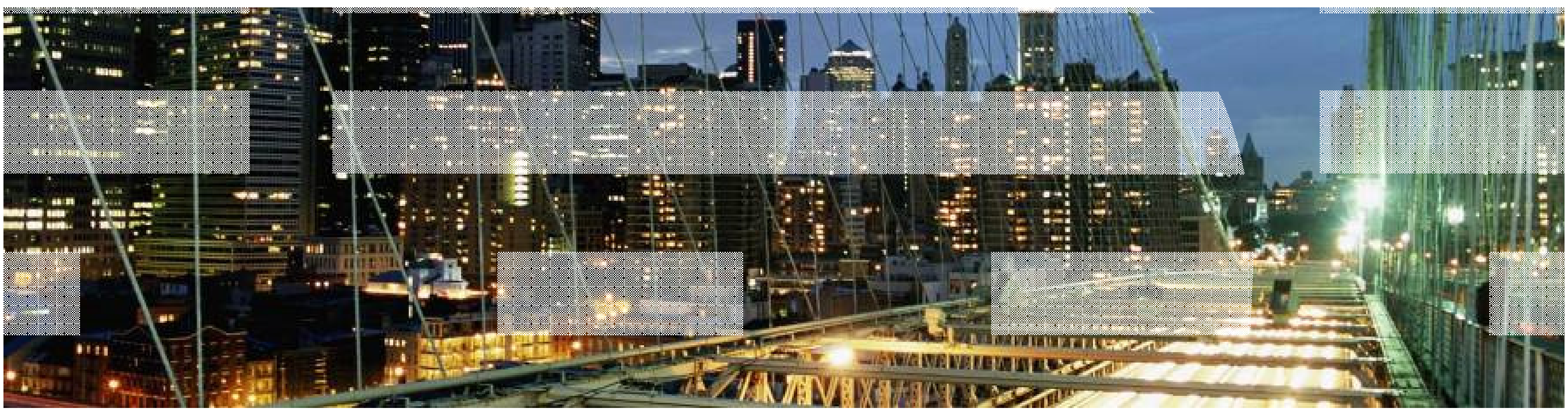
Virtualisierungstechniken

Arwed Tschoeke – Systems Architect
Dr. Manfred Gnirss – IT Specialist



Trademarks

The following are trademarks of the International Business Machines Corporation in the United States, other countries, or both.

AIX*
APPN*
BladeCenter*
CICS*
DB2*
DataPower*
DS8000*
e business(logo)*
ESCON*
eServer
FICON*
GDPS*
HiperSockets
IBM*
IBM (logo)*
IMS
InfiniBand*
Multiprise*
OS/2*
Parallel Sysplex*
Power*
POWER*
POWER7
Power Architecture*
PowerVM
PR/SM
Resource Link*
Redbooks*
REXX
RMF
S/390*
Sysplex Timer*
System/390*
System p*
System x*
z Systems*
z Systems*
z Systems9*
z Systems10*
VSE/ESA
WebSphere*
X-Architecture*
z9*
z10
z10 Business Class
z10 BC
z10 EC
z196
z114
EC12
BC12
z13
z/Architecture*
z/OS*
z/VM*
z/VSE
zEnterpriise
zSeries*

The following are trademarks or registered trademarks of other companies.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.
Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
UNIX is a registered trademark of The Open Group in thec United States and other countries.
Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.
Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

* All other products may be trademarks or registered trademarks of their respective companies.

Notes:
Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Notice Regarding Specialty Engines (e.g., zIIPs, zAAPs and IFLs):

Any information contained in this document regarding Specialty Engines ("SEs") and SE eligible workloads provides only general descriptions of the types and portions of workloads that are eligible for execution on Specialty Engines (e.g., zIIPs, zAAPs, and IFLs). IBM authorizes customers to use IBM SE only to execute the processing of Eligible Workloads of specific Programs expressly authorized by IBM as specified in the "Authorized Use Table for IBM Machines" provided at www.ibm.com/systems/support/machine_warranties/machine_code/aut.html ("AUT"). No other workload processing is authorized for execution on an SE.

IBM offers SEs at a lower price than General Processors/Central Processors because customers are authorized to use SEs only to process certain types and/or amounts of workloads as specified by IBM in the AUT.

z Systems - Overview

- Hardware
- Virtualization
- Other Hypervisors in brief
- z Systems – Virtualization
- Container

Linux on z Systems – z/VM

Hardware

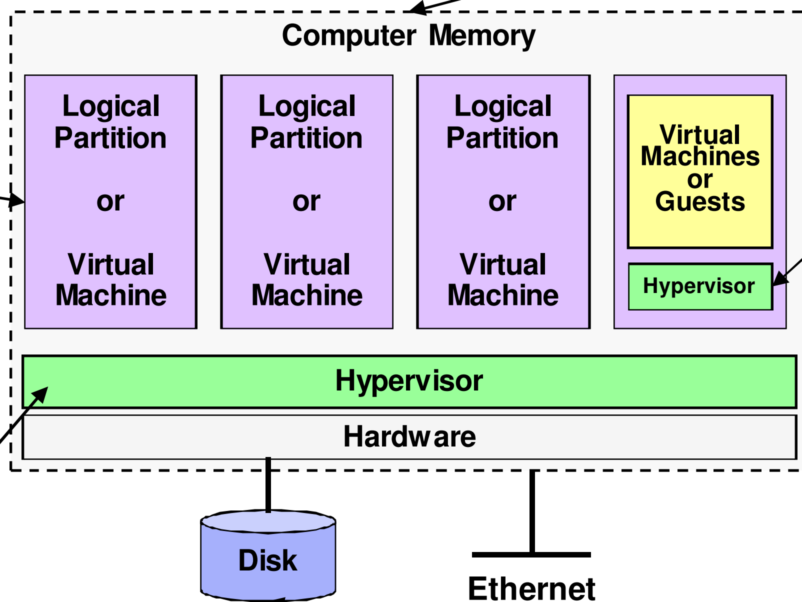
Server Virtualization Terms

Logical Partition

- Also called an **LPAR,** virtual machine, or **VM** or **guest (z/VM,KVM)**
- Runs an operating system such as **z/OS**, **Linux**, **TPF**, **z/VSE**, AIX, IBM i, Windows, Solaris

Memory Virtualization

- Dedicated to an **PR/SM** LPAR
- Shared by guests within **z/VM & KVM**



2nd Level Hypervisor

- Run an Hypervisor inside of an LPAR
- Provides unique features
- Example: **z/VM**

Hypervisor

- "Virtualization" software
- Divides real computing into logical computers or LPARs
- Referred to as "**PR/SM**" on z Systems

I/O Virtualization – Provided by

- Hypervisor (VMware)
- I/O owning LPAR (PowerVM, Xen)
- **Direct hardware virtualization (z)**

