Virtulization

Nelson L. S. da Fonseca

IEEE ComSoc Summer Scool

Albuquerque, July 17-21, 2017

Acknowledgement

- Some slides in this set of slides were kindly provided by:
  - Luiz Fernando Bittencourt, University of Campinas
  - EMC Corporation

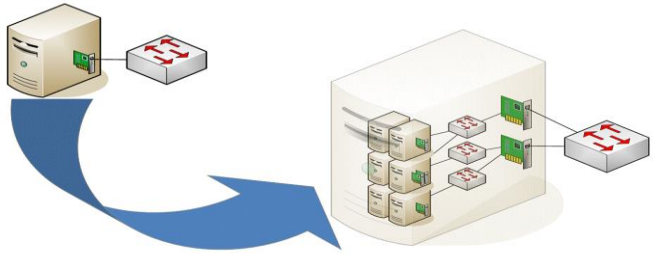
Virtualization

"Virtualization means to create a virtual version of a device or resource, such as a server, storage device, network or even an operating system where the framework divides the resource into one or more execution environments. Devices, applications and human users are able to interact with the virtual resource as if it were a real single logical resource."

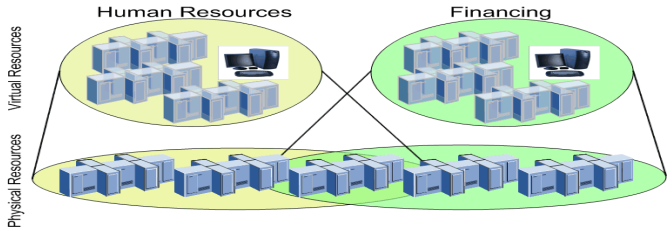
http://www.webopedia.com/TERM/V/virtualization.html

Virtualization - Features

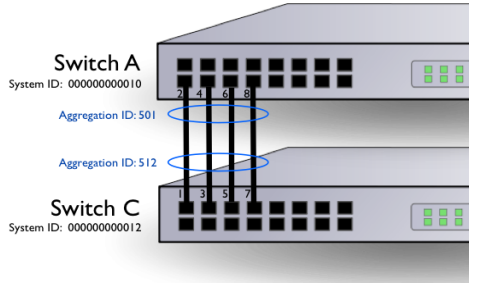
- Sharing of resources



- Isolation



- Agregation



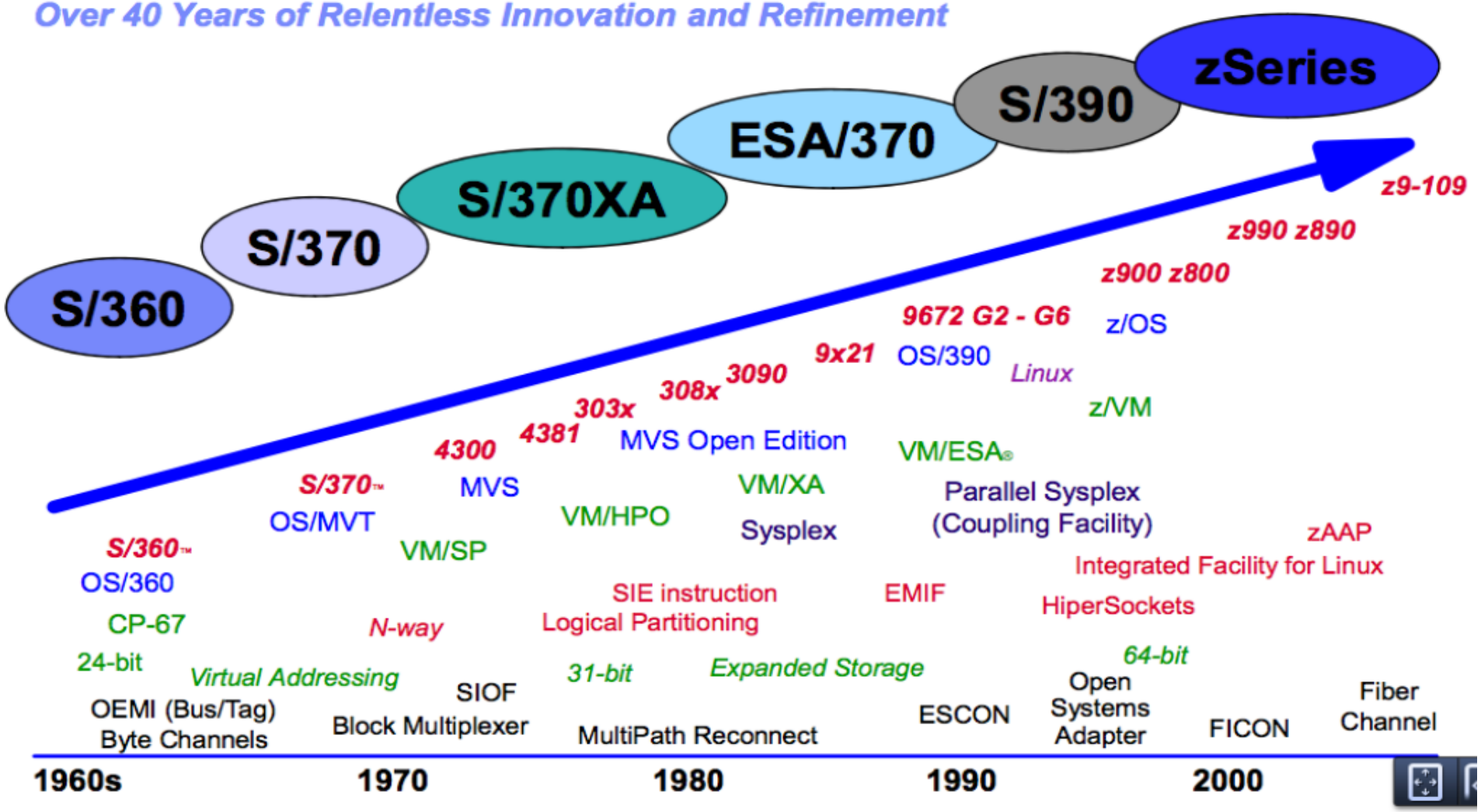
Virtualization - advantages

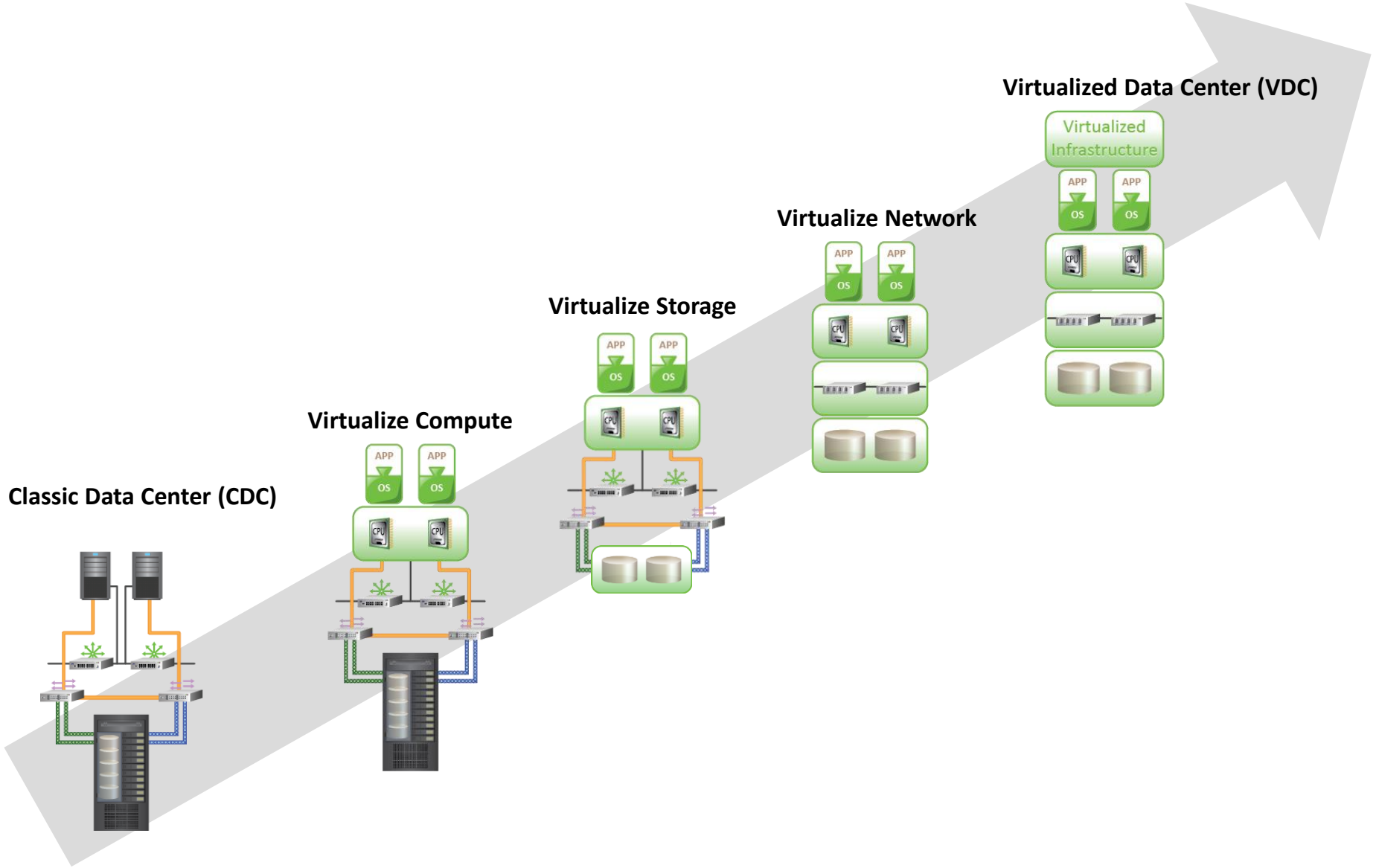
- Fast deployment
- Scalability
- Load consolidation
- Flexibility
- Mobility
- Green

Agenda:

