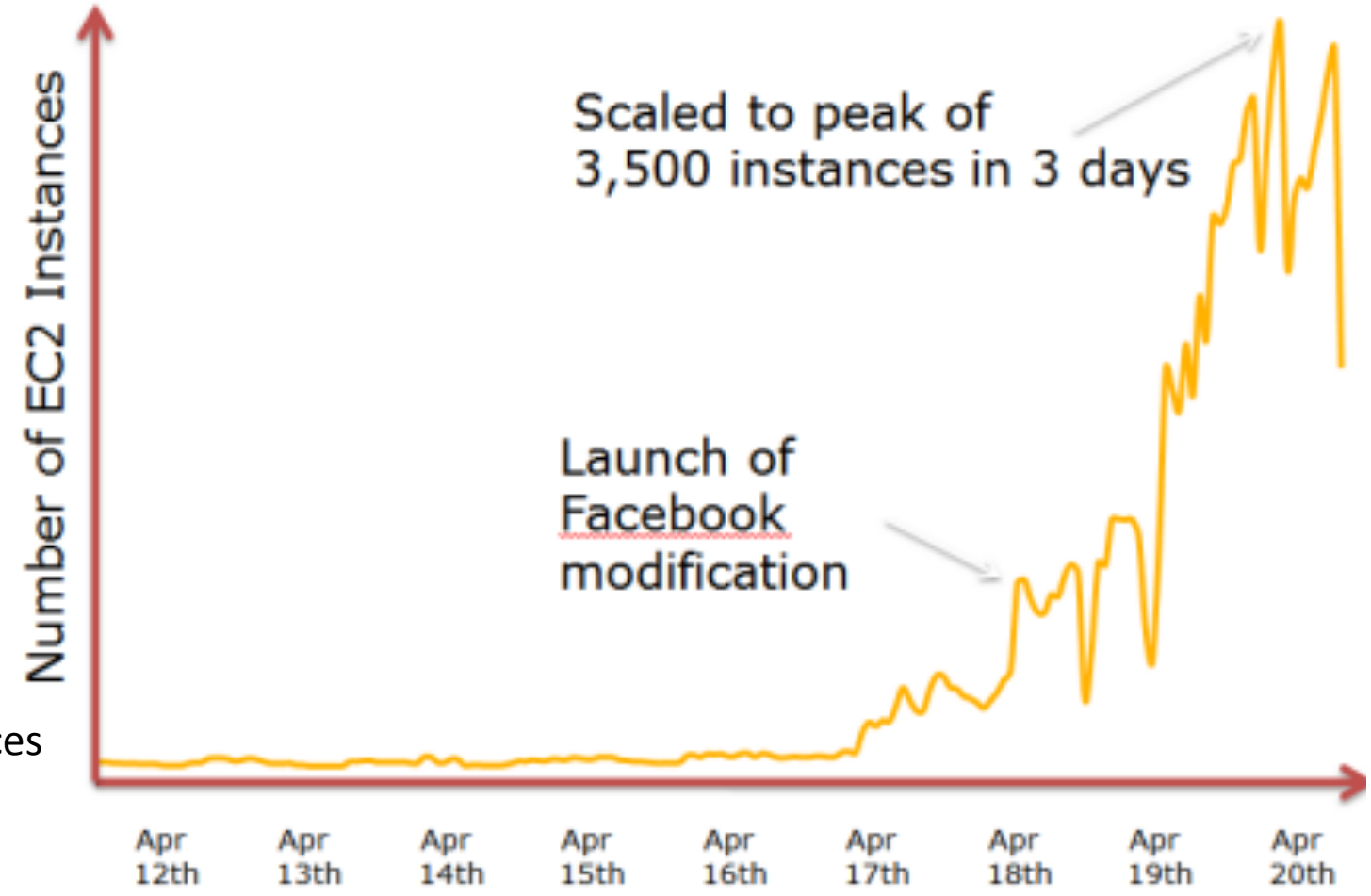
Information Services Infrastructure Some numbers (USA)

- 38 million physical servers
  - +700% growth in next 15 years
- $140b unused capacity
- 30%-50% server cost is related to power
- Average costs for a datacenter
  - $5K-$15K / sq meter
  - $2400 / server
  - $40.000 / rack
- 20-30 : 1 – Server / Administrator ratio

Information Services Infrastructure

- Datacenters are not green!
  - 1 rack full of 1U blades = 20-25kW = peak consumption for 30 houses
  - 11% of power is lost
- Provisioning for peak time results in waste
  - Peak should reach 80% capacity at most
- What about flash peaks?

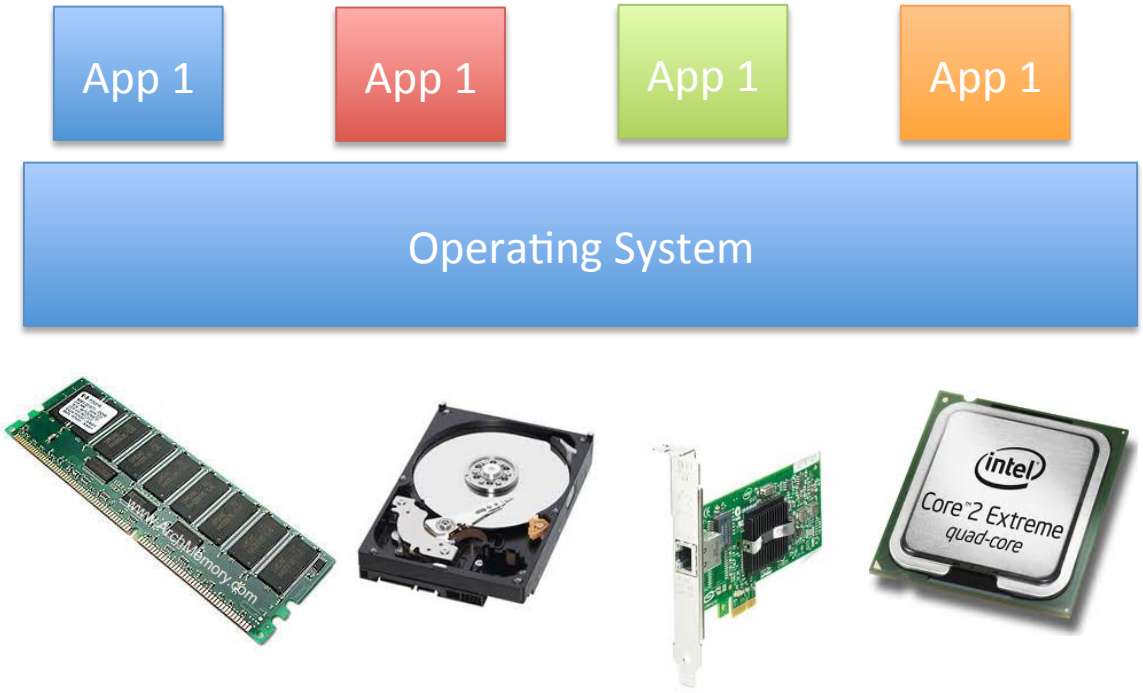
Flash Peaks



~40 instances

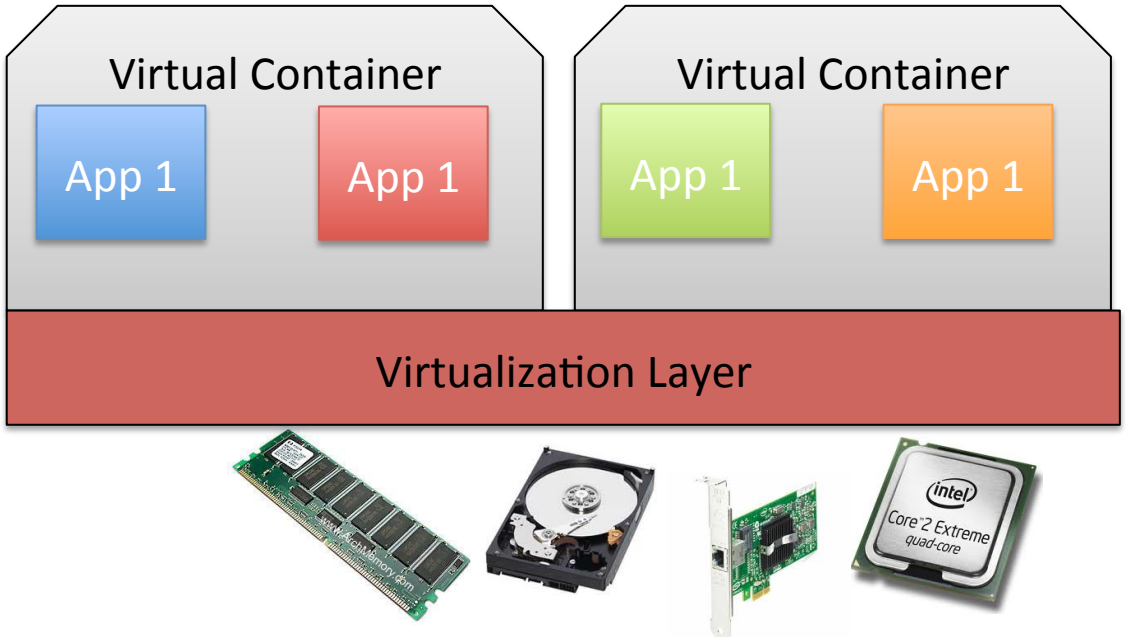
Virtualization

- Traditional model
  - One single OS controls all applications and devices



Virtualization

- Virtualized System
  - Multiple containers over the same hardware
  - Extra level of indirection
  - Multiplexes guest accesses to real hardware
    - May provide virtual hardware