Linux on z Systems – z/VM

Virtualization

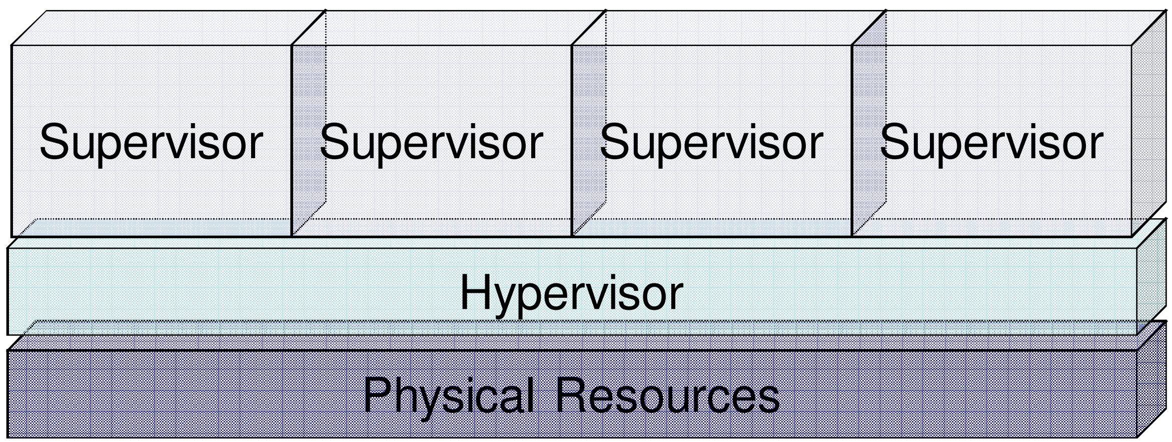
Virtualization

- **Virtualization comprises the abstraction of physical systems to virtual systems**
  - Divison of static relationships between logical system environments (environment of services and applications) and physical systems
  - Two possible directions:
    - Integration of many single physical systems to one logical system
    - Segmentation of one physical system into many logical systems
  - Such „logical systems“ are named as virtual machines
  - A virtual machine is a fully protected and isolated simulation of the underlaying hardware

...altough virtualization comprises both integration and segmentation, this presentation concentrates on segmentation.

Important Terms Concerning Virtualization

- **Supervisor and Hypervisor**
  - Supervisor is another term for an operating system of an virtual machine
    - Controls the virtual machine and its dedicated resources
  - Hypervisor (or Virtual Machine Monitor) is another term for an controller of virtual machines
    - Controls the physical system resources and dedicates them to virtual machines.
    - Controls and handles processes of virtual machines which are critical to physical hardware
    - Isolation of virtual machines
    - Switching (context switching) between virtual machines (e.g., exits, time slicing...)

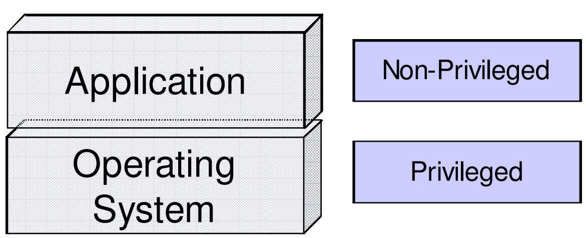


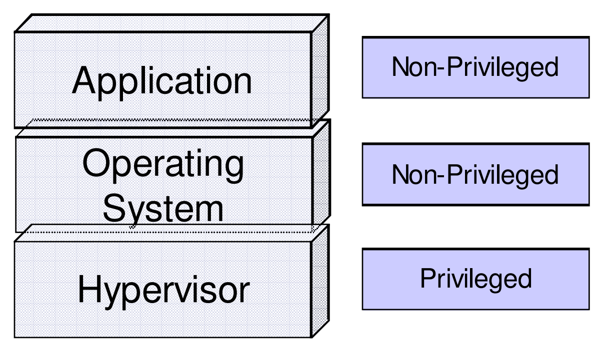
Important Terms Concerning Virtualization

- **Kernel (Privileged) Mode and User Mode**
  - Kernel Mode provides full access to system resources. It is the mode of the operating system which administers and dedicates physical system resources.
  - User Mode provides restricted access to system resources (e.g., applications)

- **Privileged and Non-Privileged instructions**
  - Privileged instructions can only be executed within Kernel Mode

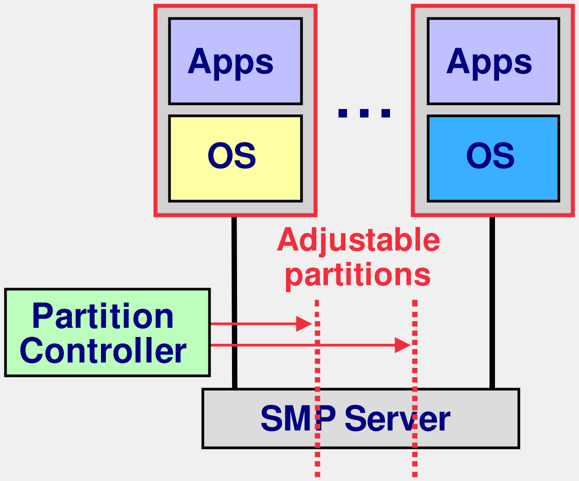
- **Sensitive and Non-Sensitive instructions**
  - Sensitive instructions invoke critical hardware areas





Server Virtualization Approaches

Hardware Partitioning

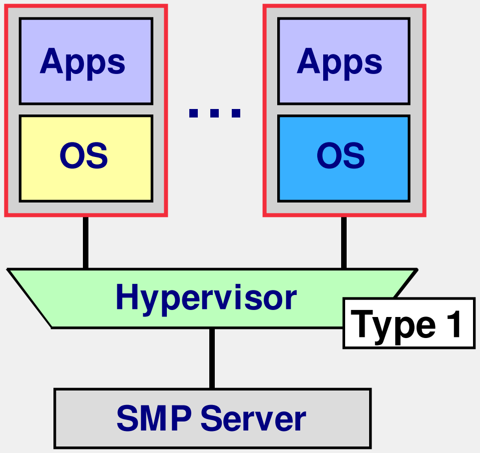


Server is subdivided into fractions each of which can run an OS

Physical partitioning
S/370 SI-to-PP and PP-to-SI, Sun Domains, HP nPars

Logical partitioning
pSeries LPAR, HP (PA) vPars

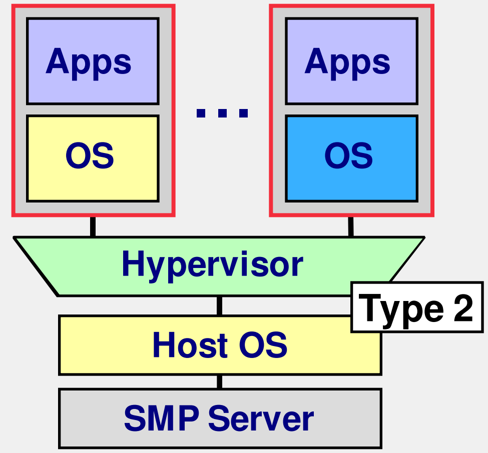
Bare Metal Hypervisor



Hypervisor provides fine-grained timesharing of all resources

Hypervisor software/firmware runs directly on server
z Systems PR/SM and z/VM
POWER Hypervisor
VMware ESXi
Xen Hypervisor, Hyper-V

Hosted Hypervisor



Hypervisor uses OS services to do timesharing of all resources

Hypervisor software runs on a host operating system
VirtualBox

HP Integrity VM
User Mode Linux

- **Hardware partitioning subdivides a server into fractions, each of which can run an OS**
- **Hypervisors use a thin layer of code to achieve fine-grained, dynamic resource sharing**
- **Type 1 hypervisors with high efficiency and availability will become dominant for servers**
- **Type 2 hypervisors will be mainly for clients where host OS integration is desirable**

Type 1 Hypervisor

Goldberg, R. Architectural Principles for Virtual Computer Systems. PhD thesis, National Technical Information Service, February 1973.