IBM Mainframe Technology Evolution

Over 40 Years of Relentless Innovation and Refinement

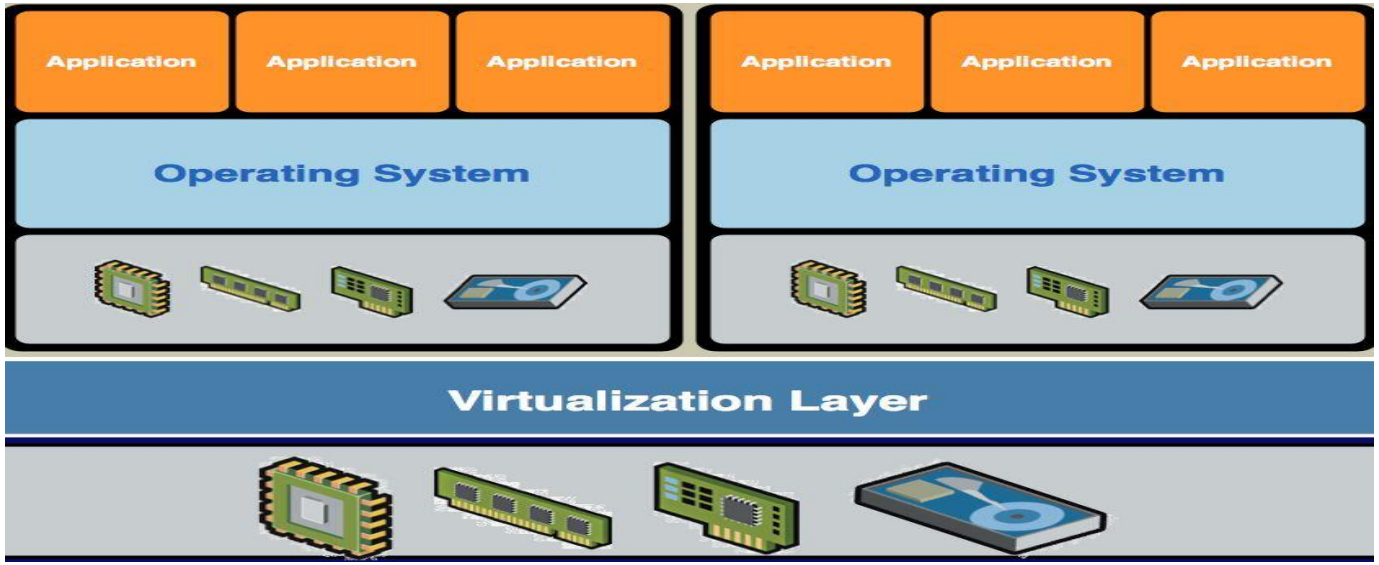


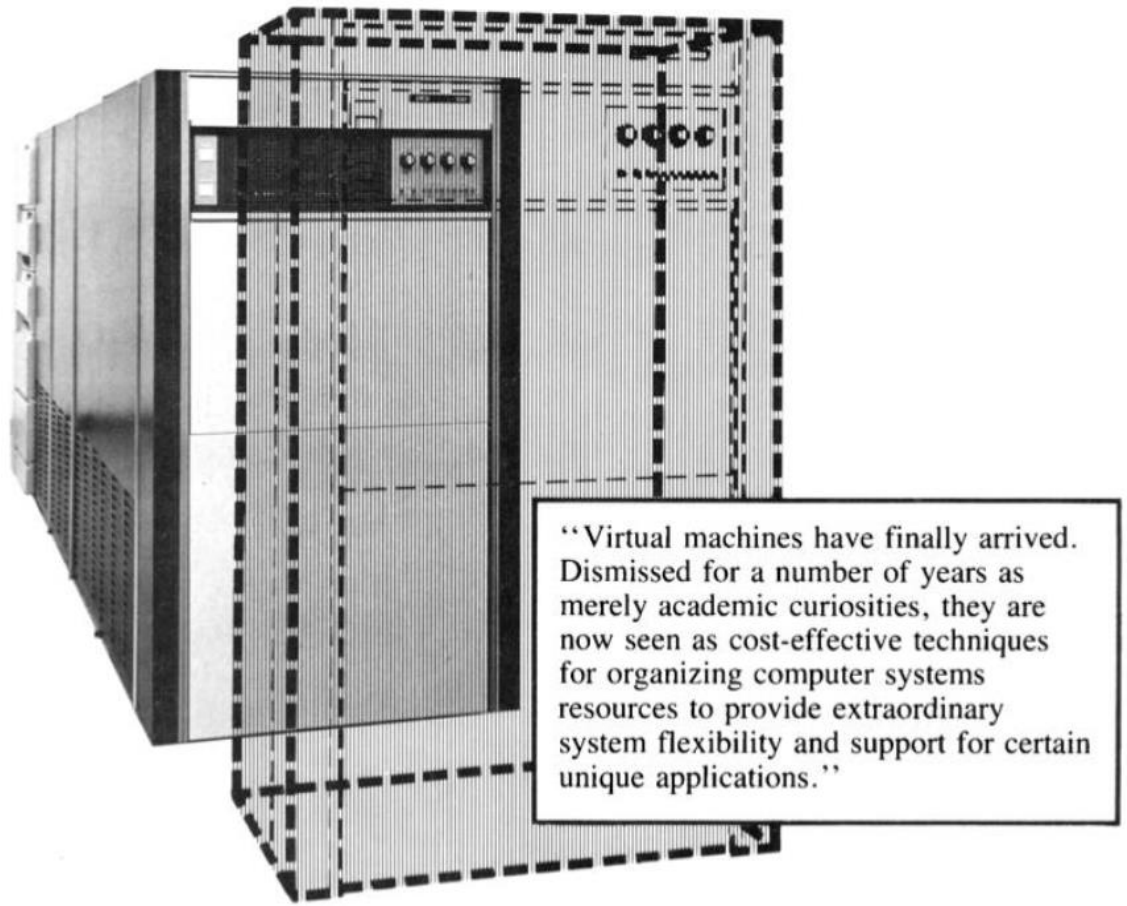

Classic Data Center (CDC)
Virtualize Compute
Virtualize Storage
Virtualize Network
Virtualized Data Center (VDC)
Virtualized Infrastructure
APP
OS
CPU

Server Virtualization

Compute Virtualization

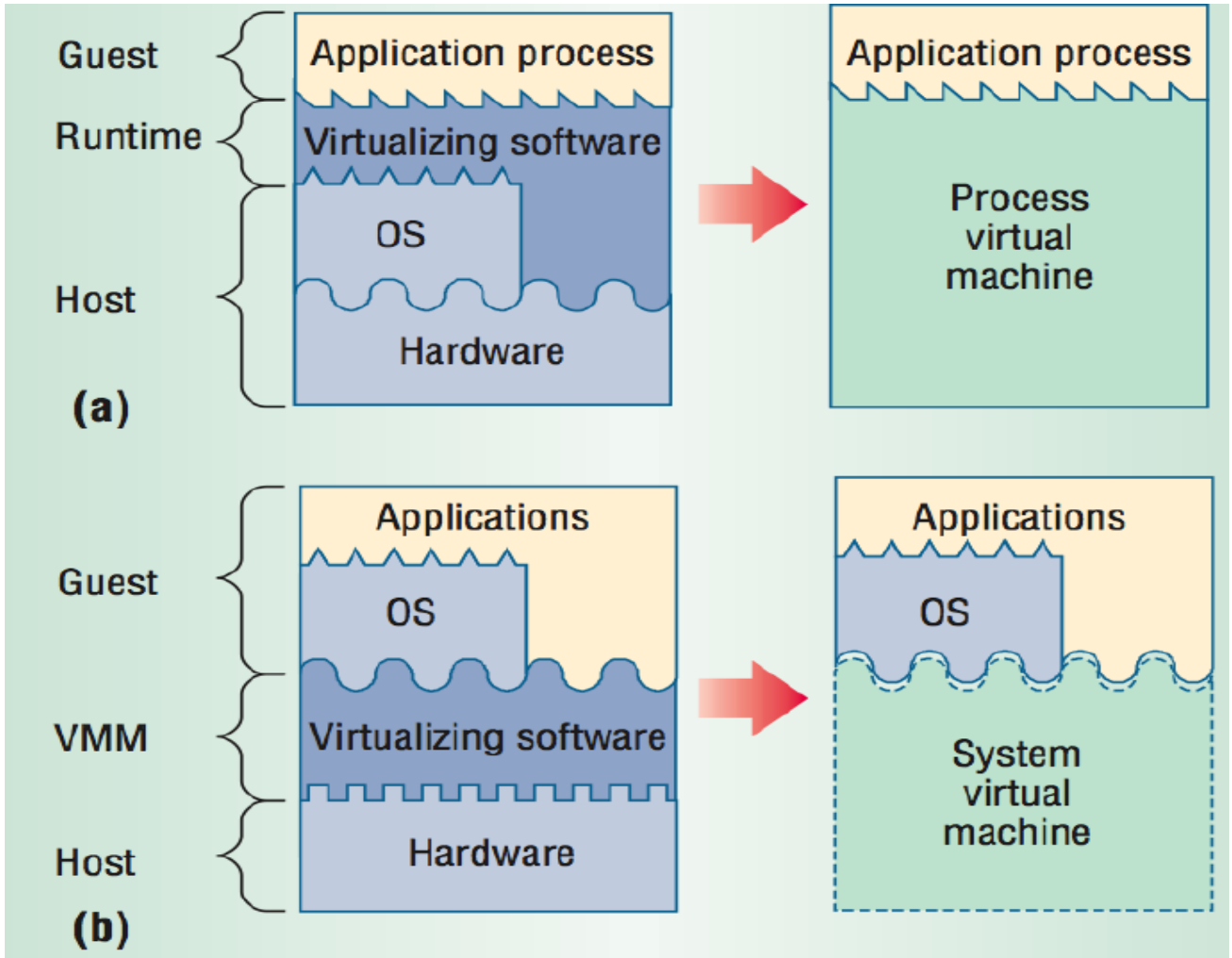
- It is a technique of masking or abstracting the physical compute hardware and enabling multiple operating systems (OSs) to run concurrently on a single or clustered physical machine(s).
- Virtual Machine is a logical entity that looks and behaves like physical machine
- Virtualization layer resides between hardware and VMs (hypervisor)
- VMs are provided with standardized hardware resources





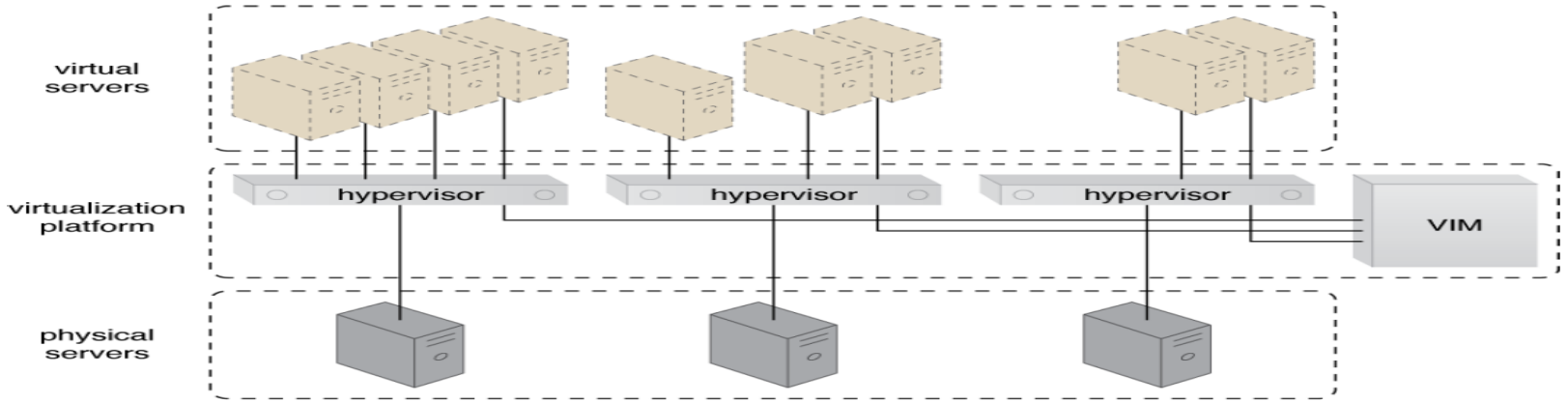
"Virtual machines have finally arrived. Dismissed for a number of years as merely academic curiosities, they are now seen as cost-effective techniques for organizing computer systems resources to provide extraordinary system flexibility and support for certain unique applications."