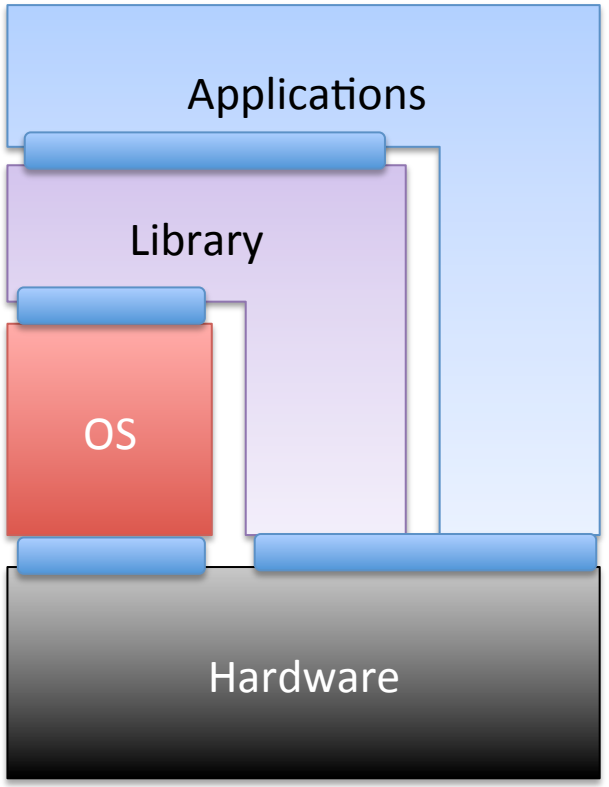


Virtualization

- Like JAVA?
  - NO! Java creates an homogeneous virtual environment to run a single application
- Like ZSNES/MAME/Project64?
  - NO! They emulate the behavior of specific, non general purpose systems
- Like COTSON?
  - NO! Simulators aim to evaluate execution one instruction at a time. Very far from real time (~50-100x perf penalty)

Virtualization

- Virtualization can occur at different levels
- Library calls
  - Same OS/HW, individual libs
- System calls
  - Same HW, individual OS view
- Hardware calls
  - Individual hardware



Virtualization

- Very abstract concept...
  - Virtual Memory, Virtual File System, Virtual Networks (VPN's)

- Virtualization allows a single computer play the role of several computers, through sharing hardware resources to multiple environments.

- Virtualization is a tecnique allowing to run several different systems in the same hardware, in a totaly separated manner.

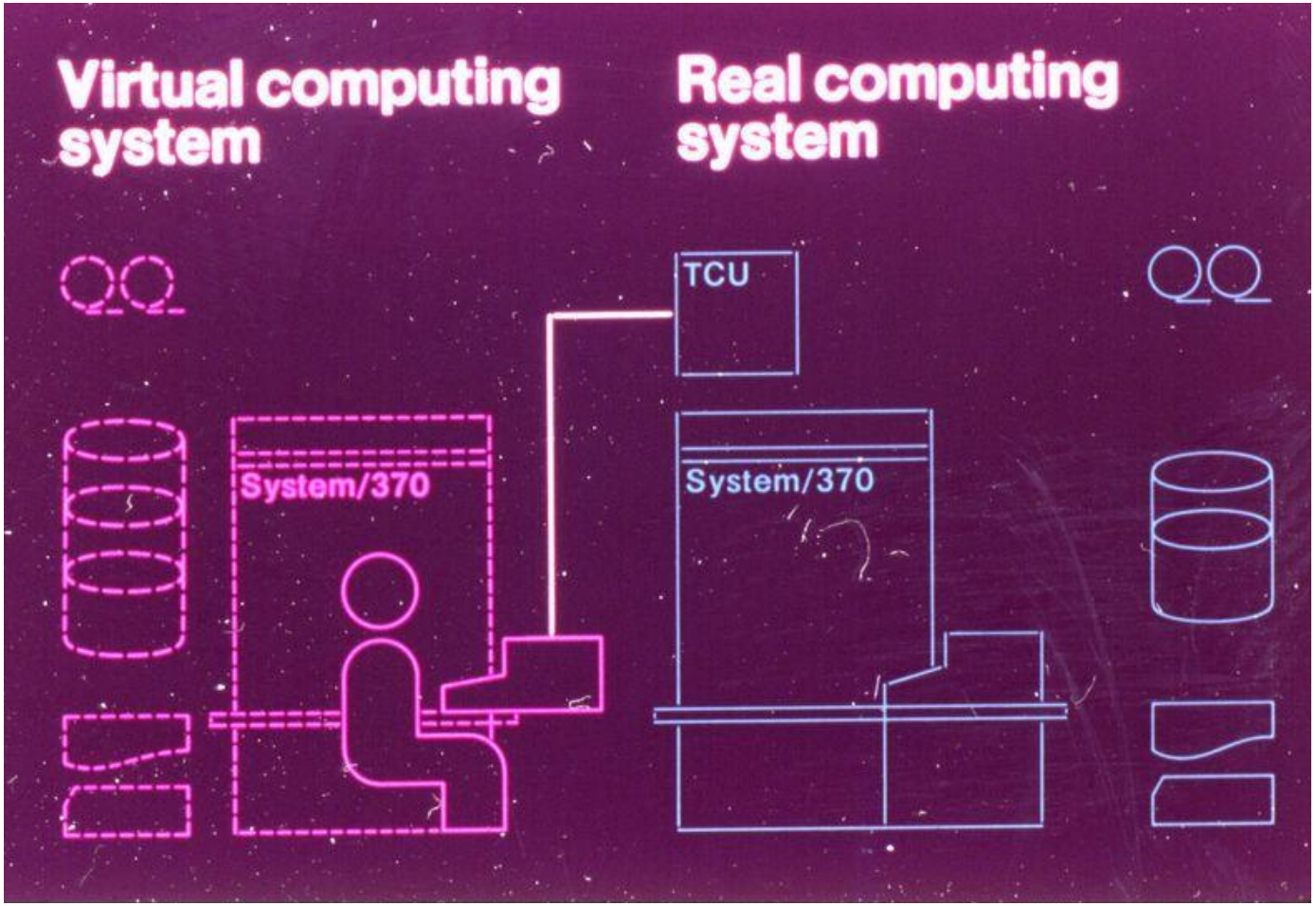
Virtualization

- Most cited definition:
  - "*Formal Requirements for Virtualizable Third Generation Architectures*" **Gerald J. and Robert P. Goldberg**, Communications of the ACM, Volume 17, Issue 7, Julho 1974
- "VMM satisfies efficiency, resource control, equivalence"
- "Theorem – For any conventional third generation machine, a VMM may be constructed if the set of sensitive instructions for that computer is a subset of the set of privileged instructions."