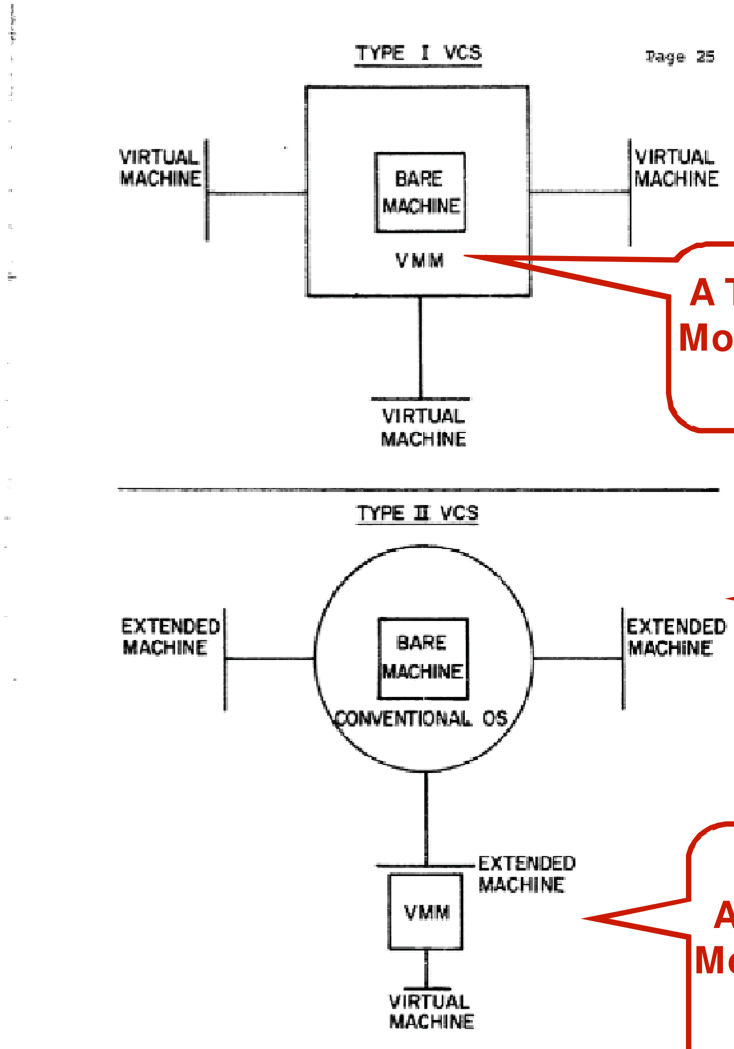


FIGURE 2-4 TYPE I vs TYPE II VCS

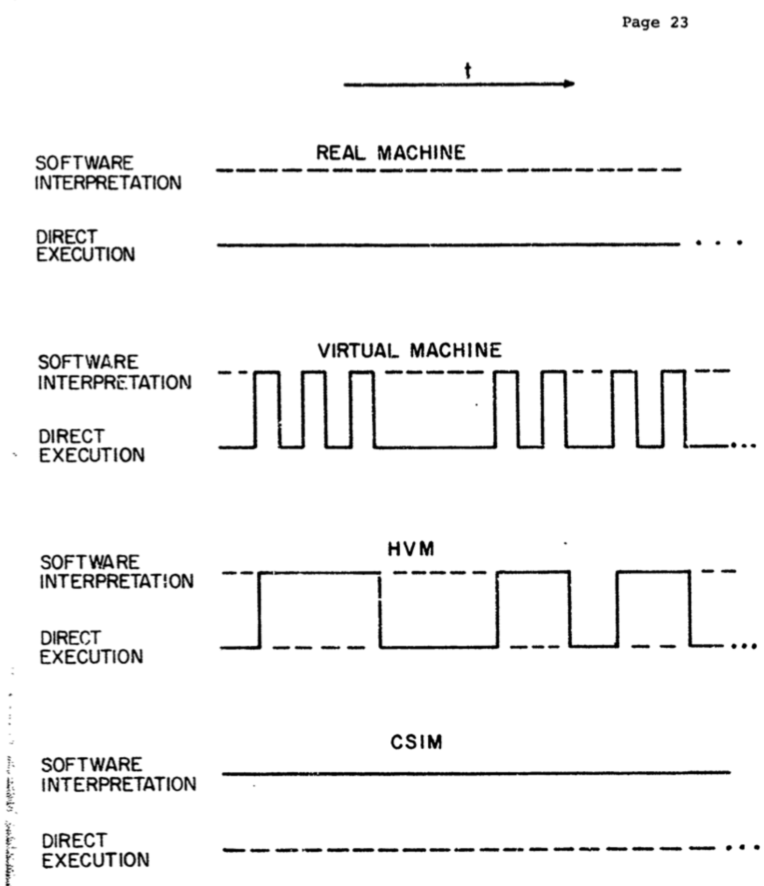
A Type 1 Virtual Machine Monitor runs as part of the Kernel

"Extended Machine" is the environment running a time-shared program. The EM evolved into a modern OS Process.

A Type II Virtual Machine Monitor runs as a standard OS Process

Software Interpretation

Goldberg, R. Architectural Principles for Virtual Computer Systems. PhD thesis, National Technical Information Service, February 1973.