Process Virtual Machine X System Virtual Machine

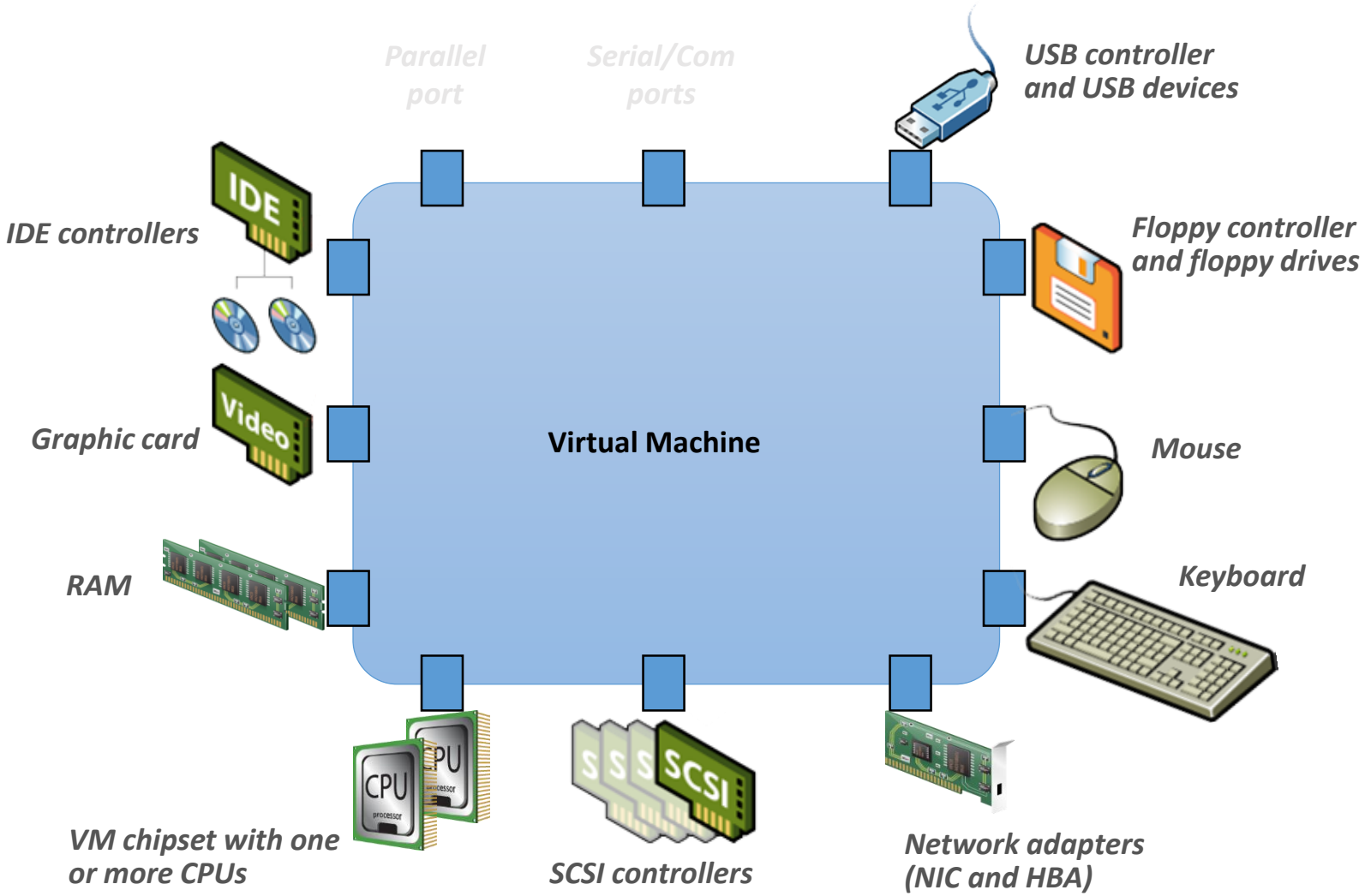


Virtual Machine

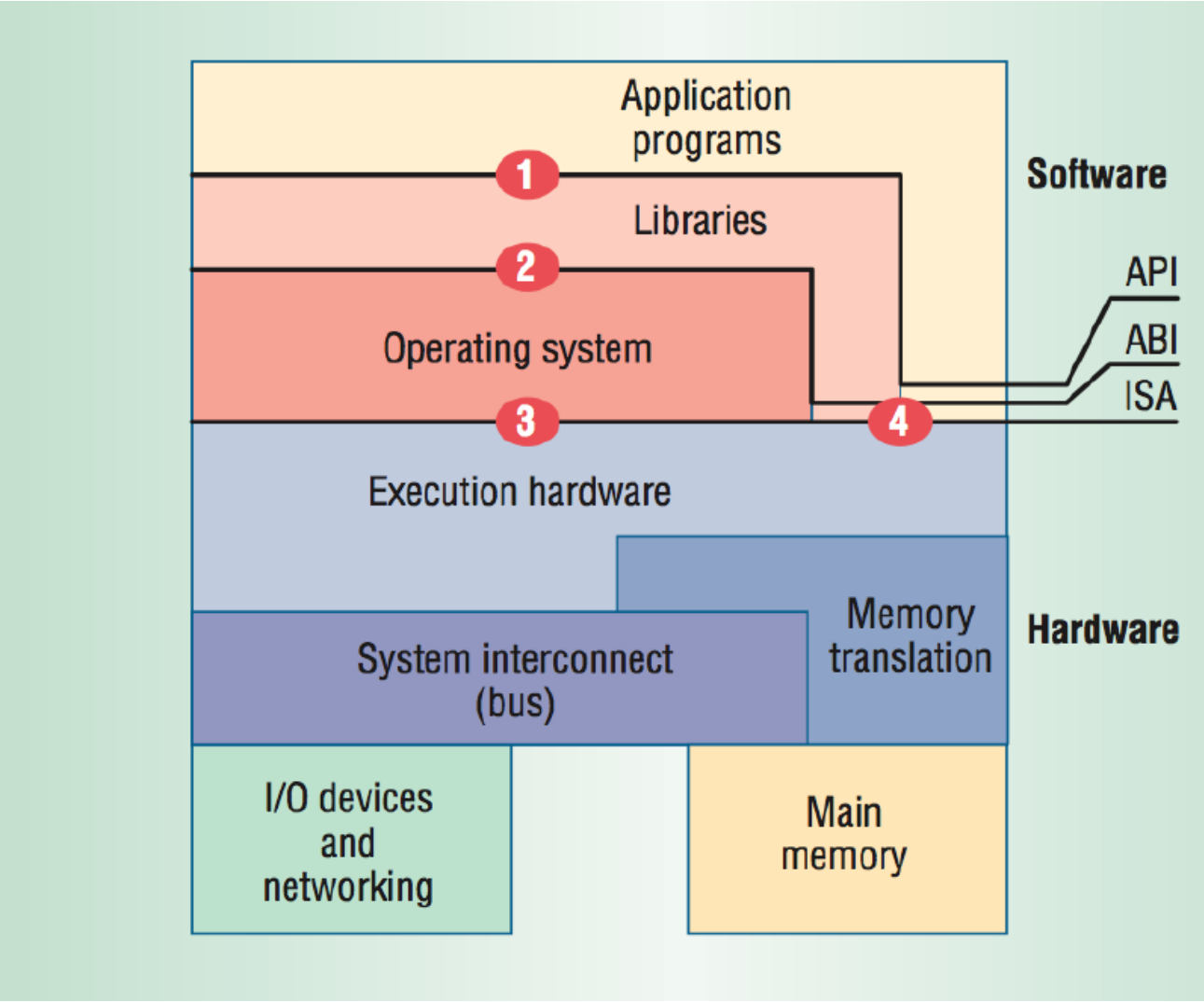
- From a user's perspective, a logical compute system
  - Runs an operating system (OS) and application like a physical machine
  - Contains virtual components such as CPU, RAM, disk, and NIC
- From a hypervisor's perspective
  - Virtual machine (VM) is a discrete set of files such as configuration file, virtual disk files, virtual BIOS file, VM swap file, and log file



Virtual Machine Hardware

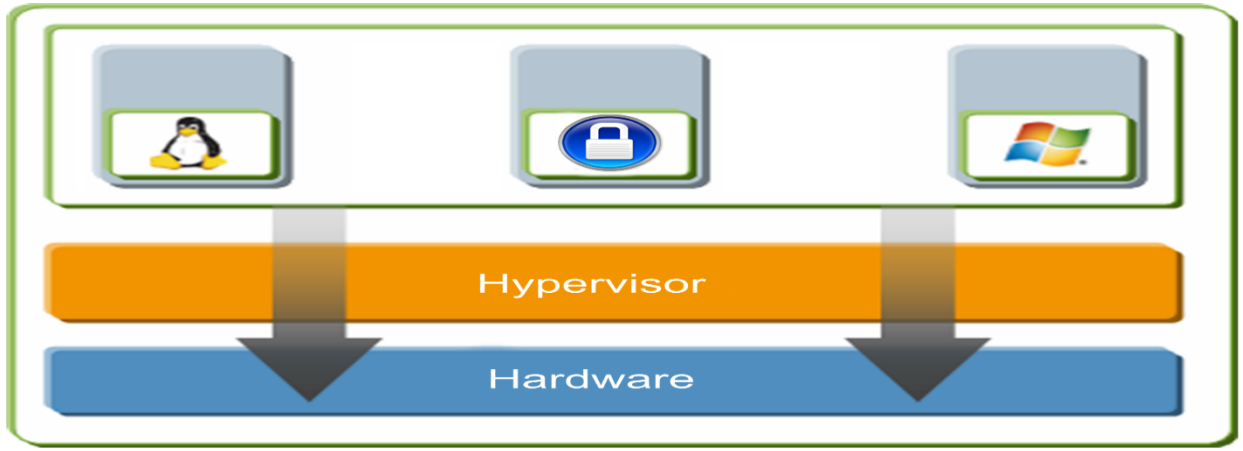


Interfaces


Application programs
1
Libraries
2
Operating system
3
4
Execution hardware
Memory translation
System interconnect (bus)
I/O devices and networking
Main memory
Software
Hardware
API
ABI
ISA

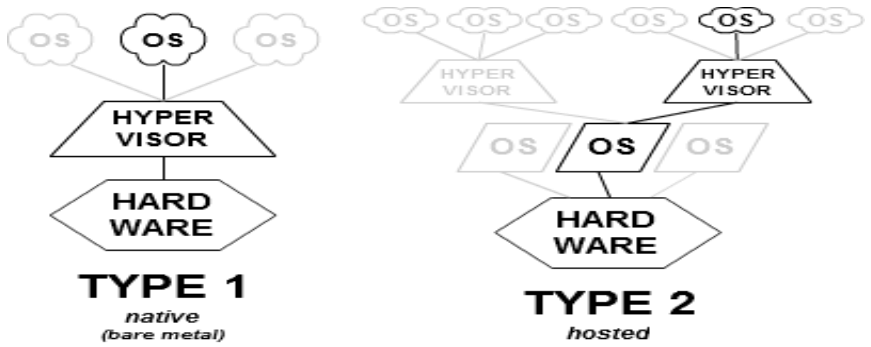
Hypervisor

- It is a software that allows multiple operating systems (OSs) to run concurrently on a physical machine and to interact directly with the physical hardware.
- Has two components
  - Kernel
  - Virtual Machine Monitor (VMM)



Type of Hypervisor

- Bare Metal Hypervisors: run directly on the host's hardware to control the hardware and to manage guest operating systems
  - XenServer, VMware ESX/ESXi and Microsoft Hyper-V
- Hosted hypervisors: run on a conventional operating system just as other computer programs do
  - VMware Workstation/Player and VirtualBox
- https://en.wikipedia.org/wiki/Comparison_of_platform_virtualization_software



Type of Virtualization

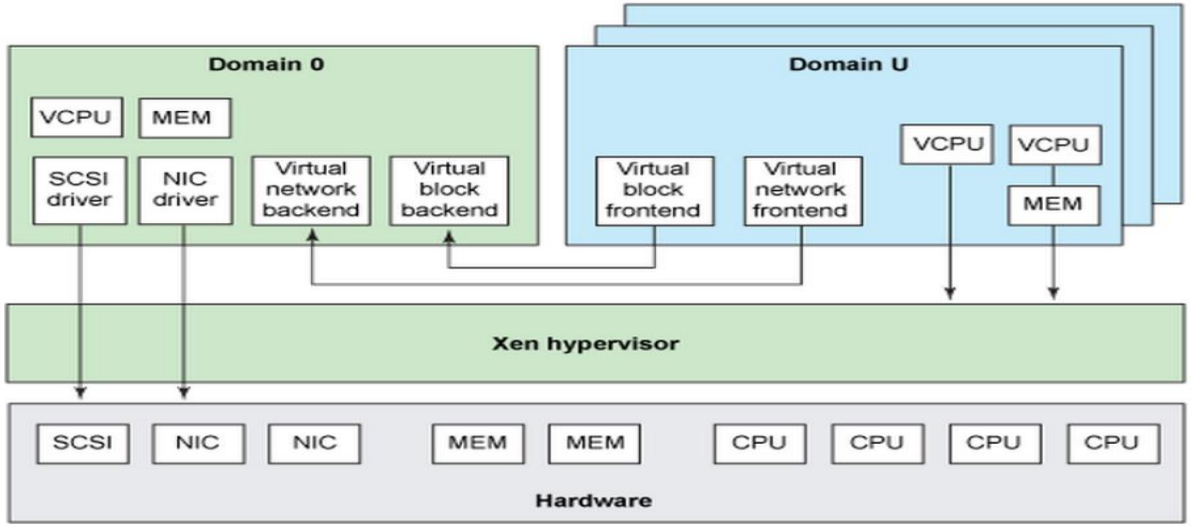
	Full Virtualization with Binary Translation	Hardware Assisted Virtualization	OS Assisted Virtualization / Paravirtualization
Technique	Binary Translation and Direct Execution	Exit to Root Mode on Privileged Instructions	Hypercalls
Guest Modification / Compatibility	Unmodified Guest OS Excellent compatibility	Unmodified Guest OS Excellent compatibility	Guest OS codified to issue Hypercalls so it can't run on Native Hardware or other Hypervisors Poor compatibility; Not available on Windows OSes
Performance	Good	Fair Current performance lags Binary Translation virtualization on various workloads but will improve over time	Better in certain cases
Used By	VMware, Microsoft, Parallels	VMware, Microsoft, Parallels, Xen	VMware, Xen
Guest OS Hypervisor Independent?	Yes	Yes	XenLinux runs only on Xen Hypervisor VMI-Linux is Hypervisor agnostic

VMWare

- VMWare Workstation 1.0 (Windows e Linux) launched in 1999
- Bynary translation and direct execution on hardware
- Instructions trapped and handled by Hypervisor

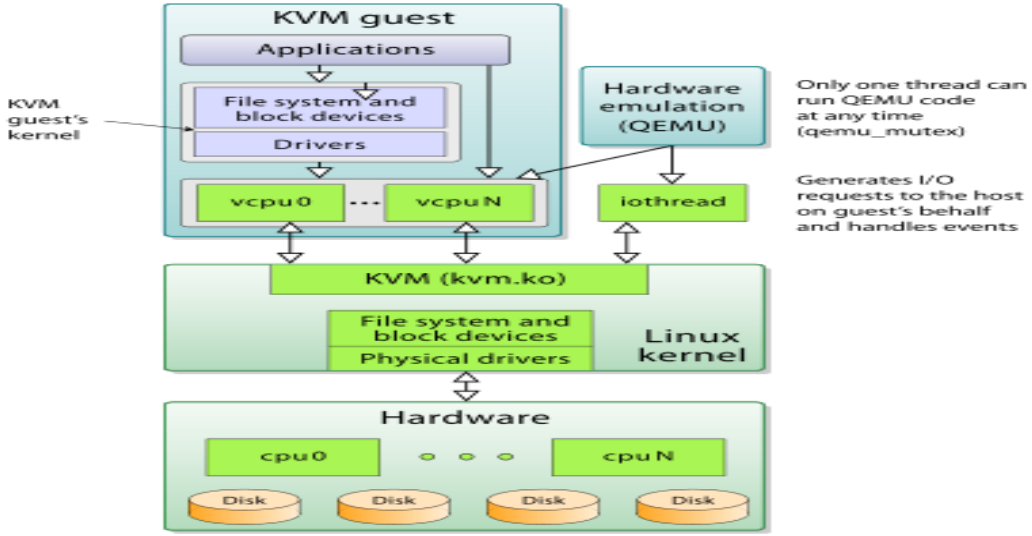
Xen

- Launching the most privileged domain ("dom0") - the only virtual machine which by default has direct access to hardware. From the dom0 the hypervisor can be managed and unprivileged domains ("domU") can be launched. The dom0 domain is typically a version of Linux, or BSD.
- Paravirtualization