Virtualization history

- Is it a new technology? No!
- IBM S/360 mainframes allows application timesharing through resource partitioning.
  - Circa 1960s, concepts still present in z/VM (2000-2009)
  - Multiple single-user OS instances
  - Guest directly on hardware
- IBM also provided Logical resource Partitioning (LPAR)
  - Creates subsets of the host hardware
  - Multiple Operating systems
  - DLPAR allows dynamic modification of partitions
  - Circa 1970

IBM S/360, ca 1972



Virtualization history

- Bochs software provides emulated environments in 1990s
  - Mostly for debugging
  - Complete environment with emulated hardware
- VMWare is founded in 1998 from ideas existing at Stanford
  - Part of EMC
  - Provided Hypervisors creating fully isolated containers
- XEN sprouts from Cambridge in 2003
  - Proposes para-virtualization (host assisted)
- Intel VT-x enhances the support virtualization acceleration
  - Circa 2005

Virtualization history

Every company jumps to virtualization

- QEMU in 2005 – allows multiple architectures
- Parallels Workstation in 2006
- Linux KVM in 2008
- Oracle Virtualbox in 2007
- Microsoft Hyper-V in 2008
- Microsoft Virtual PC in 2009

Virtualization History

Virtualization enables the flexible cloud model we have

- Datacenters are now completelly virtual
- Services in Virtual Instances in bare metal

Virtualization History

Many software packages for Virtualized environments

- VMWare ESX, ca 2002
- OpenStack, 2010 by NASA
- OpenNebula, 2008
- Apache Cloud Stack, 2010
- ...

Virtualization History

Companies adopted virtualization into their development/operation workflows

- VMs for instant deploy and validation
- Development Environments for each programmer
- Migration from Dev to Prod

Common uses

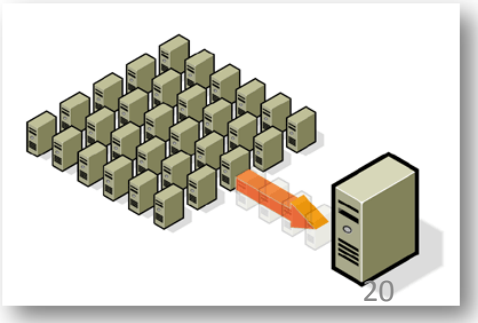
- Test and Development
  - Rapidly provision servers and create snapshots
- Server Consolidation
  - Eliminate server number by moving several systems to same metal

Common uses

- Business Continuity
  - Reduce cost and complexity by allowing easy replication and restore
- Enterprise Desktop
  - Secure, unmanaged PCs with full administration permissions
- Horizontal Scalability
  - Enable more virtual workers to scale applications

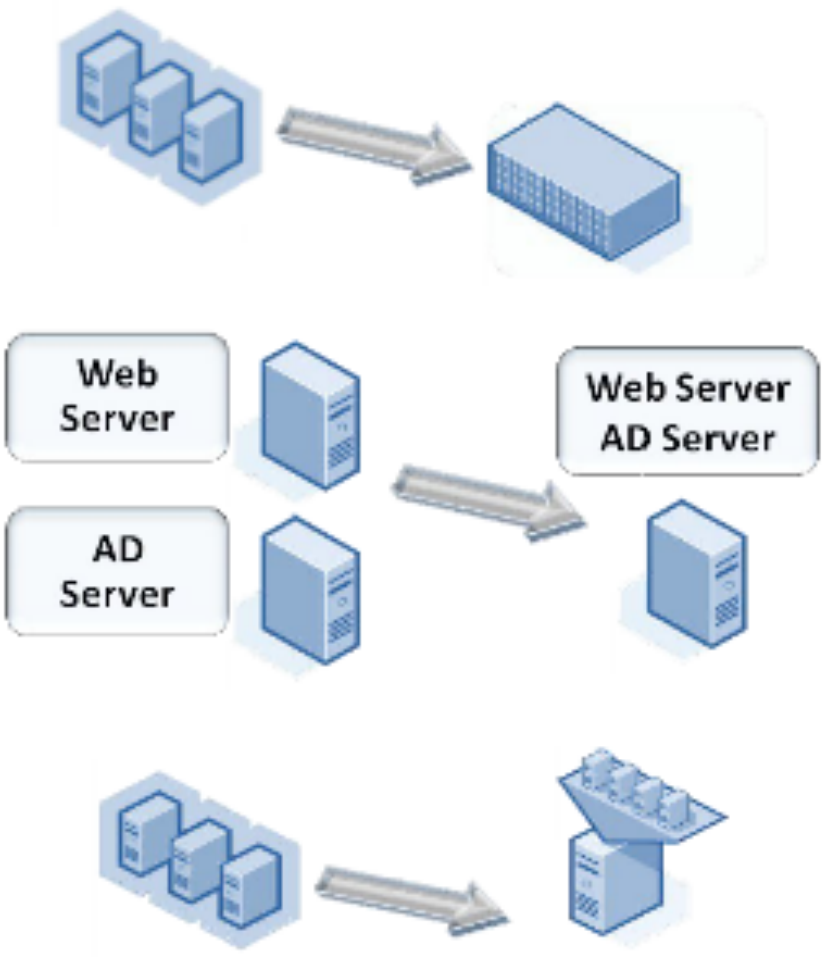
Why - Consolidation

- Paradigm 1 App -> 1 OS-> 1 Server
  - Low usage rates (<25%)
  - Either dimension for peak or off-peak time
- High infrastructure cost
  - More HVAC, more events, more raised floor.
- Management cost
    - Configure each OS, deploy each OS, update
    - Update, replace parts on each server



Resource Consolidation

- Physical Server consolidation
  - Example: rack's, blades
  - Save space, optimizes resources in cooling
  - Reduces management points
- Application consolidation
  - Saves resources (unused capacity)
  - Reduces management points
  - Conflicts! Ports, Libraries, OS
- Infrastructure Virtualization
  - Provides isolation
  - Conflicts easily solved



Why - Security

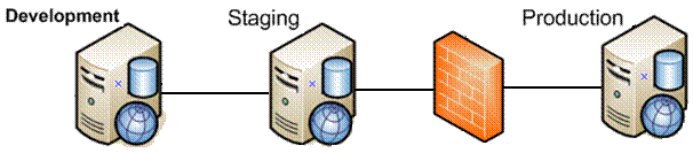
- Isolate applications
  - Each application runs in its closed environment
  - Isolate actual operating system
    - Ex. Playstation 3, Xbox 360
- Define quotas for guests
    - Limit CPU, RAM, IO priority

Why - Encapsulation

- Entire VM is a file
  - Including memory and device state
- Easy backups, auditing and restoring
  - If compromised, just copy the image and boot again
- Easy content distribution
  - Pre built virtual appliances
  - Demos
  - Create once, run anywhere

Why - Isolation

- Same hardware for development/production/ staging/sales
- Homogeneous environment
  - Applications will all see same hardware
    - Independent of market trends
  - Easier to replicate issues
    - Just replicate the guest and repeat the triggering process
- Possible to define priorities and quotas for different costumers



Why - Independence

- Applications must only take care of virtual hardware
  - Real hardware can be freely replaced
    - Legacy applications do not halt upgrade plan
  - "Hardware vendor independence"
- "Green"
  - Less Servers = Less Power Requirements
  - Less Power = Less HVAC cost
  - Less Servers = Lesser environment impact

Why NOT!

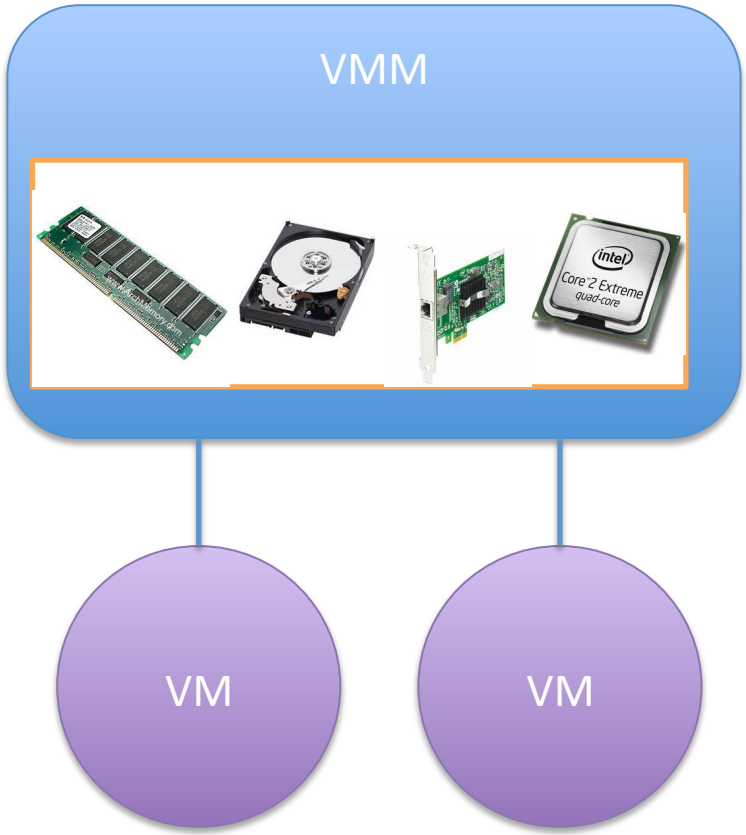
- "High Availability"
  - If hardware resources are shared, they may not be always available.
  - Virtualization is best for heterogeneous environments
- "Vendor Lock In"
  - Hardware vendor lock in replaced by Virtualization solution and management vendor lock in.
  - Each solution has its own quirks -> will impact processes
- "High performance I/O"
  - Input/Output latency and jitter are easily effected by CPU usage

Why NOT!