VIRTUAL MACHINE vs. OTHER CONSTRUCTS

FIGURE 2-3

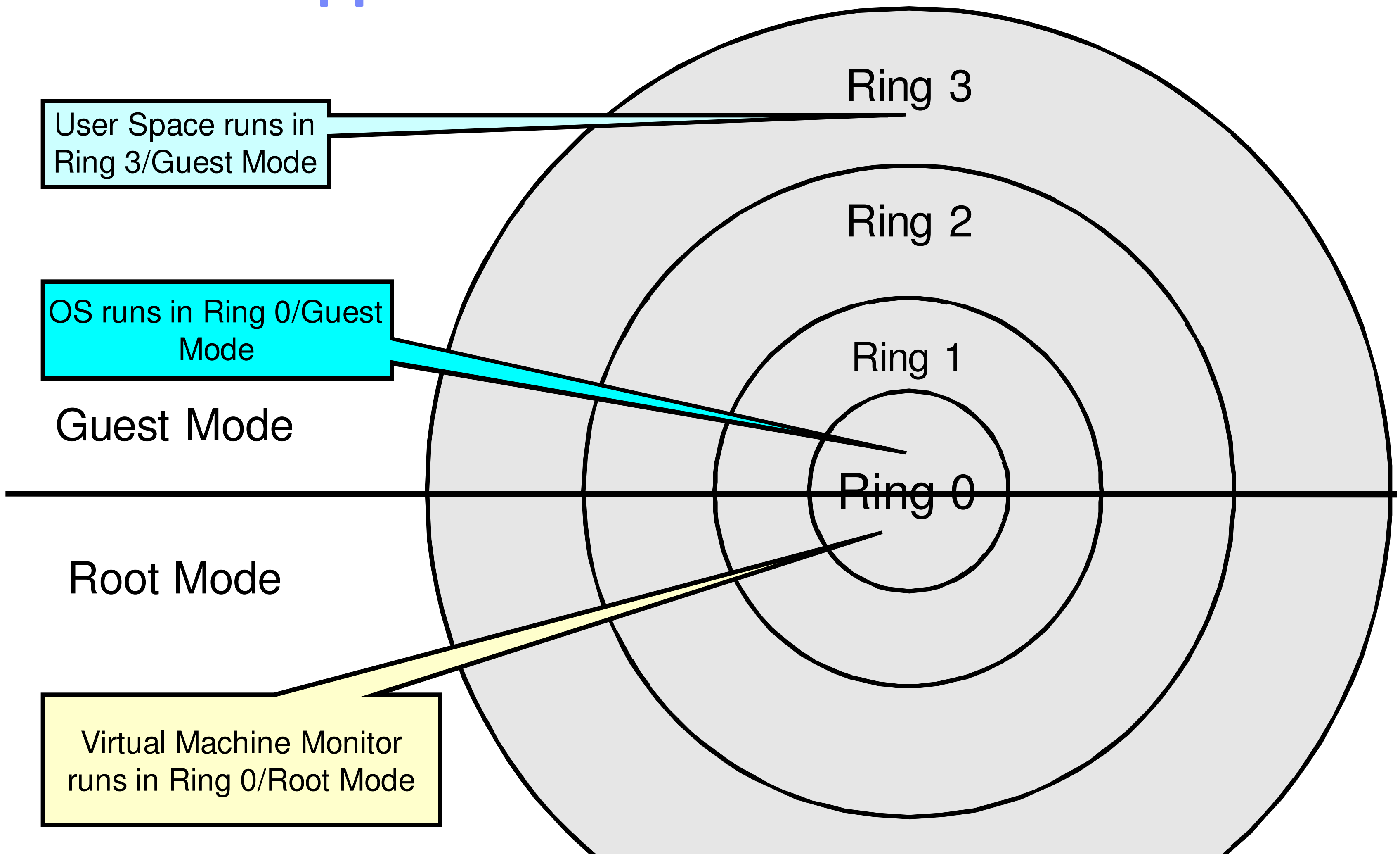
Non-Virtualized

KVM, VMware, Xen, PowerVM, z/VM

VMware Pre-VMX

Qemu

Hardware Support



Isolation of guests from each other and from VMM is enforced by hardware. Therefore breaks in isolation will be in VMM coding errors or Guest privilege escalations.

Hypervisor Implementation Methods(multiple may apply)

Trap and Emulate

Virt Mach
L
A
ST
PrivOp → Trap → Hypervisor PrivOp emulation code
L
...

- VM runs in user mode
- All privileged instructions cause traps

Examples: CP-67, VM/370
Benefits: Runs unmodified OS
Issues: Substantial overhead

Translate, Trap, and Emulate

Virt Mach
L
A
ST
TrapOp → Trap → Hypervisor PrivOp emulation code
L
...

- VM runs in user mode
- Some IA-32 instructions must be replaced with trap ops

Examples: VMware workstation
Benefits: Runs unmodified, translated OS
Issues: May have some substantial overhead

Hypervisor Calls ("Paravirtualization")

Virt Mach
L
A
ST
Hcall → Call → Hypervisor service
L
...

- VM runs in normal modes
- OS in VM calls hypervisor in case of critical Syscalls

Examples: POWER Hypervisor, Xen&KVM (optional), HP Integrity VM
Benefits: High efficiency *depending of Hypervisor code + eventual HW support*
Issues: OS-Kernel must be modified to issue Hcalls. OS & Hypervisor levels must be in sync

Direct Hardware Virtualization

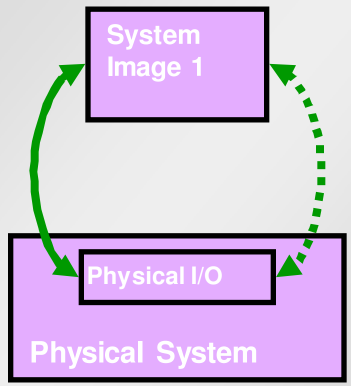
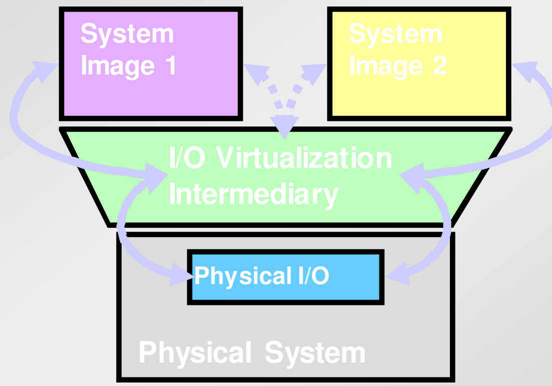
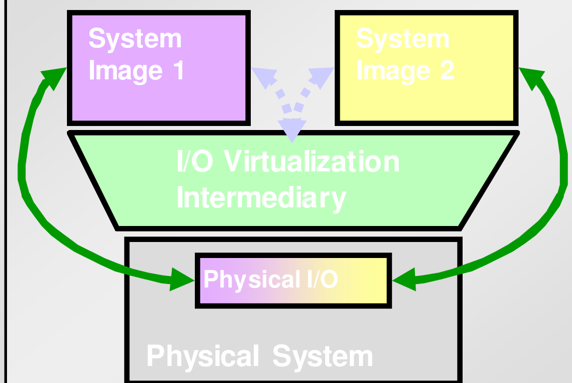
Virt Mach
L
A
ST
PrivOp(*) → Exit → Hypervisor service
L
...

- VM runs in normal modes
- HW does most of the virtualization
- SIE arch – set architecture of VM, provide status, translation & assists)
- Hypervisor provides control