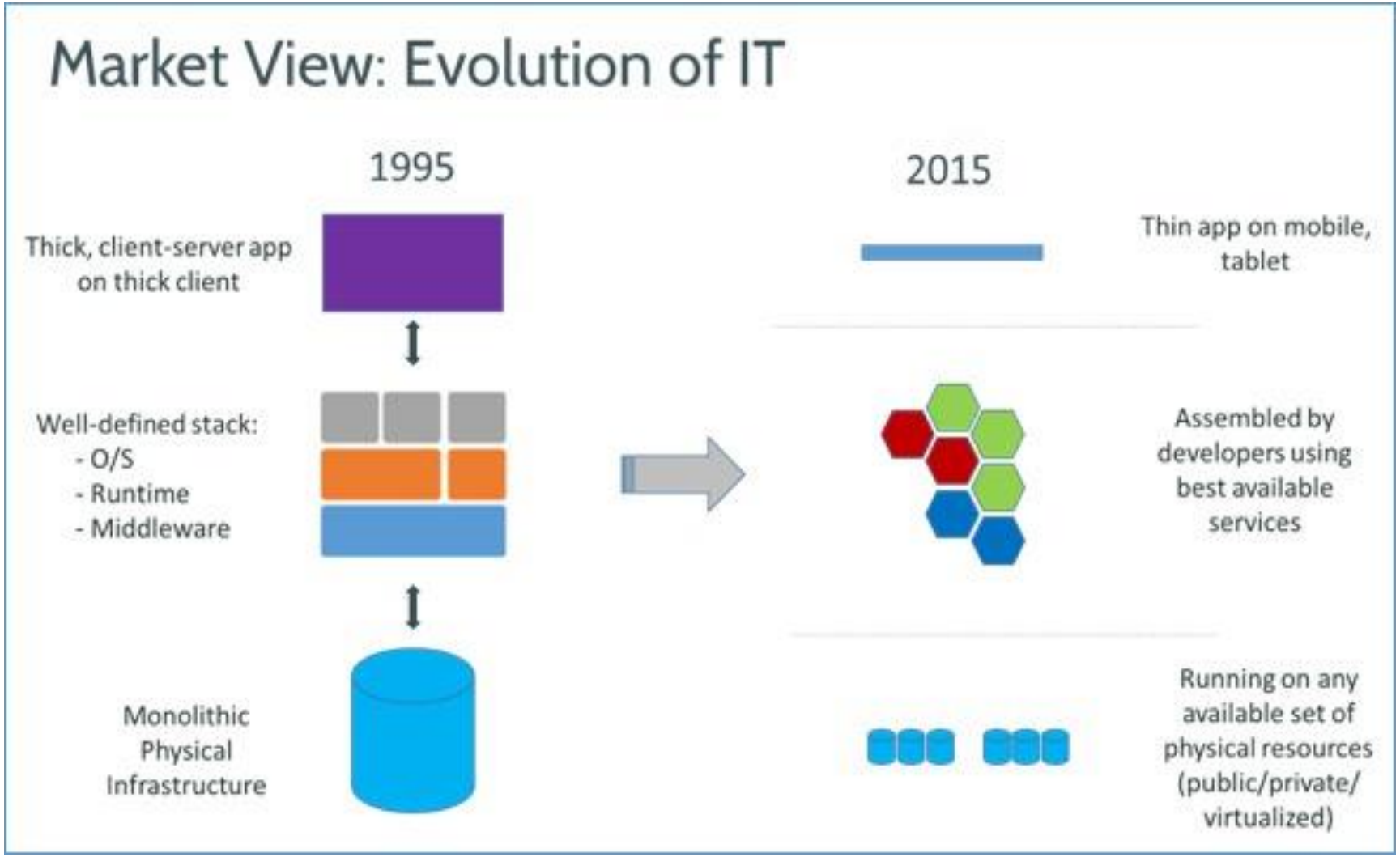
KVM

- **Kernel-based Virtual Machine** is a virtualization infrastructure for the Linux kernel that turns it into a bare metal hypervisor, which was merged into the Linux kernel mainline in February 2007
- KVM requires a processor with hardware virtualization extension
- KVM has also been ported to FreeBSD in the form of loadable kernel modules



Containers

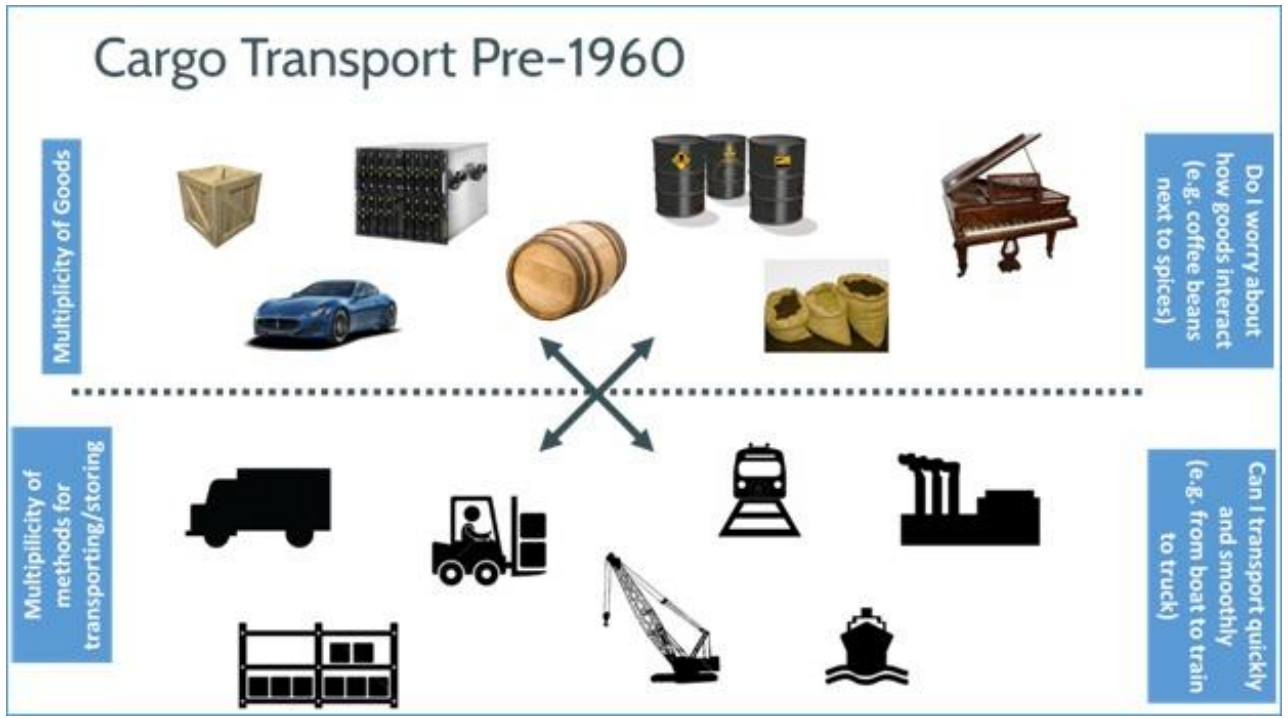
Linux Containers



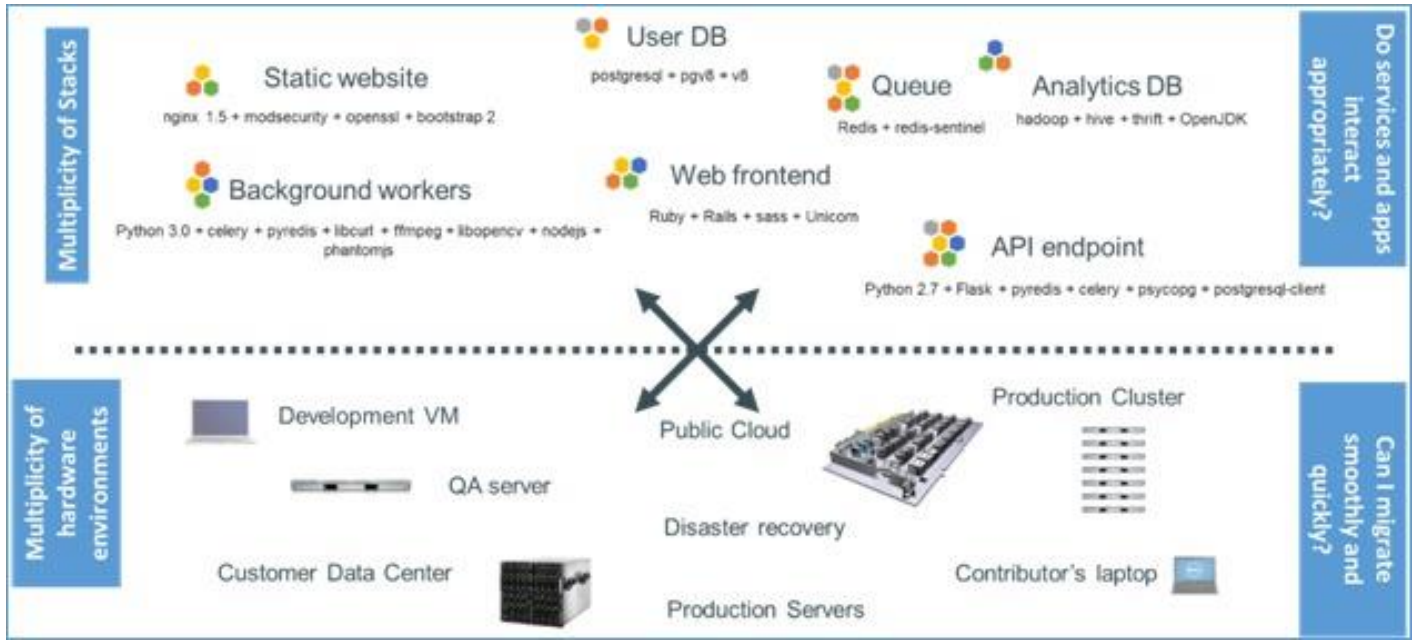
What are Linux containers?

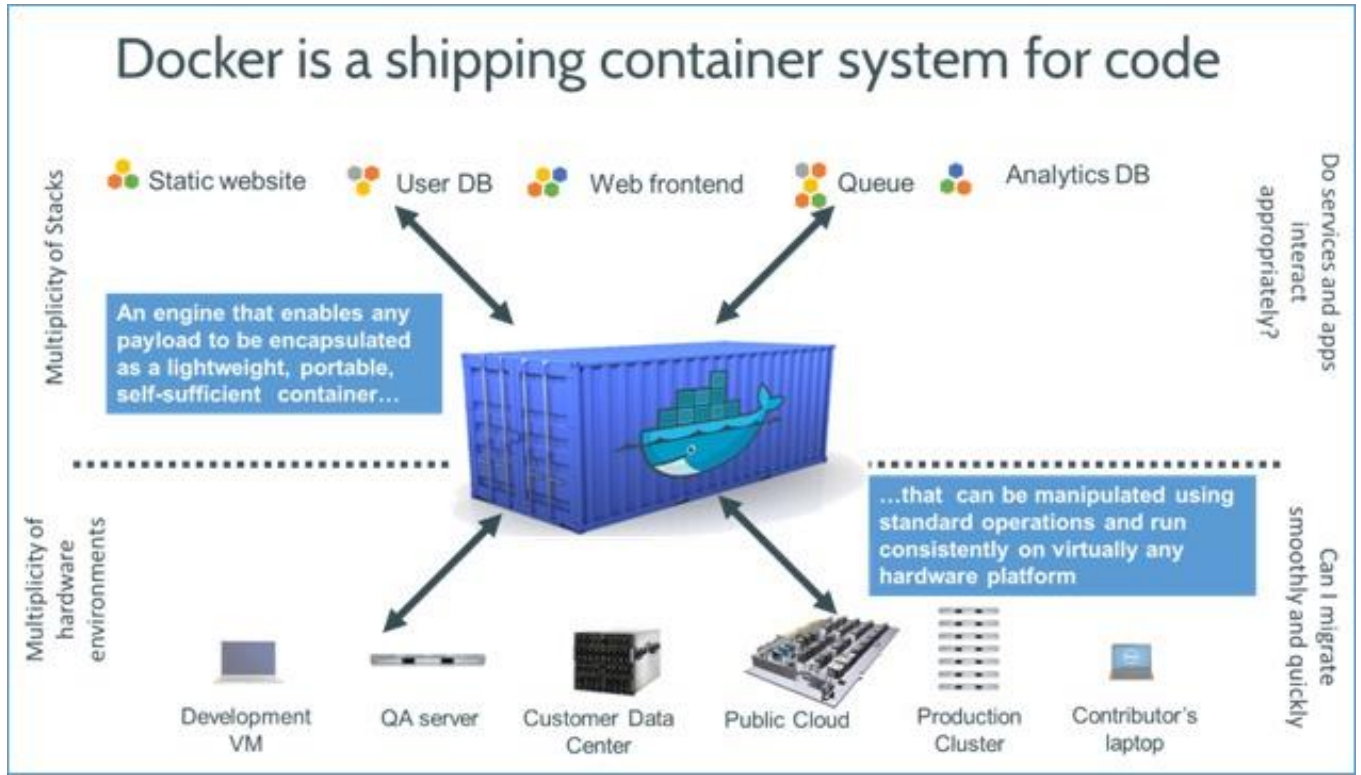
- Linux containers, in short, contain **applications** in a way that keep them isolated from the host system that they run on. Containers allow a developer to **package** up an application with **all of the parts it needs**, such as libraries and other dependencies, and ship it all out as one package. And they are designed to make it easier to provide a consistent experience as developers and system administrators move code from development environments into production in a **fast and replicable** way.

https://opensource.com/resources/what-are-linux-containers


Cargo Transport Pre-1960
Multiplicity of Goods
Do I worry about how goods interact (e.g. coffee beans next to spices)
Multipilicity of methods for transporting/storing
Can I transport quickly and smoothly (e.g. from boat to train to truck)


Multiplicity of Goods
A standard container that is loaded with virtually any goods, and stays sealed until it reaches final delivery.
Do I worry about how goods interact (e.g. coffee beans next to spices)
...in between, can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another
Multiplicity of methods for transporting/storing
Can I transport quickly and smoothly (e.g. from boat to train to truck)


Multiplicity of Stacks
Static website
nginx 1.5 + modsecurity + openssl + bootstrap 2
User DB
postgresql + pgv8 + v8
Queue
Redis + redis-sentinel
Analytics DB
hadoop + hive + thrift + OpenJDK
Background workers
Python 3.0 + celery + pyredis + libcurl + ffmpeg + libopencv + nodejs + phantomjs
Web frontend
Ruby + Rails + sass + Unicorn
API endpoint
Python 2.7 + Flask + pyredis + celery + psycopg + postgresql-client
Do services and apps interact appropriately?
Multiplicity of hardware environments
Development VM
QA server
Customer Data Center
Public Cloud
Production Cluster
Disaster recovery
Production Servers
Contributor's laptop
Can I migrate smoothly and quickly?