- Some applications require specific hardware
  - Graphic Processing Units
  - Security Dongles
  - Multimedia encoders
- Decreased peak performance
  - Each abstraction layer will degrade performance by some amount
    - At least 5-10% depending on workload
  - 10% efficiency can represent milions of euros

VMM (Hypervisor)

- Virtual Machine Monitor
  - Runs on bare metal
- Partition resources through different Guest hosts
  - May provide emulated hardware
  - May support passthrough
- Each guest OS uses emulated hardware as it was real

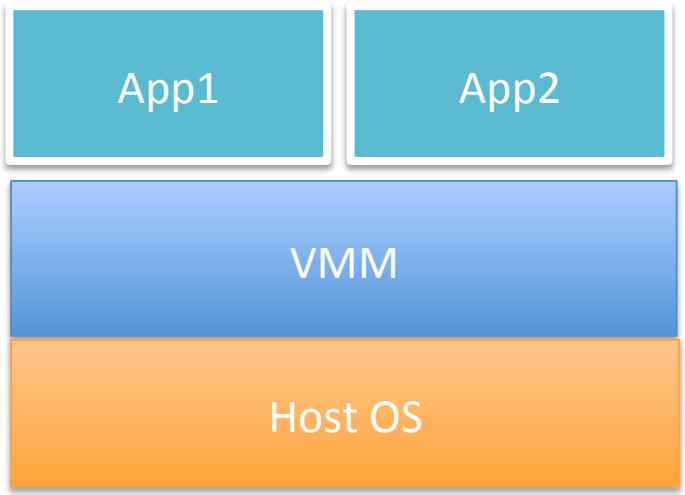


Virtualization Methods

- OS level virtualization
- Hosted Virtualization
- Bare Metal Virtualization
- Para-Virtualization
- Hardware Assisted Virtualization

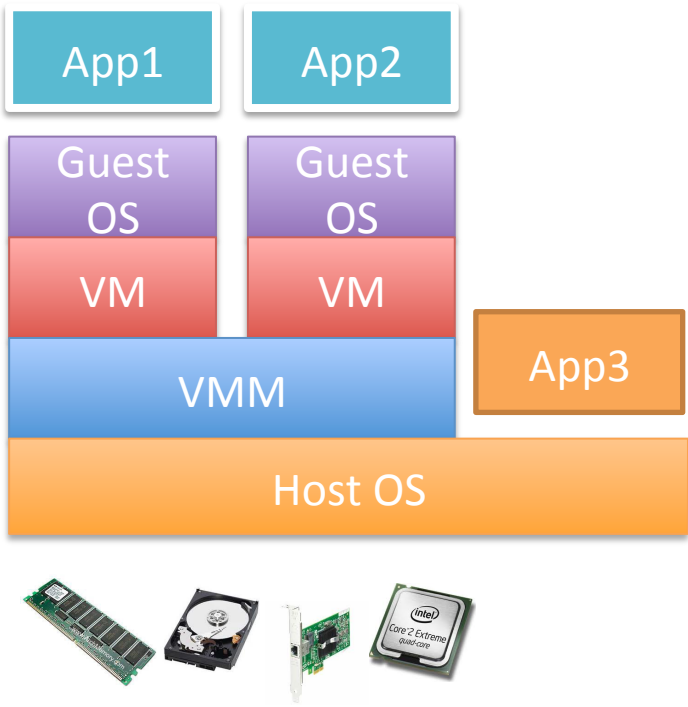
OS level virtualization

- Same Operating System for all applications
  - HW may be partitioned
  - HW is not emulated
- Advantages
  - Low Overhead
  - Maximum performance
  - Maximum resource reusal
- Disadvantages
  - Isolation
  - Stability
- Ex: OpenVZ, chroot, Virtuozzo, Docker



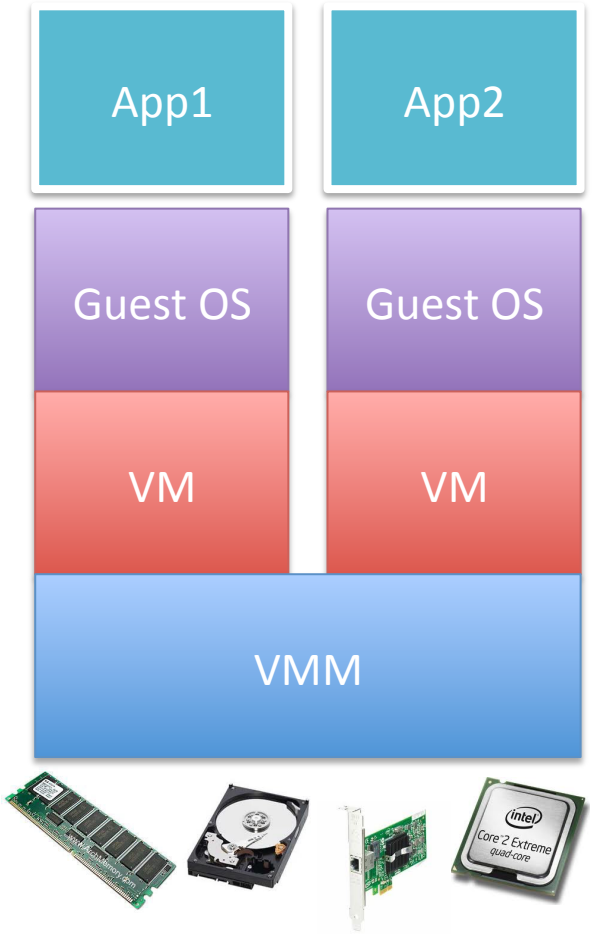
Hosted Virtualization

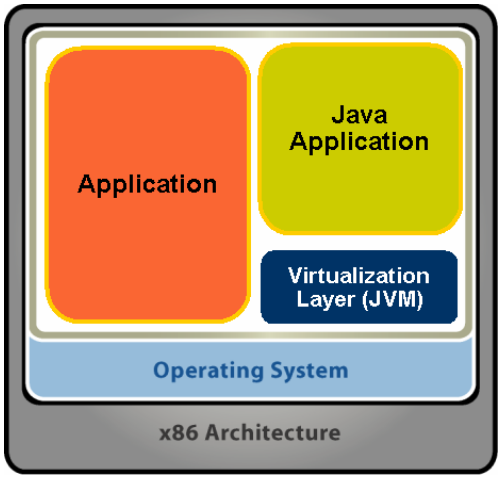
- Application or Host OS provides emulated hardware to guests
  - Each guest sees individual hardware
  - Hardware support varies with host OS
- Advantages
  - Compatibility
  - Isolation
  - Unmodified OS
- Disadvantages
  - More resources required
  - Lower performance
    - Full OS running
- Ex: Virtualbox, VMWare Server, QEMU



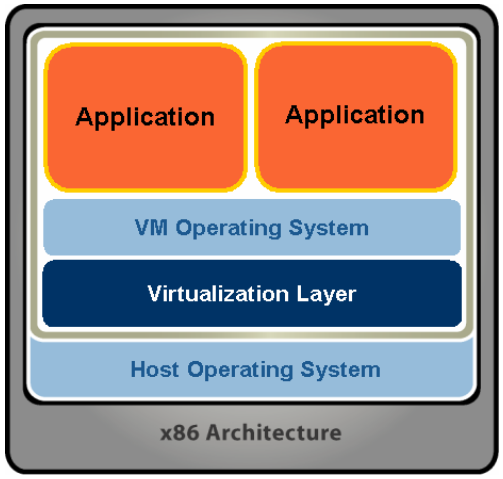
Bare Metal Virtualization

- Hypervisor executes on hardware
  - No full blown host OS
  - Each guest sees individual hardware
- Advantages
  - Stability
  - Isolation
  - Unmodified OS
  - Higher Performance
- Disadvantages
  - Hypervisor supports reduced set of hardware devices
- Ex: VMWare ESX