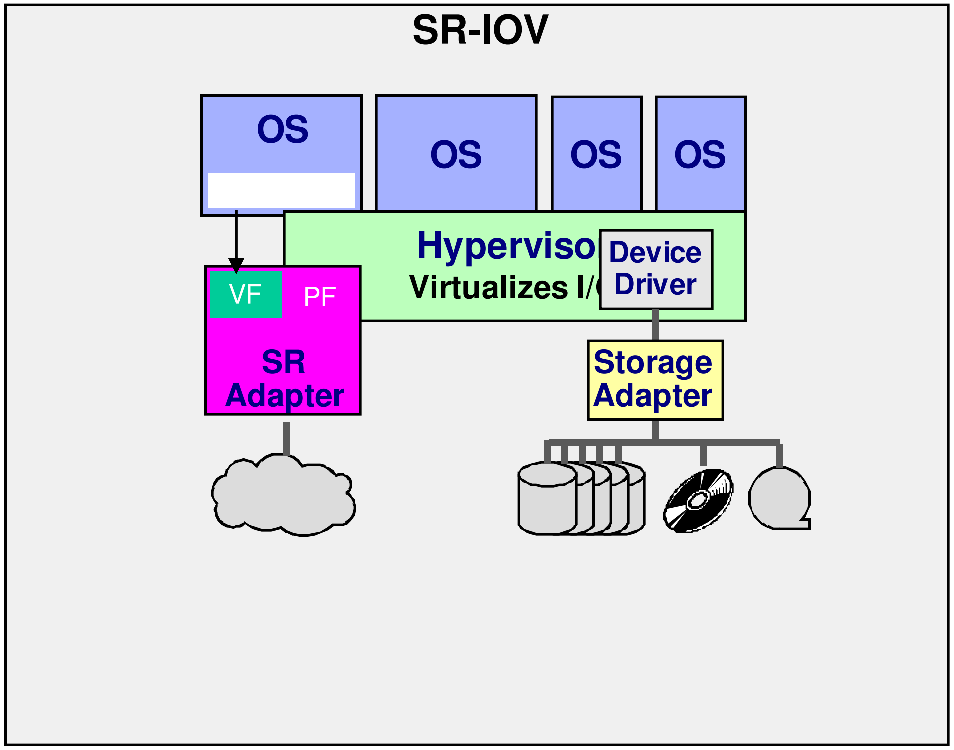
(*) ONLY for some control instructions, executed rather infrequent

Examples: PR/SM, z/VM (also use hypervisor calls) for a some functional enhancements
Benefits: Highest efficiency depending on HW/ucode support. Runs unmodified OS
Issues: Requires HW & ucode support

SR-IOV

	Dedicated Adapter (No Virtualization)	Adapter Shared Through Intermediary	Natively Shared Adapter
Graphic Depiction			
Intermediary Role	None	Virtualizes physical I/O by intervening on configuration and data transfer operations	Manages assignment of Virtual Resources by intervening on configuration operations
Configuration Operation Path	SI direct to Adapter	VI serves as proxy (SI to VI; VI to Adapter)	VI serves as proxy (SI to VI; VI to Adapter)
Data Transfer Operation Path	SI direct to Adapter	VI serves as proxy (SI to VI; VI to Adapter)	SI direct to Adapter

SR-IOV in virtual systems



VF: Virtual function
PF: Physical function

How does it work?
•without SR-IOV the hypervisor creates virtual IO-adapter which impacts on performance
•with SR-IOV this work is done by the adapter itself
•IO-devices have physical functions, each with virtual functions; virtual functions are mapped to a guest system of the hypervisor
•via a special device driver the guest can directly adress the VF of an adapter without the support of the hypervisor