Docker is a shipping container system for code
Multiplicity of Stacks
Static website
User DB
Web frontend
Queue
Analytics DB
Do services and apps interact appropriately?
An engine that enables any payload to be encapsulated as a lightweight, portable, self-sufficient container…
…that can be manipulated using standard operations and run consistently on virtually any hardware platform
Multiplicity of hardware environments
Development VM
QA server
Customer Data Center
Public Cloud
Production Cluster
Contributor's laptop
Can I migrate smoothly and quickly

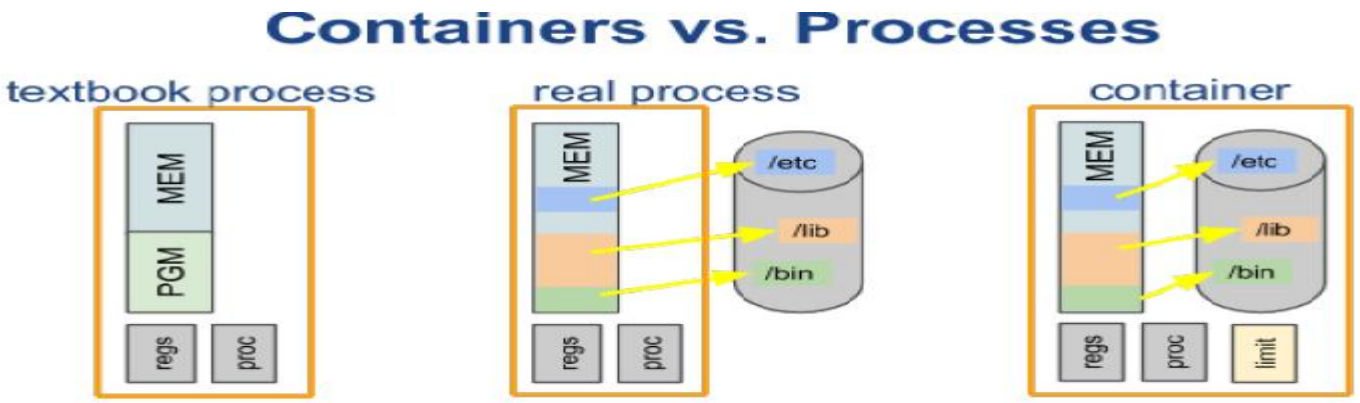
Why use containers?

- Reduces build & deploy times
- Cost control and granularity
- Container technology simplifies cloud portability.Run same application in different clouds
- Container encapsulates applications and defines their interface with the surrounding system

Containers

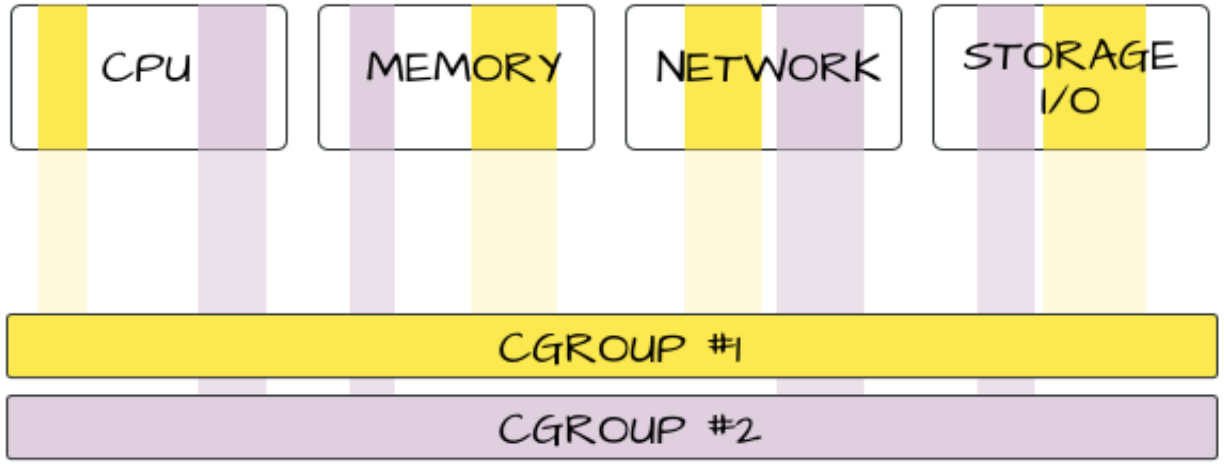
Containers really are processes with their full environment - having its own address space, program, CPU state, and process table entry.

The program text is actually memory mapped from the filesystem into the process address space and often consists of dozens of shared libraries in addition to the program itself, thus all these files are really part of the process.



Cgroups

- cgroups (abbreviated from control groups) is a Linux kernel feature that limits, accounts for, and isolates the resource usage (CPU, memory, disk I/O, network, etc.) of a collection of processes.



Using Containers

- **Creation of a container**

```
# mkdir -p /container/test/lib64 /container/test/bin
```

- **Copying a progam to a container**

```
# cp -l $(ldd /bin/bash | egrep -o "/lib64/.* ") /container/test/lib64/
# cp /bin/bash /container/test/bin/
```

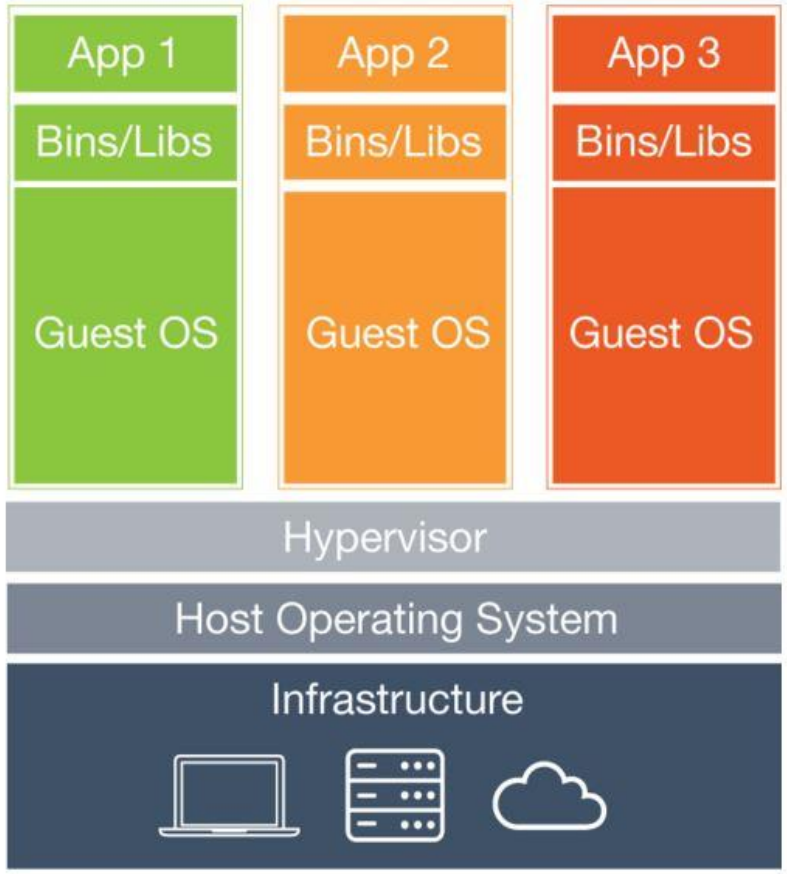
- **Accessing the container**

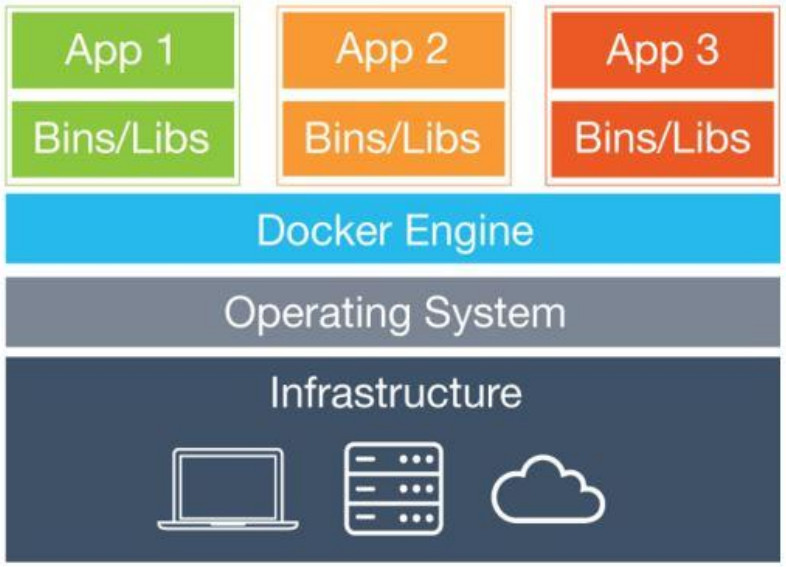
```
# chroot /container/test/
# export PATH=/bin
```

- **Container with network resource segregation**

```
# ip netns exec net-meu-container \
   cgexec -g memory,cpuset:net-meu-container \
    chroot /container/meu-container/ /bin/bash
```

Virtual Machines x Containers





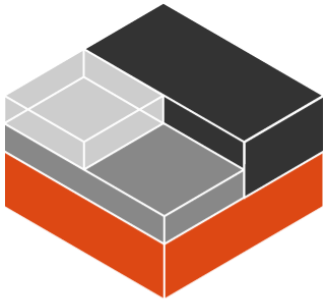
Virtual Machines x Containers

- Do you need to run the maximum amount of particular applications on a minimum of servers? If that's you, then you want to use containers -- keeping in mind that you're going to need to have a close eye on your systems running containers until container security is locked down.

- If you need to run multiple applications on servers and/or have a wide variety of operating systems you'll want to use VMs. And if security is close to job number one for your company, then you're also going to want to stay with VMs for now.

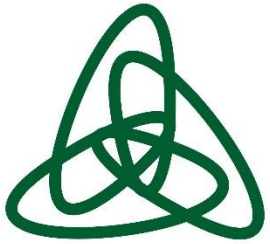
http://www.itworld.com/article/2915530/virtualization/containers-vs-virtual-machines-how-to-tell-which-is-the-right-choice-for-your-enterprise.html

Plataforms and Tools



lxd

Virtuozzo

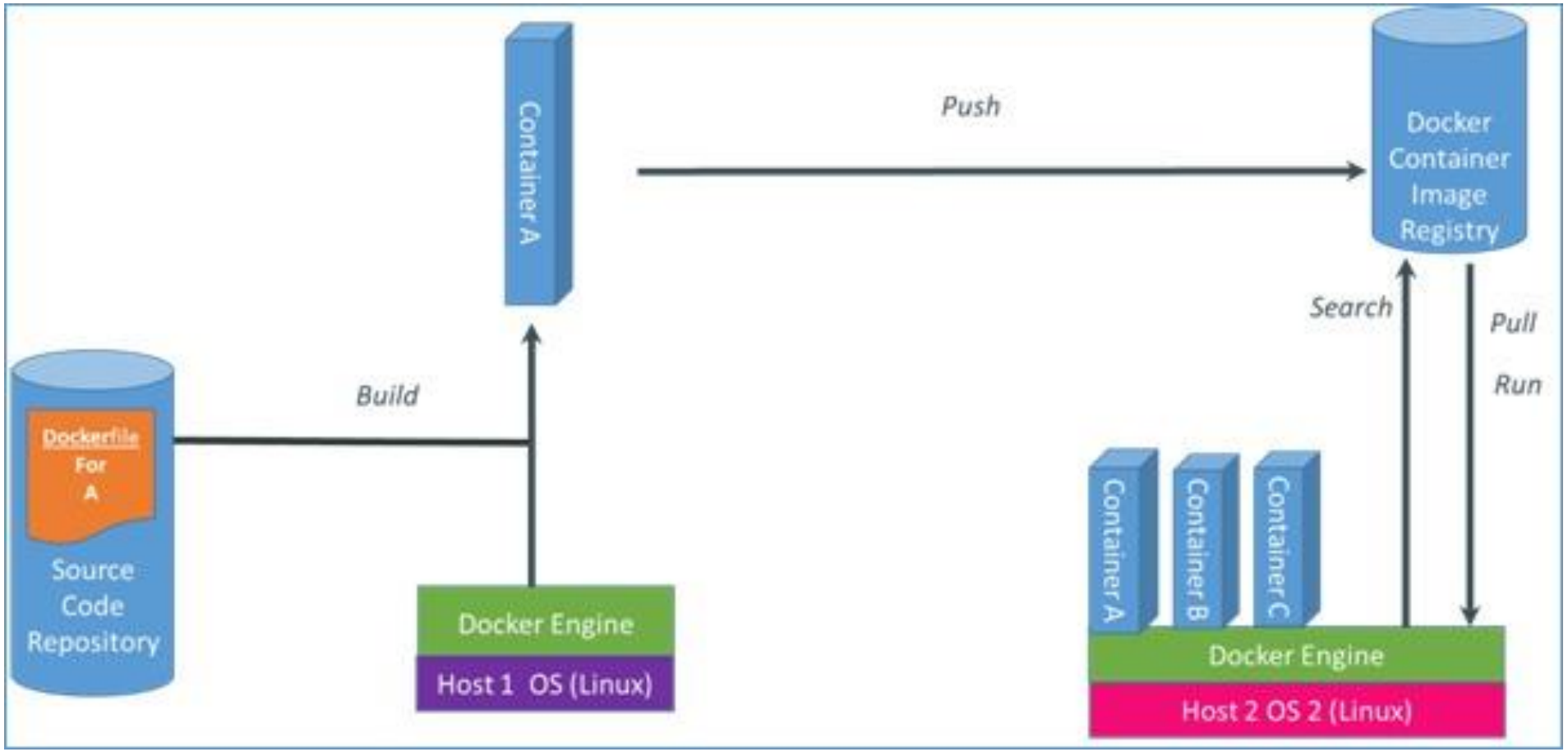


OpenVZ
Linux Containers

Docker Container

Docker *is an open-sourceproject that automates the deployment of applications inside software containers, by providing an additional layer of abstraction and automation of operating system–level virtualizationon Linux*