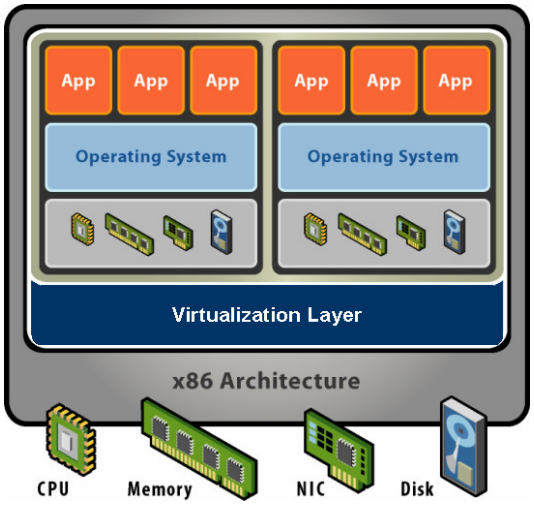




Language Level



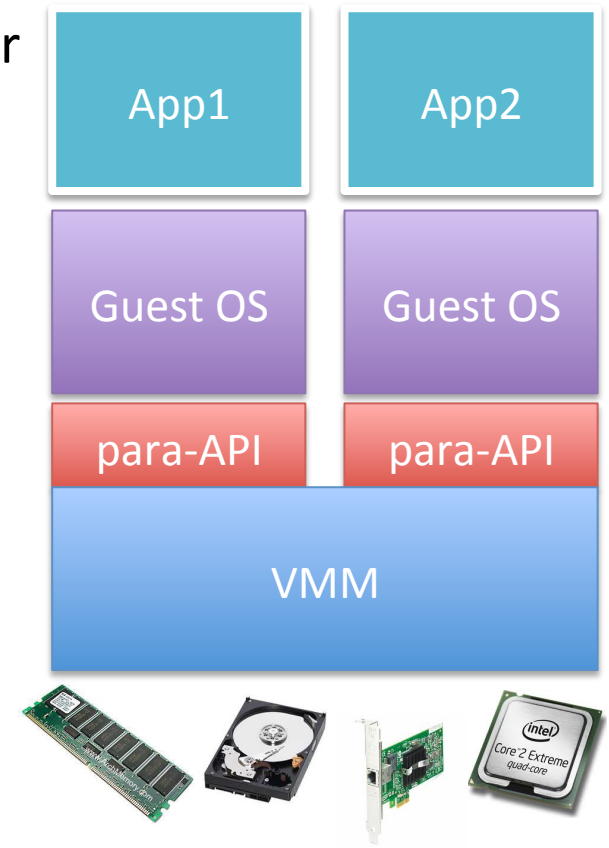
OS Level



Hardware Level

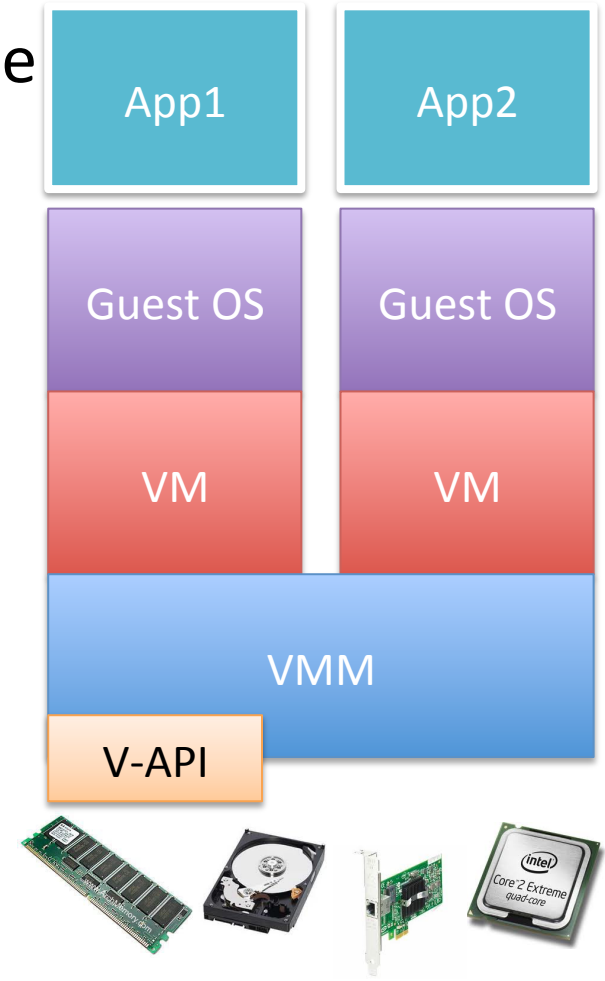
Para-Virtualization

- Bare metal hypervisor with para-API
  - Guest OS must/may interact with hypervisor
  - Para-Virtualized hardware is provided
- Advantages
  - Higher performance
    - No need to fully emulate hardware
    - Particular important to IO
  - Resource usage
- Disadvantages
  - Stability (para-API changes)
  - Requires support of guest OS
- Ex: VMWare ESX, XEN, KVM



Hardware Assisted Virtualization

- Hardware provides means to accelerate virtualization
  - HVM – Hardware Virtual Machines
- Advantages
  - Unmodified OS
  - Maximum performance
  - Stability
- Disadvantages
  - Hardware vendor lock in
  - CPUs becomes more complex
- Ex: IBM System/370, Intel VTx, AMD-V



How to virtualize - Storage

- Hard disks
  - Real device
    - Maximum performance, lower flexibility
  - Fixed size: pre-allocated with maximum size
    - Higher performance, wasted storage capacity
  - Dinamic Size: allocated as needed
    - Higher efficiency, lower performance
  - Remote Disk: Stored in SAN/NAS
    - Maximum flexibility, lower performance
- Optical drives
  - Existing image
  - Passthrough real device

How to virtualize - Network

- Host Only – A TAP interface connects host to guest
- Internal Network – Communication with a virtual network just for VMs
- NAT – guest is behind NAT
  - Can access the outside, but cannot receive connections
- Bridged – A real network device is connected to the guest through a virtual switch