Linux on z Systems – z/VM

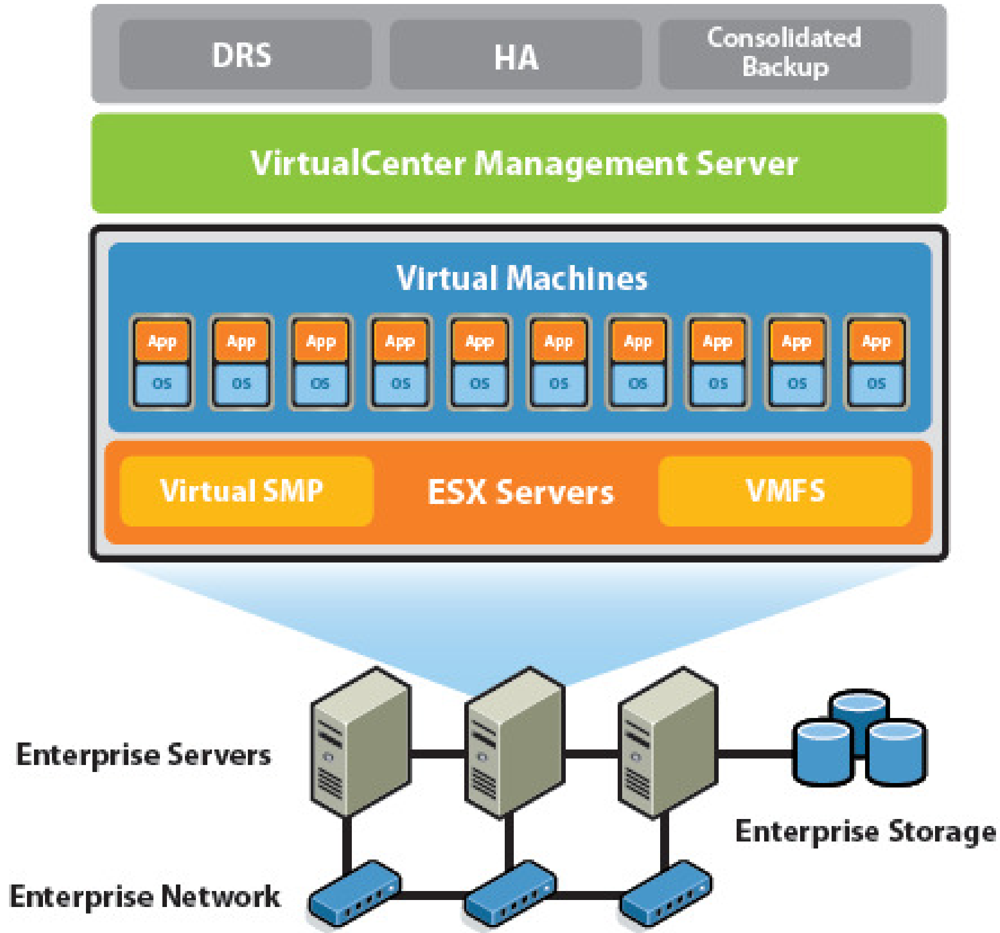
Other Hypervisors in brief

VMware Virtual Infrastructure

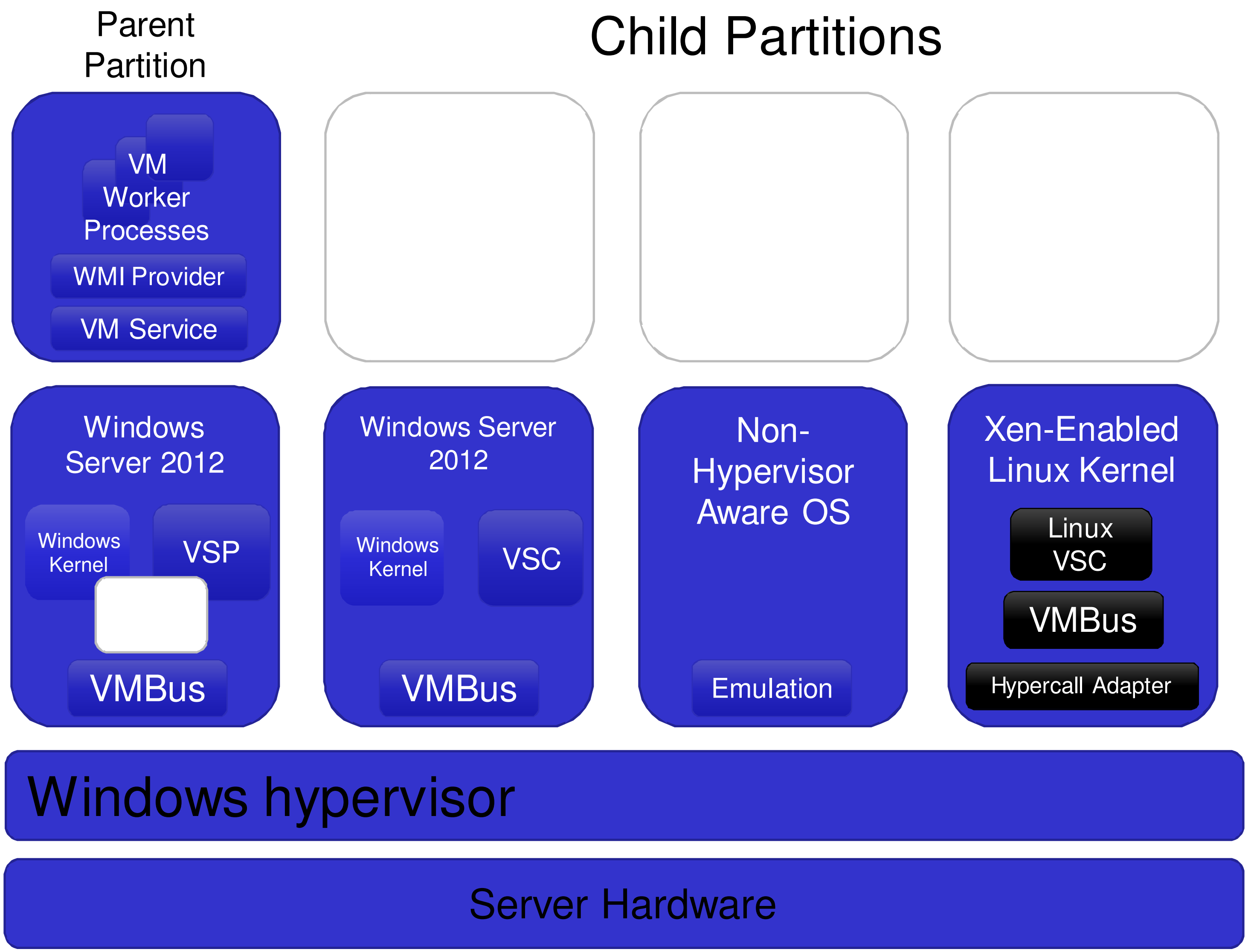
physical:
- 320 logical CPU
- 6 TB RAM

guest:
- 512vms, 4096vCPU
- 64-way Virtual SMP
- 1TB guest memory

Features:
- Unified GUI
- VMotion (guest & storage)
- DRS
- HA
- Update Manager
- DPM
- SRM



Hyper-V - Architecture



physical:

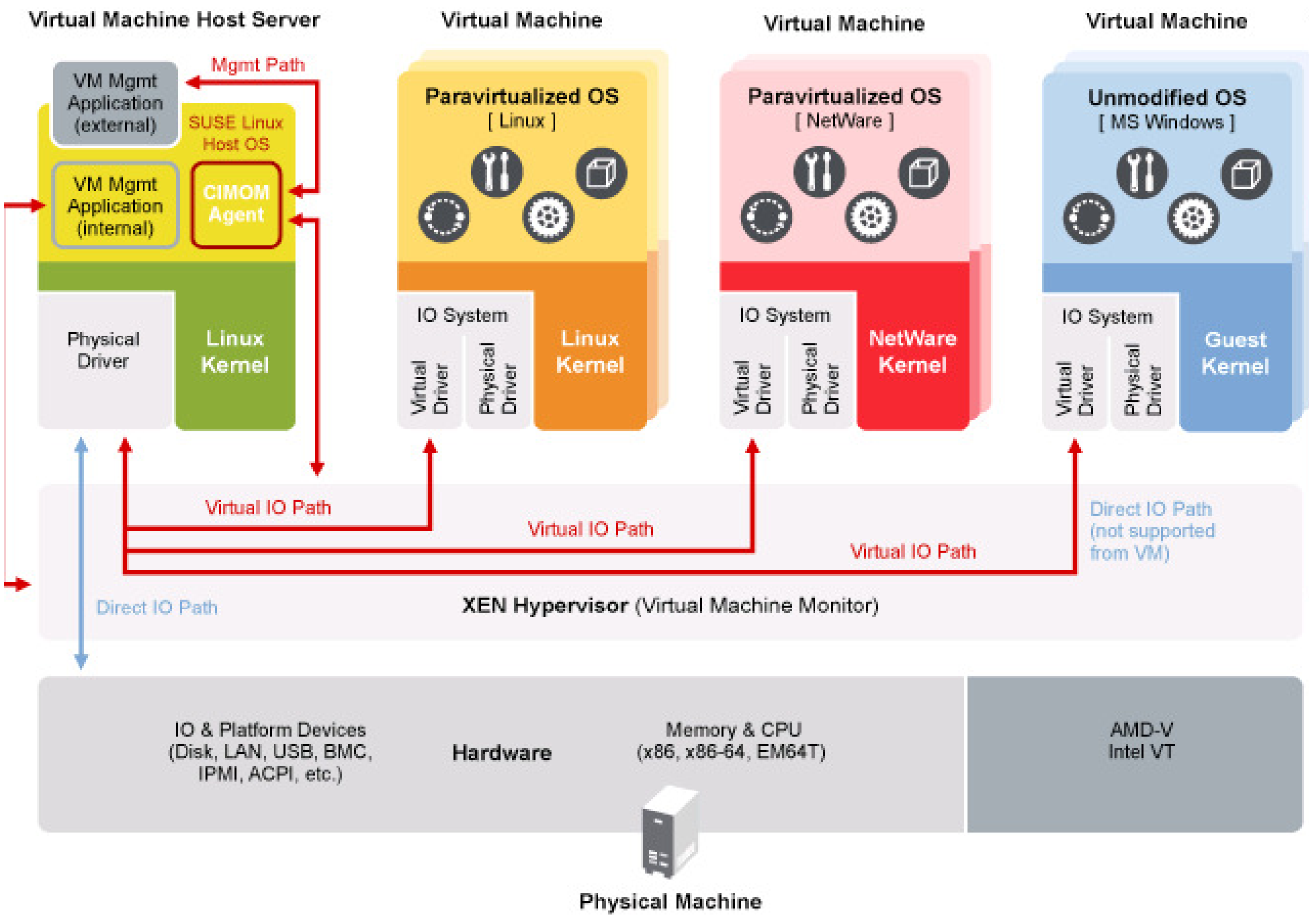
- 320 log. CPU
- 4 TB RAM
- 1024vms, 2048vCPU

guest:

- 64 vCPUs
- 1 TB

Xen

- Primarily
  - paravirtualization
- Full virtualization
  - possible
- Citrix uses Xen as
  - strategic platform
- physical:
  - 160 log. CPU
  - 1TB
  - 500/650* guests
- guests:
  - 16/32* vCPU
  - 192 GB



*** Windows/Linux**

KVM-Architecture

Included in Linux kernel since 2006, maintained by the community, utilizes Linux security

Runs Linux, Windows and other operating system guests, paravirtualized drivers available