Docker Image Registry

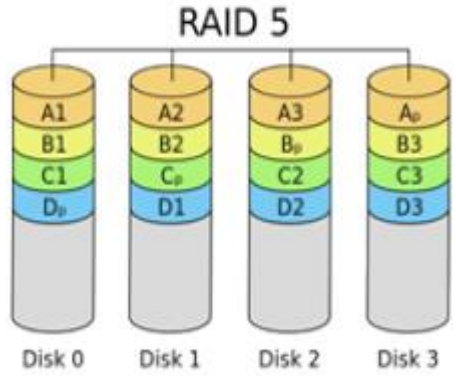


Storage Virtualization

Storage Virtualization

- Storage virtualization means that applications can use storage without any concern for where it resides, and what the technical interface is
- Advantages:
  - Adds or removes storage without any downtime
  - Provides non-disruptive data migration between storage devices
  - Remote storage devices appear local
  - Data is spread over multiple physical disks to improve reliability and performance

Redundant Array of Independent Disks (RAID)



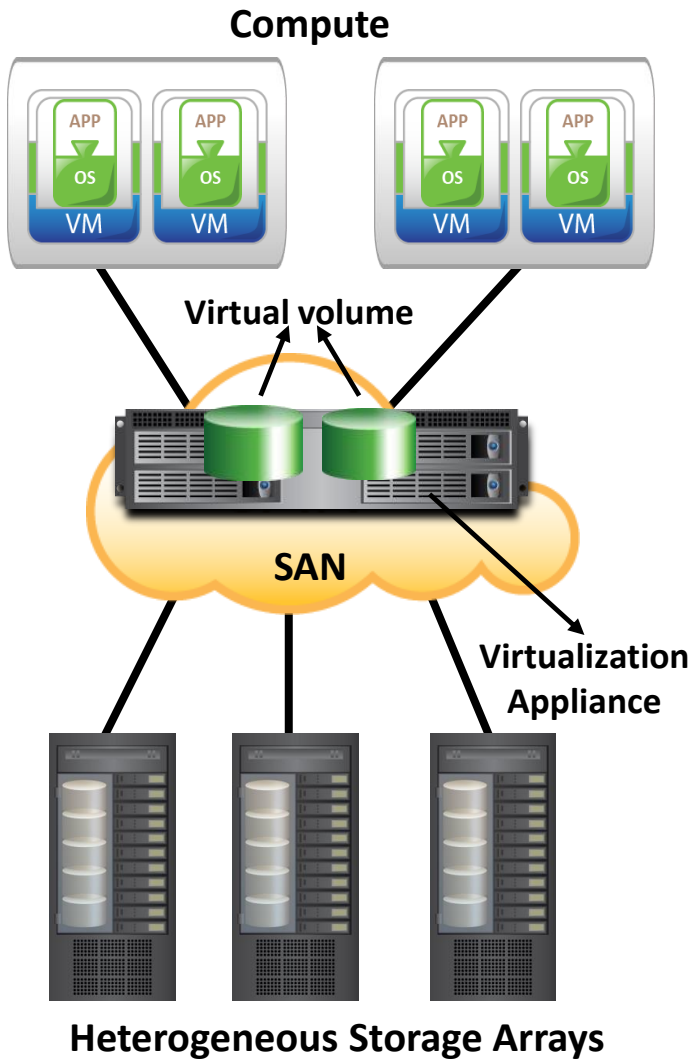
- Utilizes multiple disk drives as a set to provide protection, capacity, and/or performance benefits
- Overcomes limitations of disk drives
- Improves storage system performance by serving I/Os from multiple disks simultaneously

Block-level and File-level Virtualization

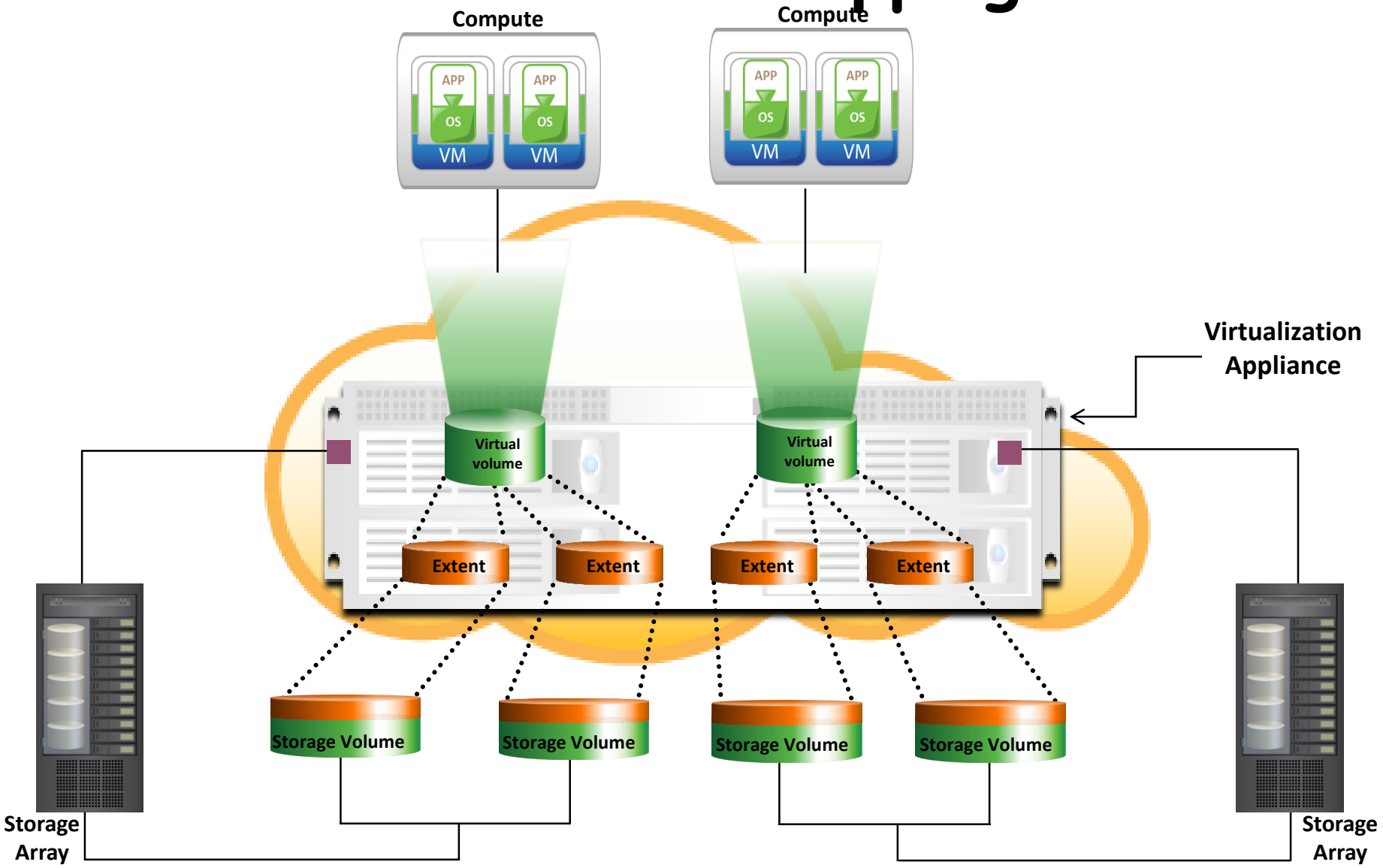
- Network-based virtualization embeds storage virtualization intelligence at the network layer
- Provides ability to
  - Pool heterogeneous storage resources
  - Perform non-disruptive data migration
  - Manage a pool of storage resources from a single management interface
- Network-based storage virtualization is applied at
  - Block-level (SAN)
  - File-level (NAS)

Block-level Storage Virtualization

- Creates an abstraction layer at SAN, between physical storage resources and volumes presented to compute
- Uses virtualization appliance to perform mapping operation
- Makes underlying storage infrastructure transparent to compute
- Enables significant cost and resource optimization

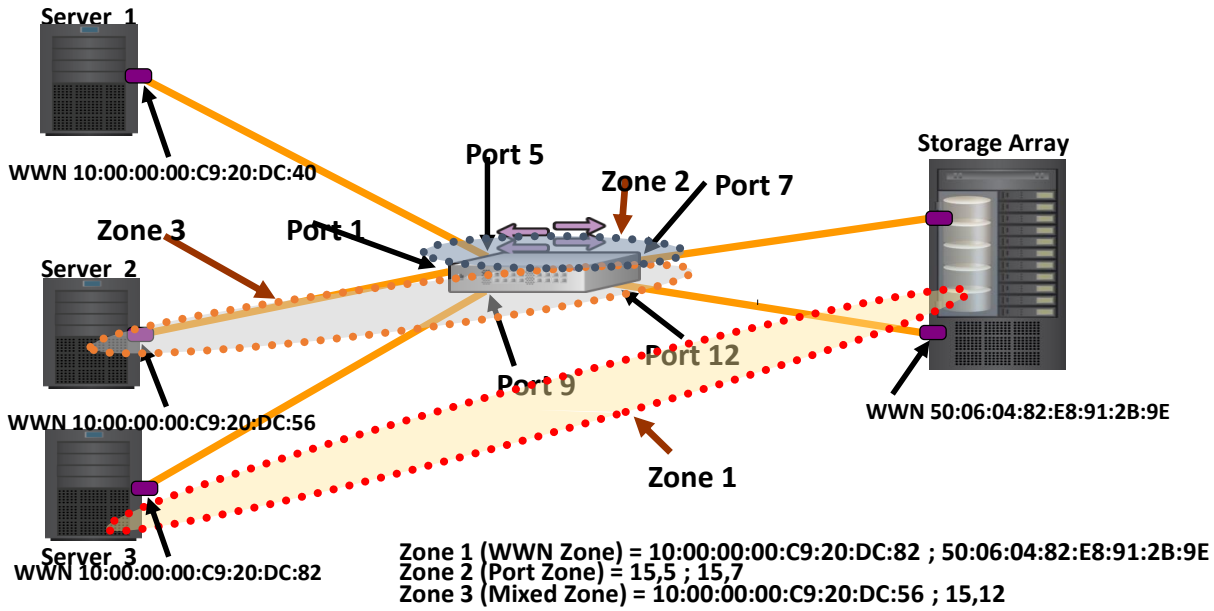


Physical to Virtual Volume Mapping



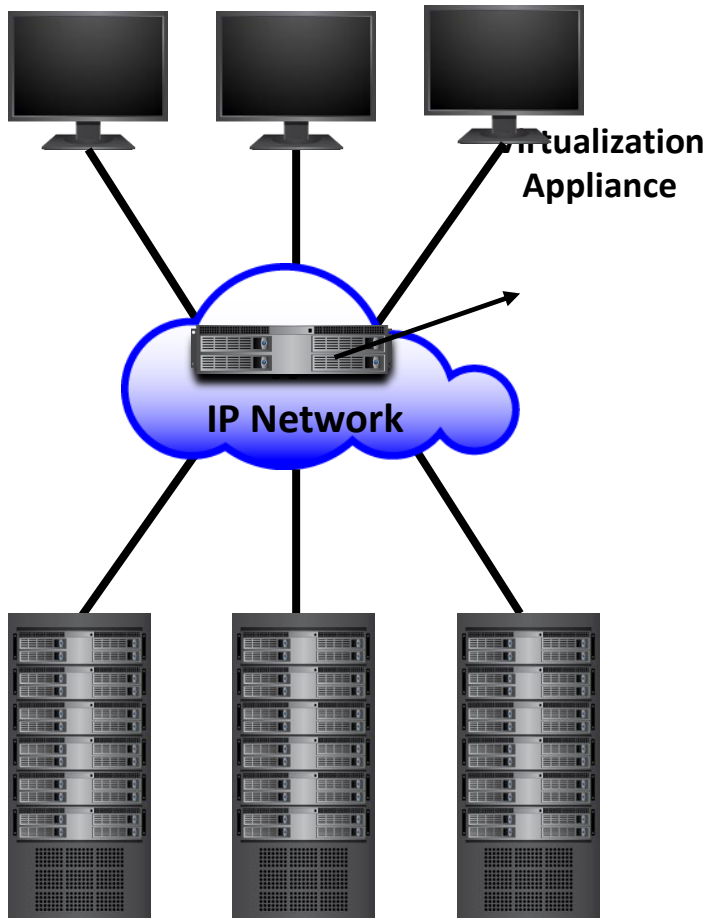
Virtual Storage Area Network (VSAN)

- Zones in a FC SAN provides isolation among tenants. Some switch ports can see only some other switch ports
- Similar to VLANs