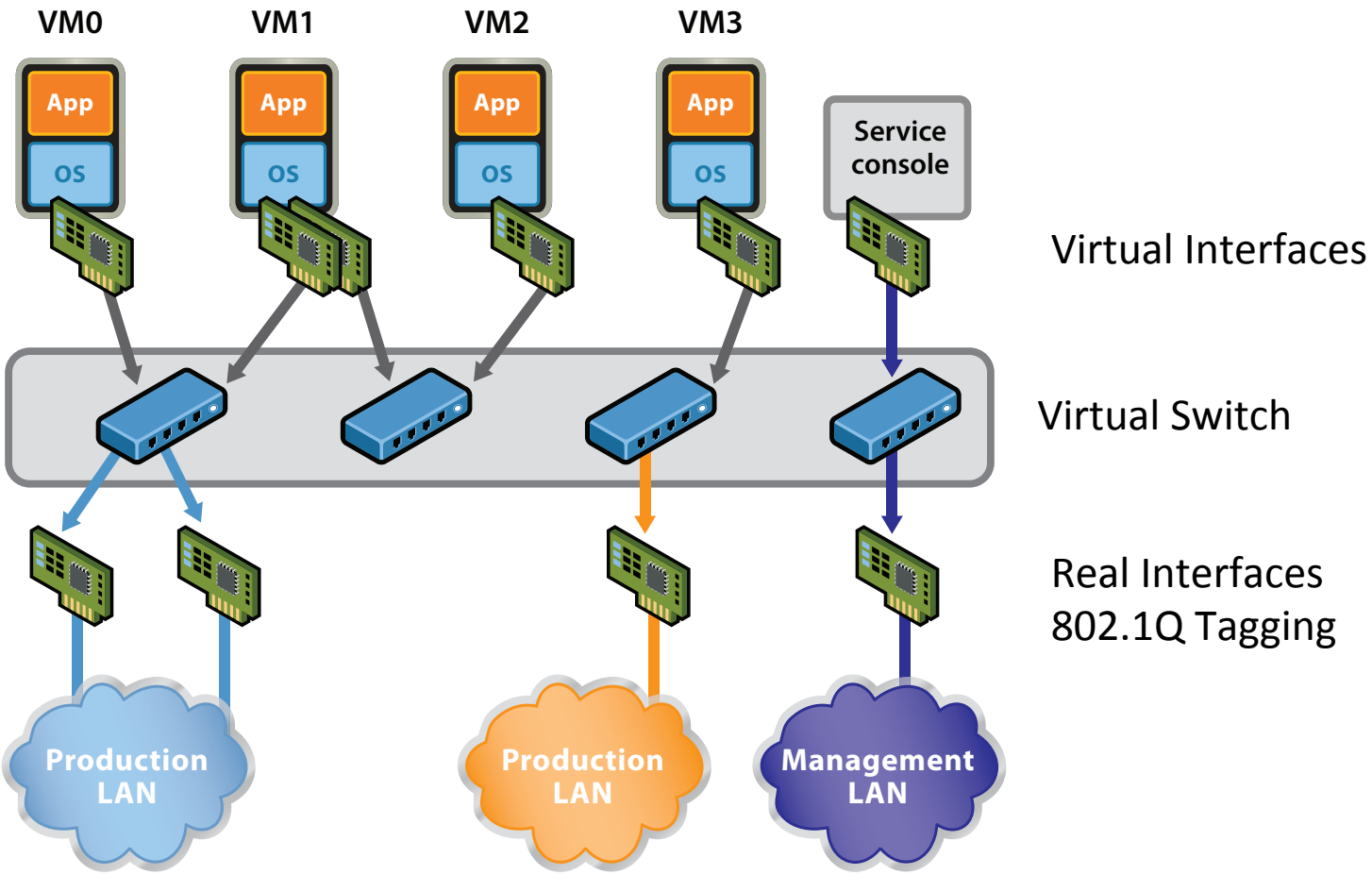
How to virtualize - Network

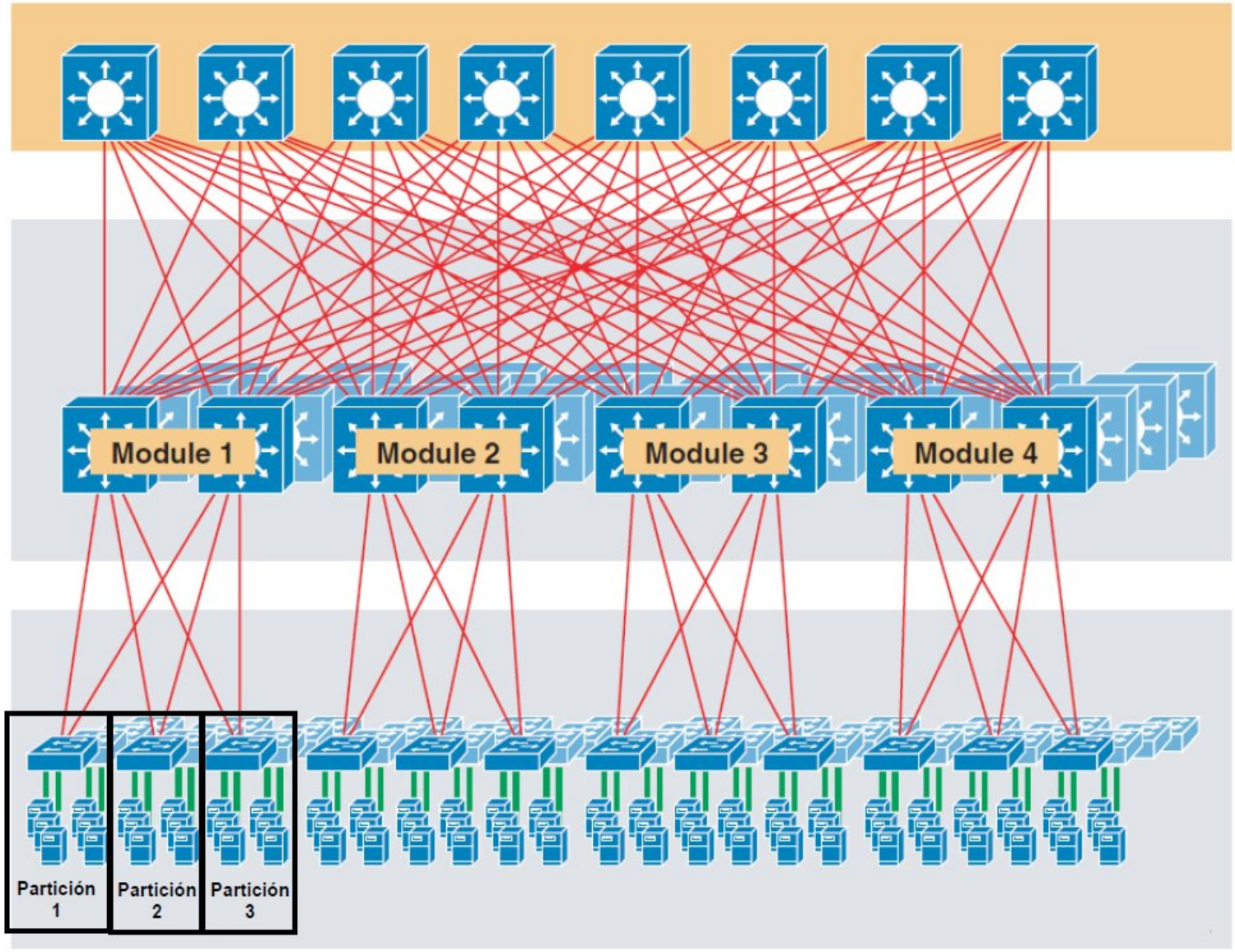
- Applications are isolated inside VMs, their communication isn't
  - Neighbor guests are able to eavesdrop/forge traffic
- Neighbors inside same host
- Neighbors on a close host

How to virtualize - Network

- 802.1Q VLAN
- Each domain (costumer) in its VLAN
  - 4096 IDs available
- Routers filter access between VLANs
  - L3 Routing between VLANs
- Problems: number grows rapidly
  - Switches are severely limited <150 VLANs
    - UA uses around 120.
  - Hypervisor must support 802.1Q mapping
  - Burden to manage
  - Top of Rack switches must handle MACs and VLANs for all VMs
    - 20 blades, 100 VMs per blade