Advanced features

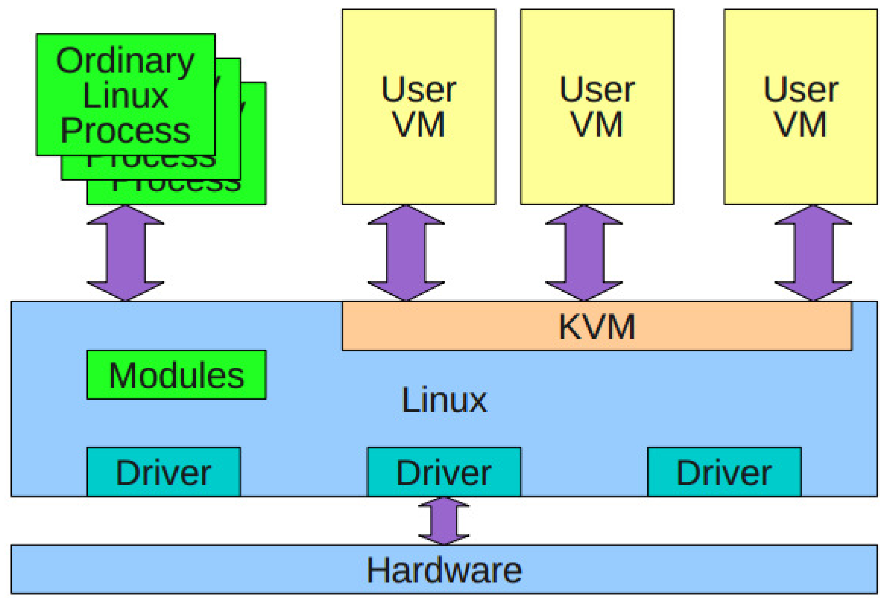
- **Live migration**
- **Memory page sharing**
- **Thin provisioning**
- **PCI Pass-through**

physical:

- 160 log. CPU
- 4 TB
- no limited number of guests

guests:

- 160 vCPU
- 4 TB



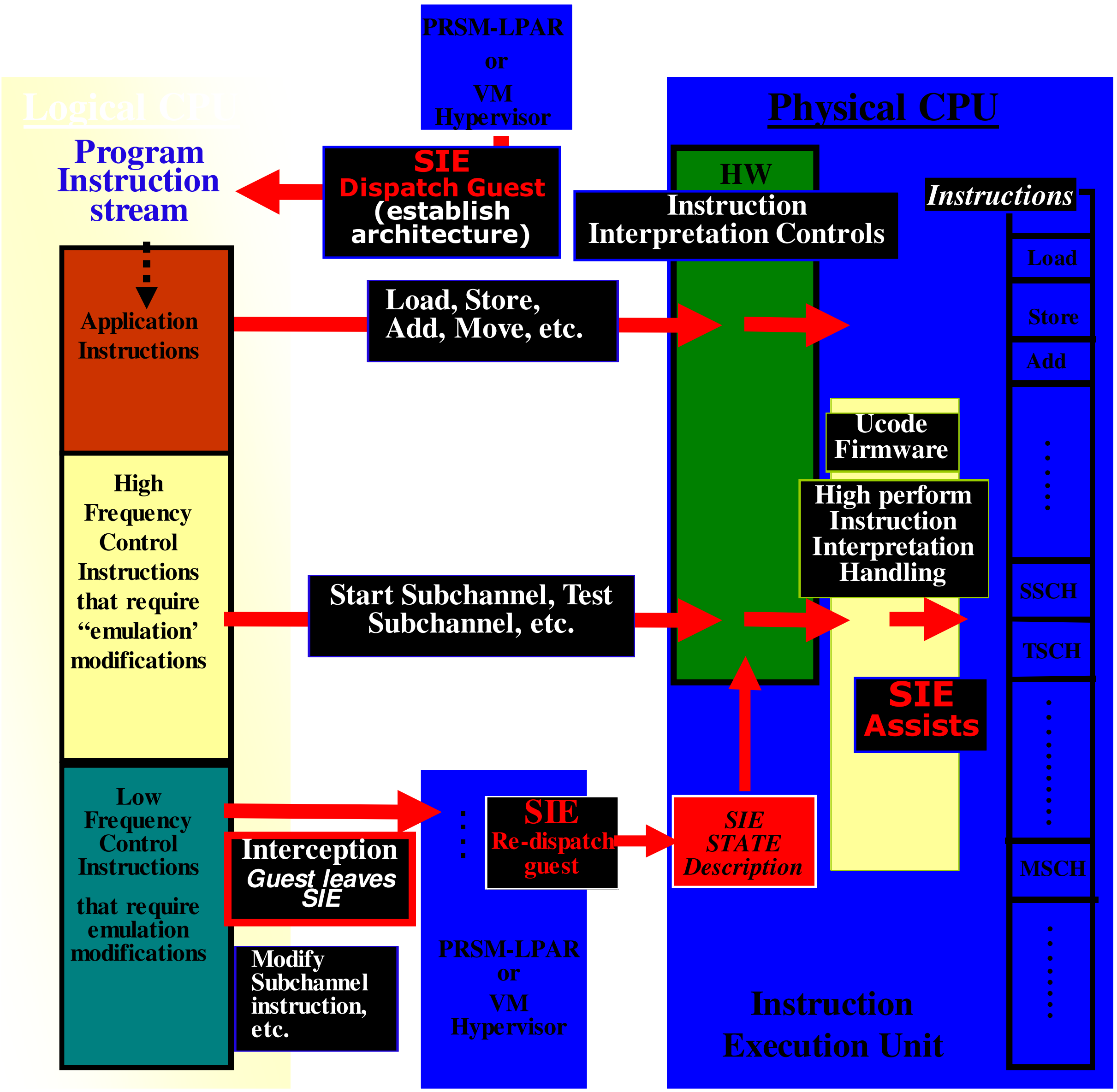
The Market from a Gardner Perspective

Source: Gartner (July 2014)

Linux on z Systems – z/VM

z Systems Virtualization

Basic Direct HW virtualization – *transparent to applications/OS*



z Systems with SIE

(Start Interpretive Instruction Execution)

**** z Systems runs ALWAYS in PR/SM-LPAR mode under SIE***

**** LPAR is the "only" game in town, meaning performance items and other functionalities is developed accordingly***

zVM invokes SIE to run VM's (SIE under SIE)