File-level Storage Virtualization

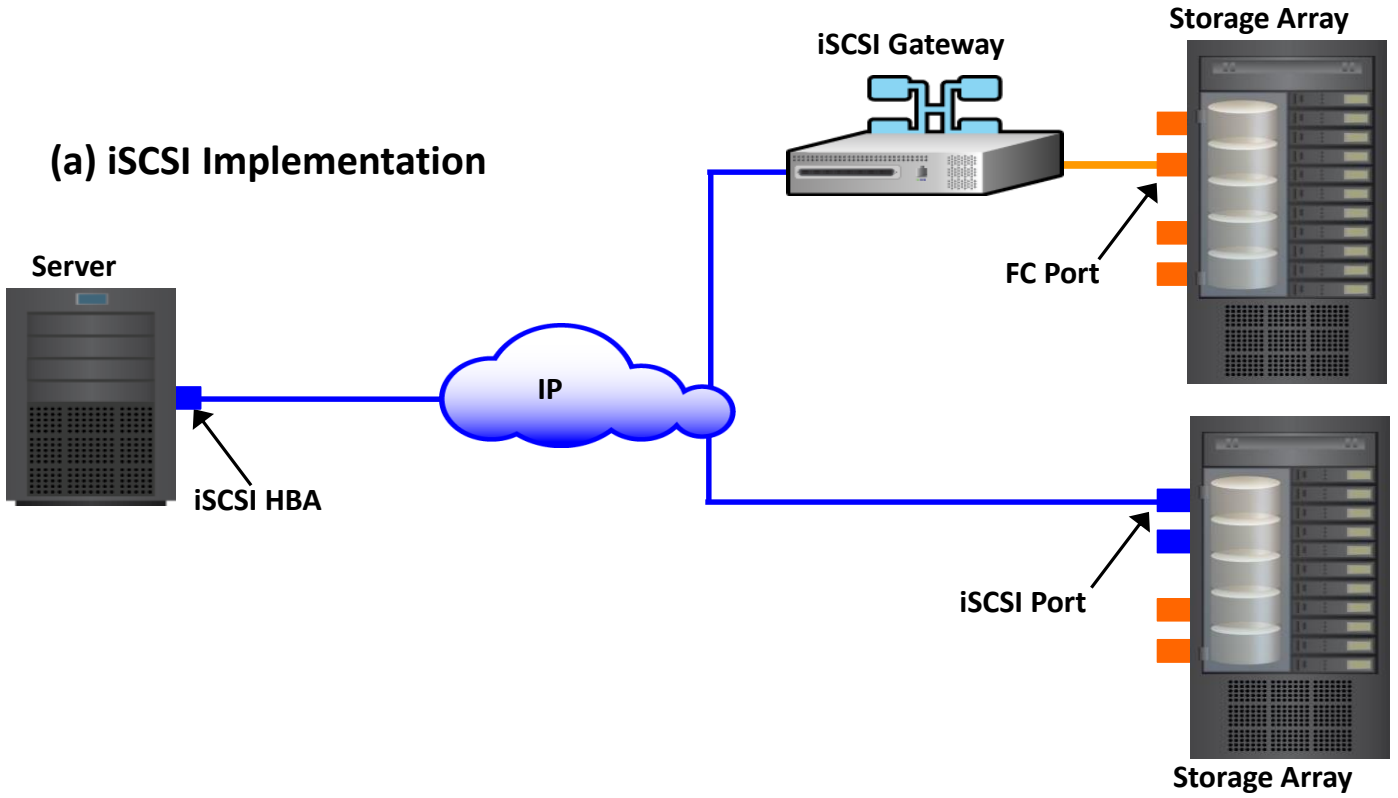
- Provides an abstraction in the NAS/File servers environment
  - Eliminates dependencies between the file and its location
- Enables movement of files between NAS systems without impacting client access
- Provides opportunities to optimize storage utilization
- Implemented using global namespace



iSCSI

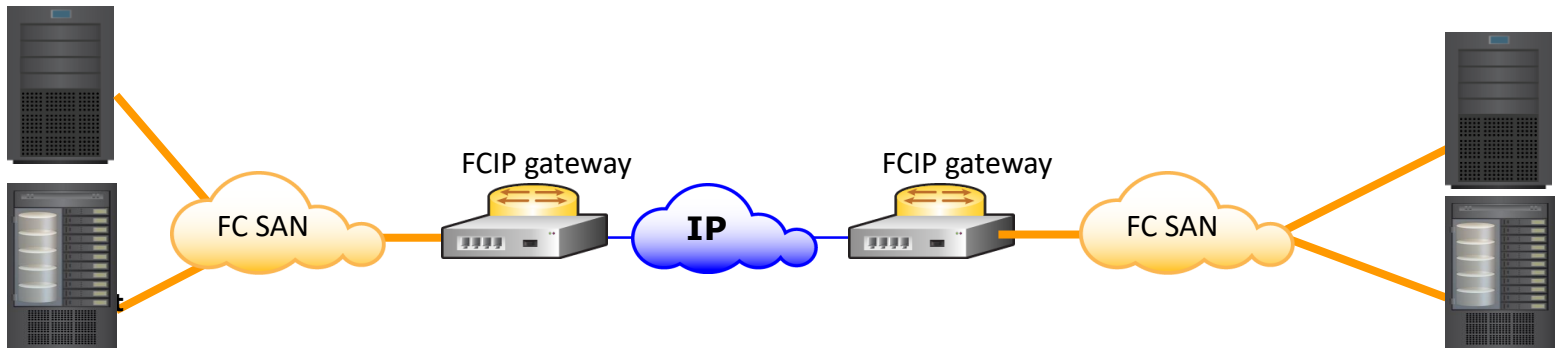
- iSCSI allows two hosts to negotiate and then exchange SCSI commands using Internet Protocol (IP) networks.
- Requires no dedicated cabling
- often seen as a low-cost alternative to Fibre Channel, which requires dedicated infrastructure
- performance of an iSCSI SAN deployment can be severely degraded if not operated on a dedicated network or subnet (LAN or VLAN), due to competition for a fixed amount of bandwidth.

iSCSI

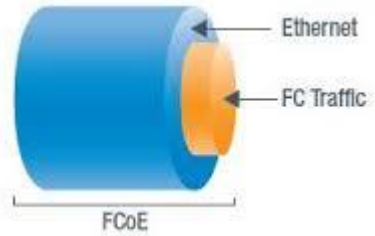


Internet Fiber Channel Protocol (iFCP)

- a gateway-to-gateway network protocol standard, which provides Fibre Channel fabric functionality to fibre channel devices over an IP network (1 Gbit/s, 2 Gbit/s, 4 Gbit/s, 8 Gbit/s, 10 Gbit/s variants).
- Interconnect FC devices using TCP/IP, uses TCP congestion control
- SAN frames are converted to IP packets at the source and sent to the destination

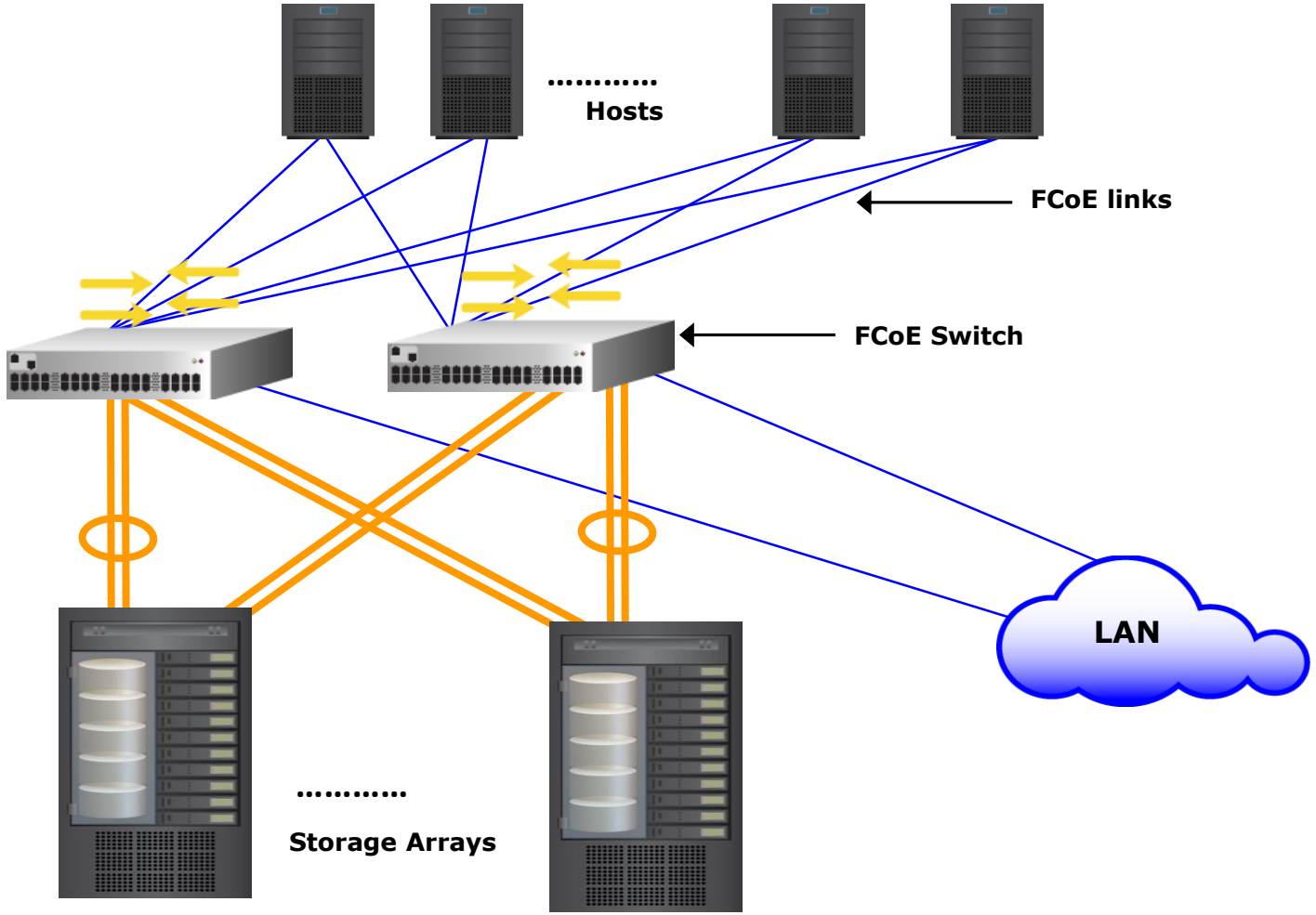


Fibre Channel over Ethernet (FCoE)



- Encapsulates Fibre Channel frames for transport over Enhanced Ethernet networks
- Enables the consolidation of SAN traffic and Ethernet traffic onto a common 10 Gigabit Ethernet infrastructure
- Required mapping between FCIDs and Ethernet MAC addresses

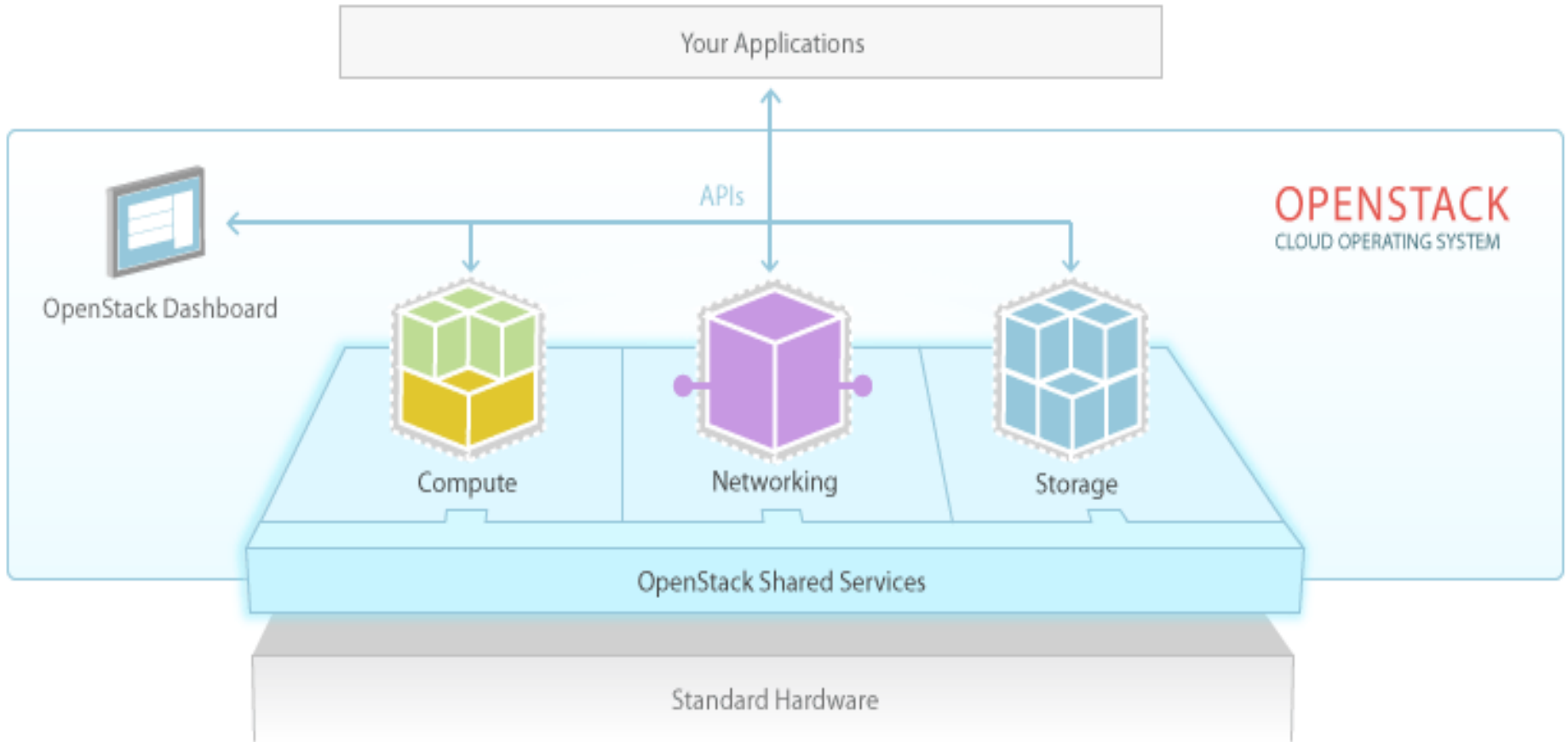
Fibre Channel over Ethernet (FCoE)



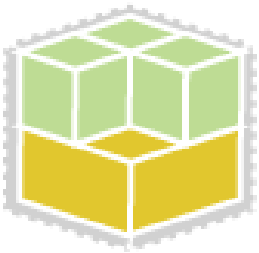
OpenStack

- OpenStack is a cloud operating system that controls large pools of compute, storage, and networking resources throughout a datacenter, all managed through a dashboard that gives administrators control while empowering their users to provision resources through a web interface.

https://www.openstack.org/

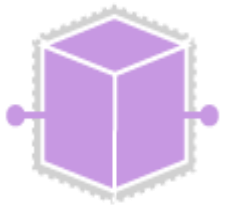

Your Applications
APIs
OPENSTACK
CLOUD OPERATING SYSTEM
OpenStack Dashboard
Compute
Networking
Storage
OpenStack Shared Services
Standard Hardware

OpenStack Compute



- Manage virtualized commodity server resources
- API with rate limiting and authentication
- Distributed and asynchronous architecture
- Virtual Machine (VM) image management
- Live VM management
- Role Based Access Control (RBAC)
- Store and Manage files programmatically via API
- Security Groups
- VM Image Caching on compute nodes
- Dashboard with fully integrated support for self-service provisioning

OpenStack Networking



- OpenStack Networking is a pluggable, scalable and API-driven system for managing networks and IP addresses. Like other aspects of the cloud operating system, it can be used by administrators and users to increase the value of existing datacenter assets. OpenStack Networking ensures the network will not be the bottleneck or limiting factor in a cloud deployment and gives users real self service, even over their network configurations.