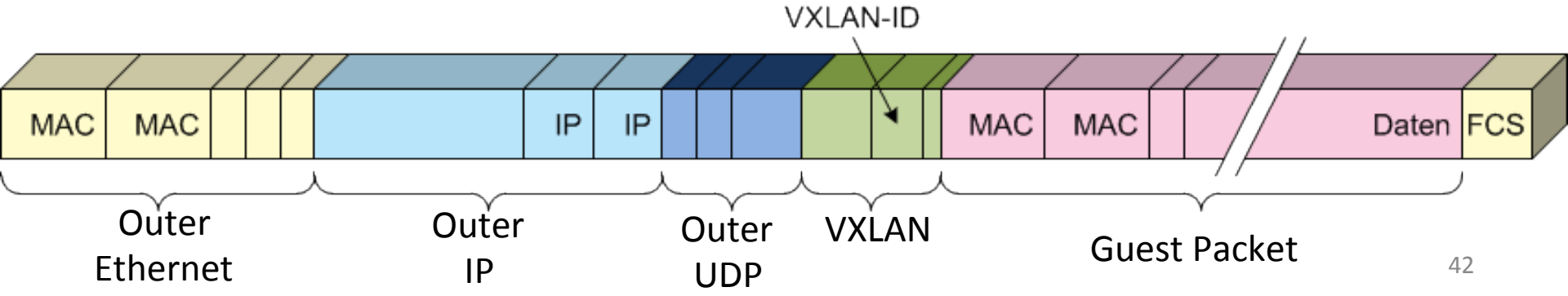
How to virtualize - Network




Module 1
Module 2
Module 3
Module 4
Partición 1
Partición 2
Partición 3

How to virtualize - Network

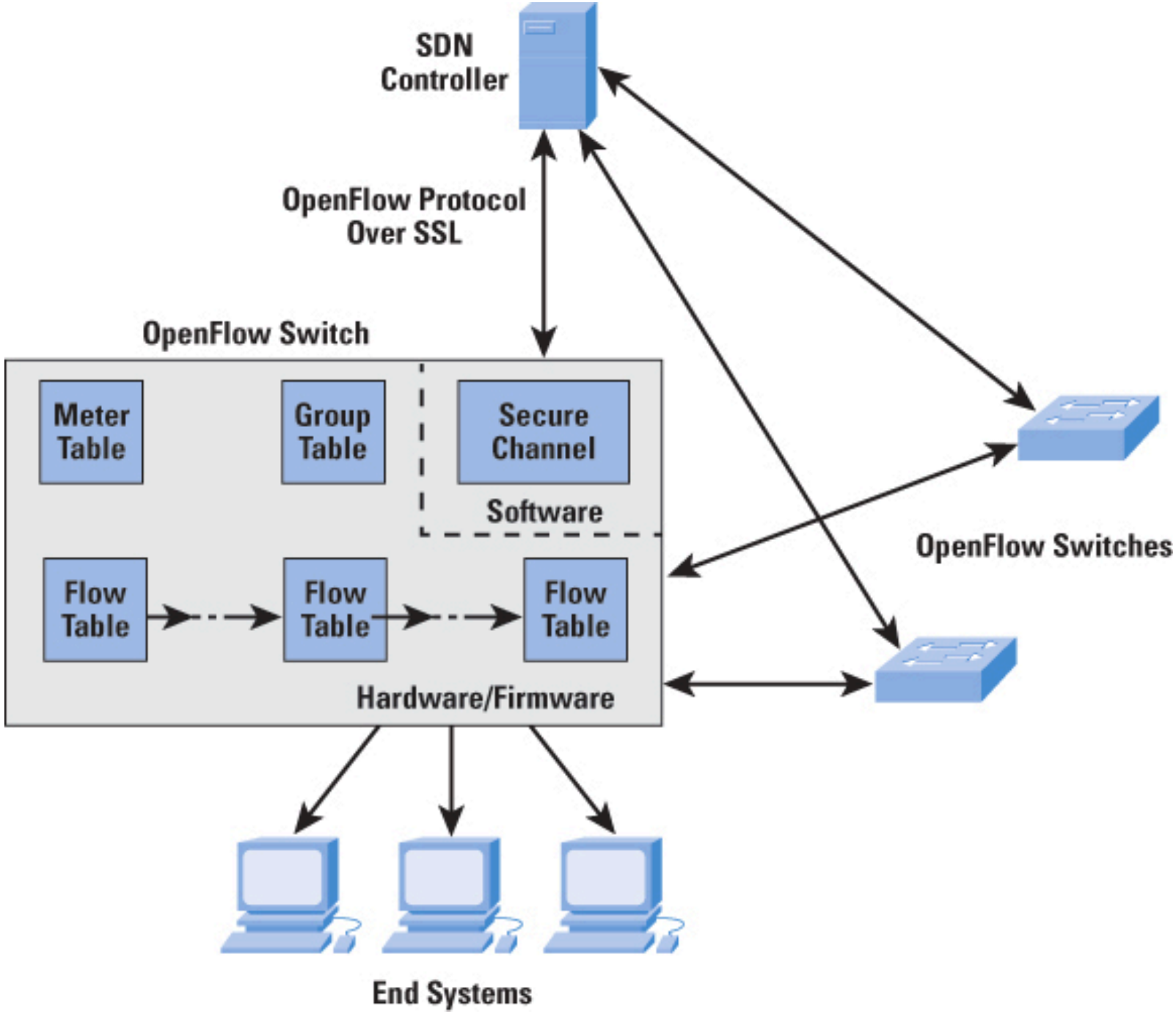
- VXLAN – Ethernet over UDP
  - Outer Addresses are from host
  - Inner Address are from guest
  - VXLAN header provides 24bit ID
  - Problems:
    - Hardware support is limited to high end equipment
    - Overhead



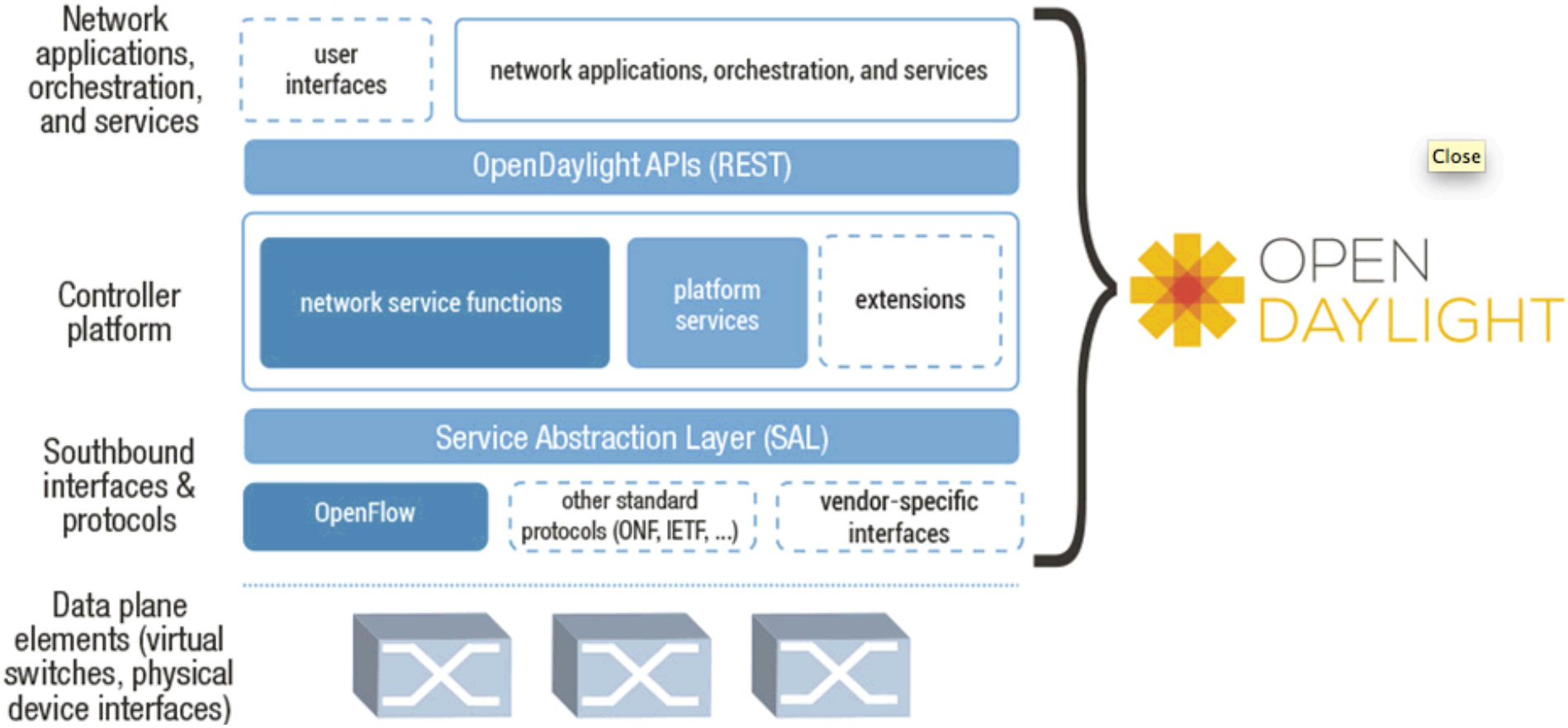
How to virtualize - Network

- Software Defined Networks
  - Integrated with virtualization platform
    - OpenStack, OpenDayLight
  - Departure from VLAN/Spanning Tree concepts

- OpenFlow based Switching
  - Central controller for all L2 infrastructure


SDN
Controller
OpenFlow Protocol
Over SSL
OpenFlow Switch
Meter
Table
Group
Table
Secure
Channel
Software
Flow
Table
Flow
Table
Flow
Table
Hardware/Firmware
OpenFlow Switches
End Systems