OpenStack Networking

- OpenStack provides flexible networking models to suit the needs of different applications or user groups. Standard models include flat networks or VLANs for separation of servers and traffic.
- OpenStack Networking manages IP addresses, allowing for dedicated static IPs or DHCP. Floating IPs allow traffic to be dynamically rerouted to any of your compute resources, which allows you to redirect traffic during maintenance or in the case of failure.
- Users can create their own networks, control traffic and connect servers and devices to one or more networks.
- The pluggable backend architecture lets users take advantage of commodity gear or advanced networking services from supported vendors.
- Administrators can take advantage of software-defined networking (SDN) technology like OpenFlow to allow for high levels of multi-tenancy and massive scale.
- OpenStack Networking has an extension framework allowing additional network services, such as intrusion detection systems (IDS), load balancing, firewalls and virtual private networks (VPN) to be deployed and managed.

OpenStack Storage

- Object Storage is ideal for cost effective, scale-out storage. It provides a fully distributed, API-accessible storage platform that can be integrated directly into applications or used for backup, archiving and data retention. Block Storage allows block devices to be exposed and connected to compute instances for expanded storage, better performance and integration with enterprise storage platforms

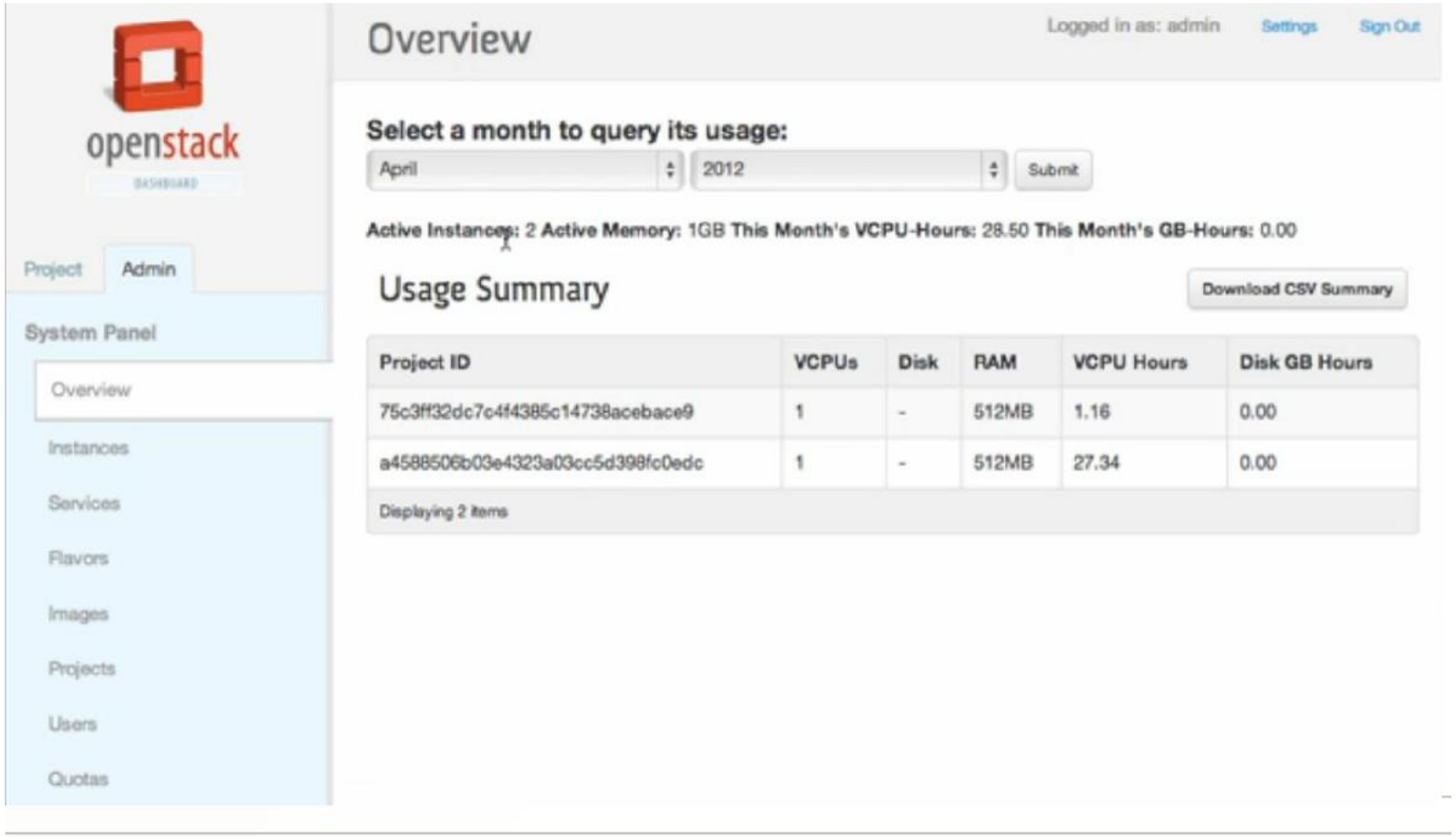
OpenStack Storage

- OpenStack provides redundant, scalable object storage using clusters of standardized servers capable of storing petabytes of data
- Object Storage is not a traditional file system, but rather a distributed storage system for static data such as virtual machine images, photo storage, email storage, backups and archives. Having no central "brain" or master point of control provides greater scalability, redundancy and durability.
- Objects and files are written to multiple disk drives spread throughout servers in the data center, with the OpenStack software responsible for ensuring data replication and integrity across the cluster.
- Storage clusters scale horizontally simply by adding new servers. Should a server or hard drive fail, OpenStack replicates its content from other active nodes to new locations in the cluster. Because OpenStack uses software logic to ensure data replication and distribution across different devices, inexpensive commodity hard drives and servers can be used in lieu of more expensive equipment.

OpenStack Storage

- OpenStack provides persistent block level storage devices for use with OpenStack compute instances.
- The block storage system manages the creation, attaching and detaching of the block devices to servers. Block storage volumes are fully integrated into OpenStack Compute and the Dashboard allowing for cloud users to manage their own storage needs.
- In addition to using simple Linux server storage, it has unified storage support for numerous storage platforms including Ceph, NetApp, Nexenta, SolidFire, and Zadara.
- Block storage is appropriate for performance sensitive scenarios such as database storage, expandable file systems, or providing a server with access to raw block level storage.
- Snapshot management provides powerful functionality for backing up data stored on block storage volumes. Snapshots can be restored or used to create a new block storage volume.

OpenStack Dashboard



Community with Broad Participation

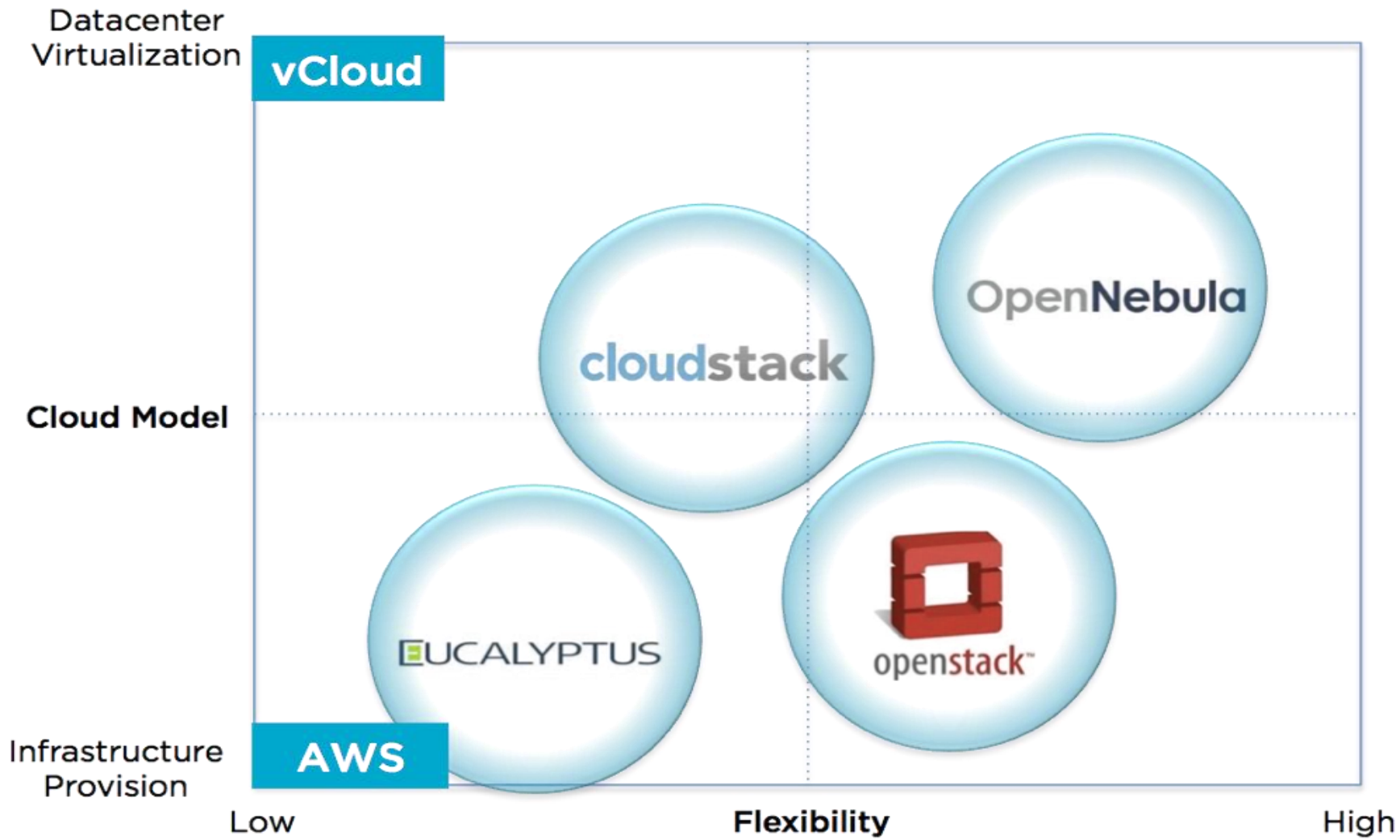
Apache CloudStack

- Apache CloudStack is open source software designed to deploy and manage large networks of virtual machines, as a highly available, highly scalable Infrastructure as a Service (IaaS) cloud computing platform.
- CloudStack is a turnkey solution that includes the entire "stack" of features most organizations want with an IaaS cloud: compute orchestration, Network-as-a-Service, user and account management, a full and open native API, resource accounting, and a first-class User Interface (UI).
- CloudStack currently supports the most popular hypervisors: VMware, KVM, XenServer, Xen Cloud Platform (XCP) and Hyper-V.

CloudStack

- Apache CloudStack is a Java-based project that provides a management server and agents (if needed) for hypervisor hosts so that you can run an IaaS cloud. Some, but not all, of the features and functionality provided by CloudStack:
- Works with hosts running XenServer/XCP, KVM, Hyper-V, and/or VMware ESXi with vSphere
- Provides a friendly Web-based UI for managing the cloud
- Provides a native API
- May provide an Amazon S3/EC2 compatible API (optional)
- Manages storage for instances running on the hypervisors (primary storage) as well as templates, snapshots, and ISO images (secondary storage)
- Orchestrates network services from the data link layer (L2) to some application layer (L7) services, such as DHCP, NAT, firewall, VPN, and so on
- Accounting of network, compute, and storage resources
- Multi-tenancy/account separation
- User management

CloudStack Users


Datacenter Virtualization
vCloud
cloudstack
OpenNebula
Cloud Model
EUCALYPTUS
openstack
Infrastructure Provision
AWS
Low
Flexibility
High