OpenDaylight

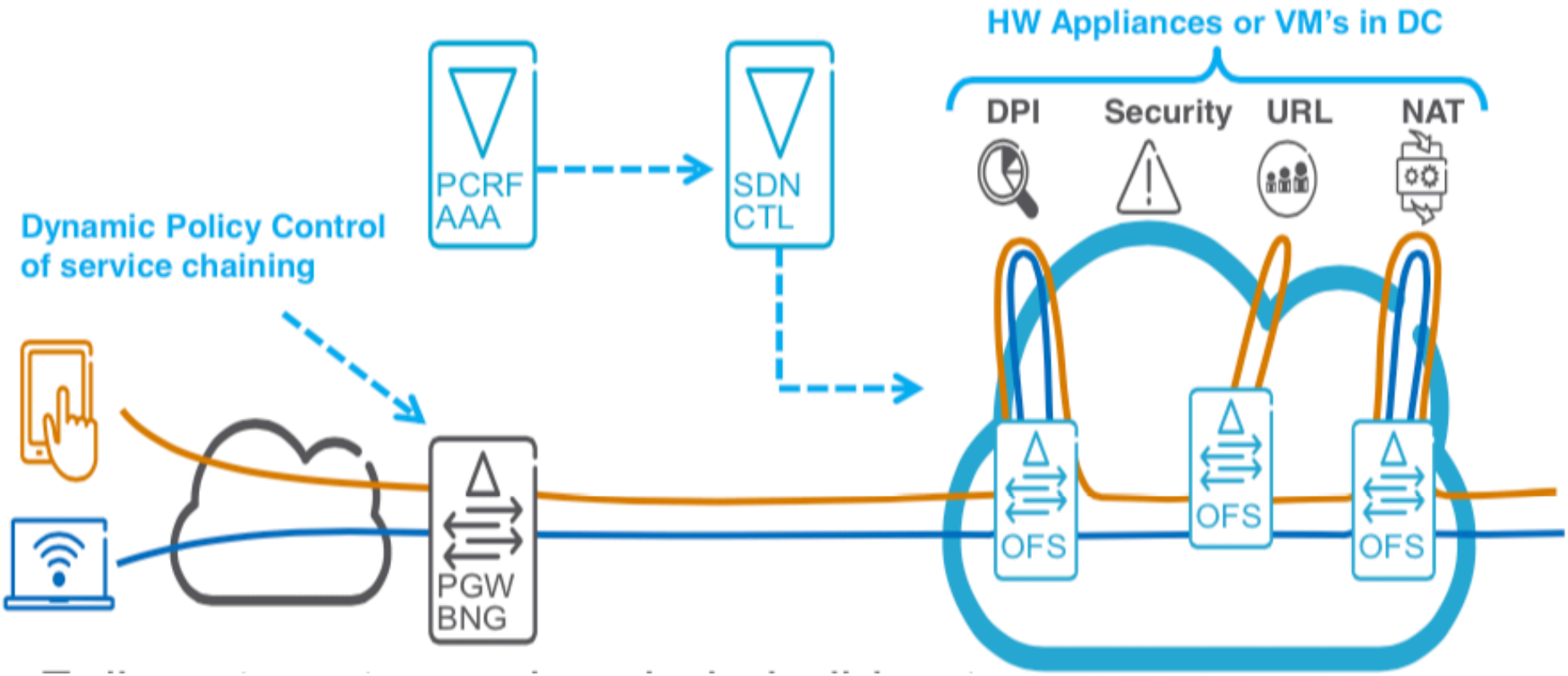


How to virtualize - Network

- Network Function Virtualization

- Nework functions also virtualized (besides computation instances)
  - DHCP
  - DNS servers
  - Routers
  - Home Gateways
  - Gaming consoles

Virtual CPE

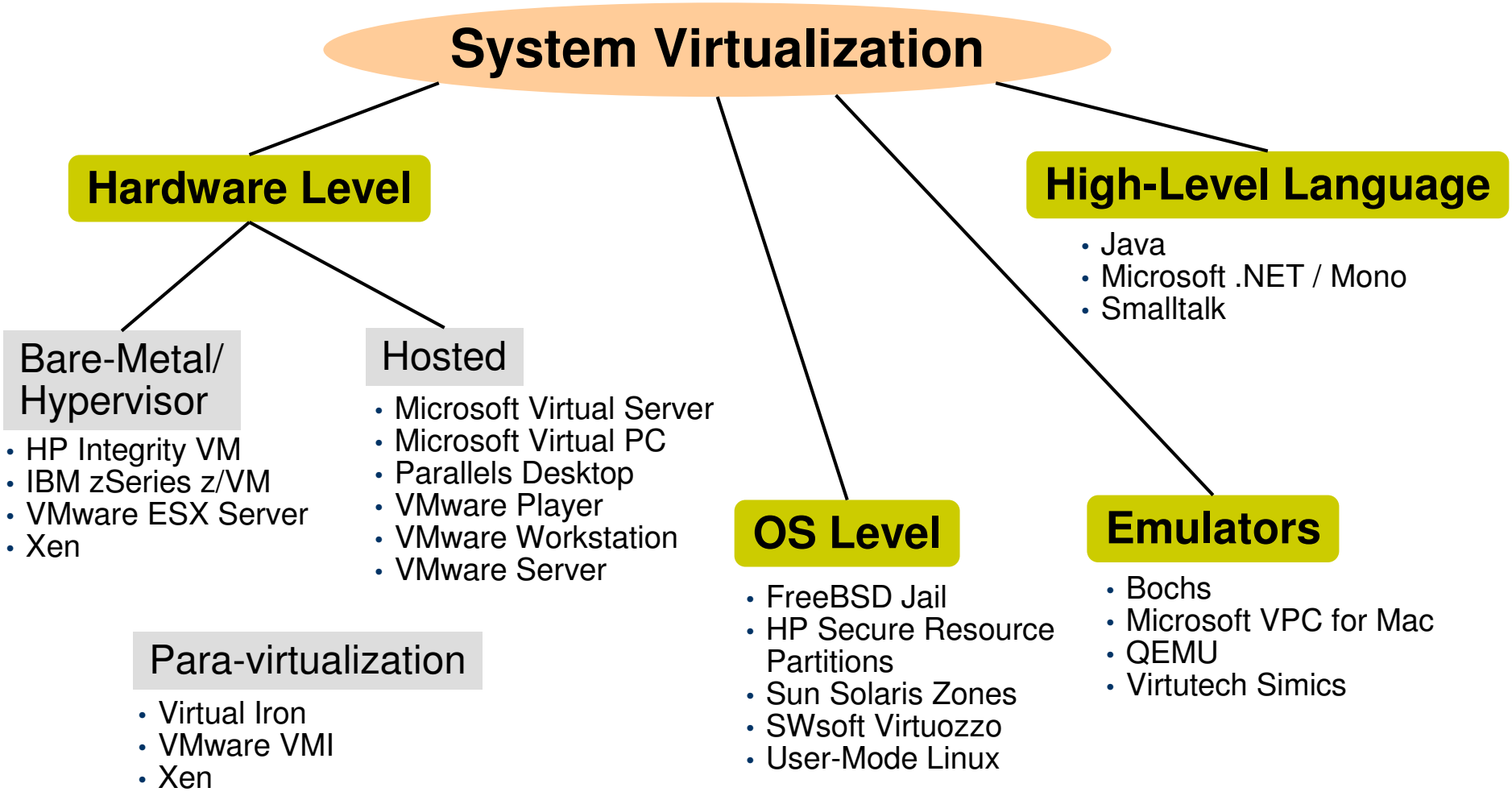

HW Appliances or VM's in DC
DPI
Security
URL
NAT
PCRF
AAA
SDN
CTL
Dynamic Policy Control
of service chaining
PGW
BNG
OFS
OFS
OFS

How to virtualize - Misc

- PCI, SCSI and USB devices: Passthrough
  - Real hardware is mapped to guest
- SD/MMC readers: emulated
- Graphics card: emulated
  - VESA interface provided

How to virtualize - Memory

- Memory is almost directly available
  - Total guest memory can be higher than host memory: Overcommit
- Ballooning: guests keep memory (buffers, caches, etc..)
  - If VMM needs RAM it may allocate memory inside guests (with driver), pin memory and then free it.
- Swapping: low used pages are sent to swap file
  - Can be compressed before write to disk
- Page Sharing: hypervisor may find similar pages and keep only one copy
  - Copy on write


System Virtualization
Hardware Level
Bare-Metal/ Hypervisor
• HP Integrity VM
• IBM zSeries z/VM
• VMware ESX Server
• Xen
Hosted
• Microsoft Virtual Server
• Microsoft Virtual PC
• Parallels Desktop
• VMware Player
• VMware Workstation
• VMware Server
Para-virtualization
• Virtual Iron
• VMware VMI
• Xen
OS Level
• FreeBSD Jail
• HP Secure Resource Partitions
• Sun Solaris Zones
• SWsoft Virtuozzo
• User-Mode Linux
High-Level Language
• Java
• Microsoft .NET / Mono
• Smalltalk
Emulators
• Bochs
• Microsoft VPC for Mac
• QEMU
• Virtutech Simics

Assignment

- Choose a virtualization solution
- Evaluate the solution under the scope of a service provider
  - Multiple users, dynamic networking, dynamic allocation, quotas, remote storage
- Prepare some slides explaining the main points, pros and cons of the solution.
  - Focus in features and performance

Further Reading

- Keith Adams and Ale Agesen, “A Comparison of Software and Hardware Techniques for x86 Virtualization”, (ASPLOS 2006)
- Fast Transparent Migration of Virtual Machines (USENIX 2005)
- Paul Barham, et al. , “Xen and the Art of Virtualization”, (SOPS 2003)
- J. E. Smith and Ravi Nair, “An Overview of Virtual Machine Architectures”