**** Efficient for performance - and new version of OS and Hypervisor***

Positioning of z Systems & Intel/AMD

- ***z Systems:***

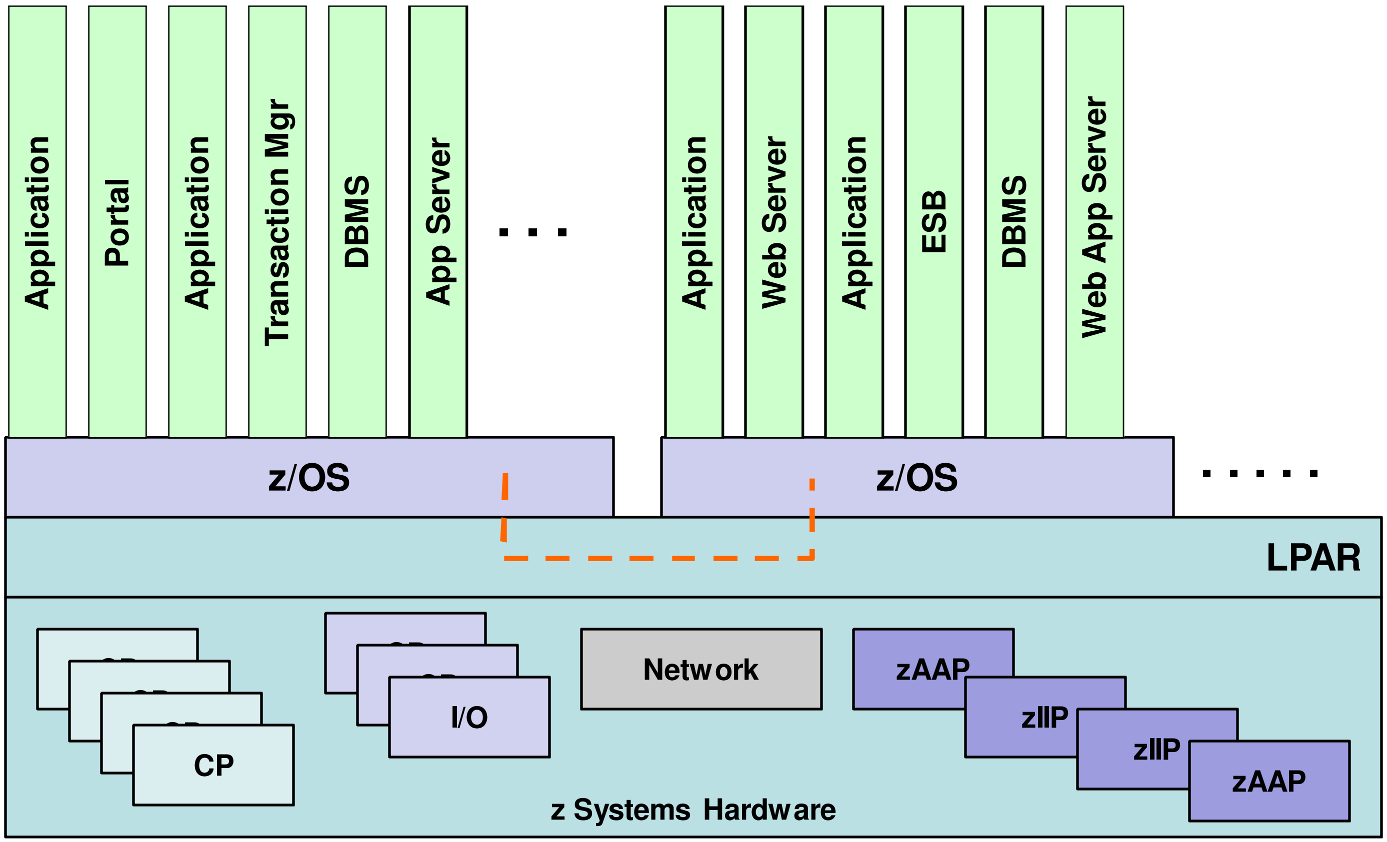
the requirement for the underlying HW support is NOT an issue for z Systems, - since the basic z Systems Architecture & HW design has implemented this for "decades".

- ***Intel and AMD***

is developing some virtualization HW support based on a similar structure like the SIE architecture as it was externally documented 25 years ago.

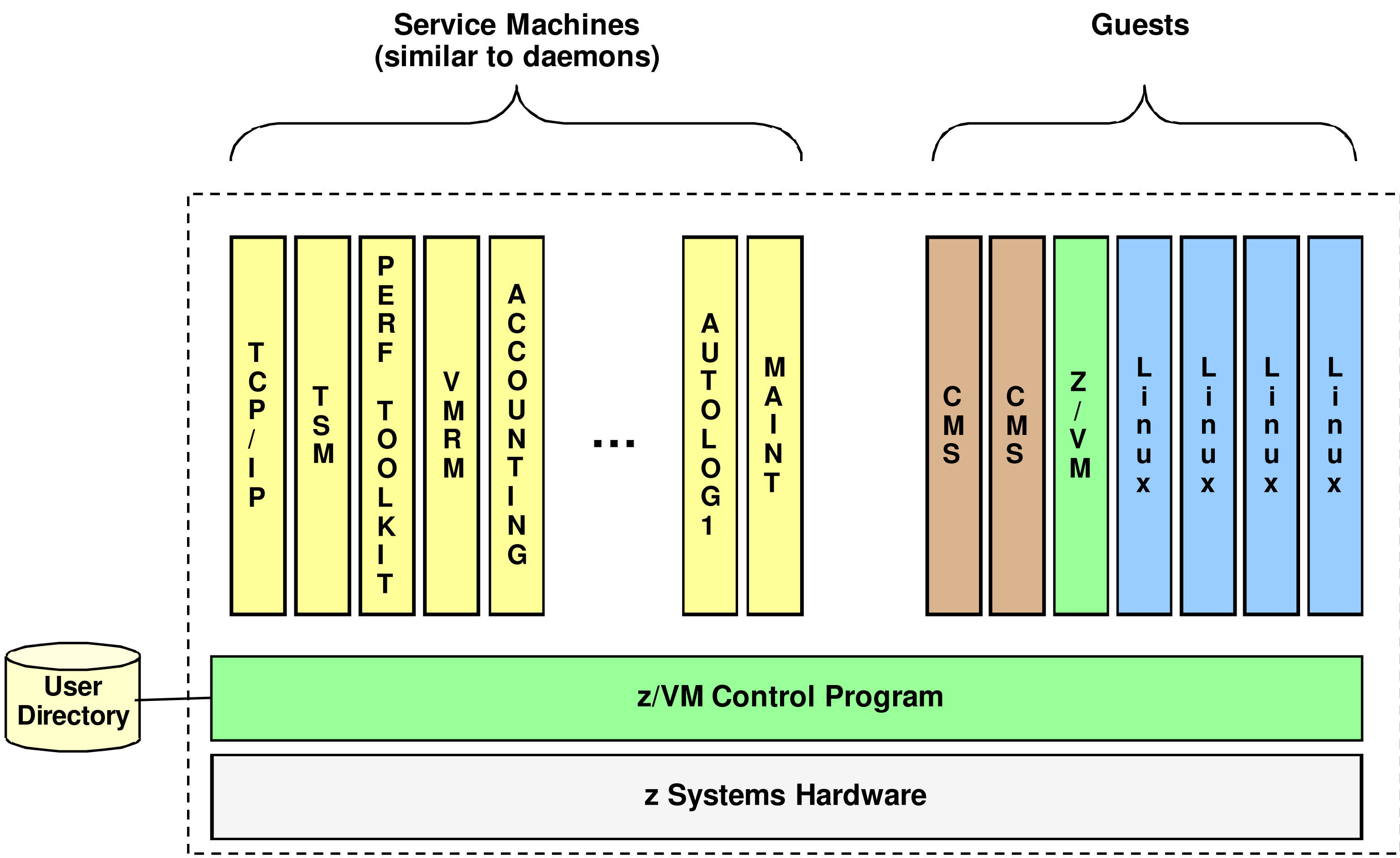
The amount of HW and SIE assist functionalities are seemingly rather limited at this point.

z/OS – Application Virtualization

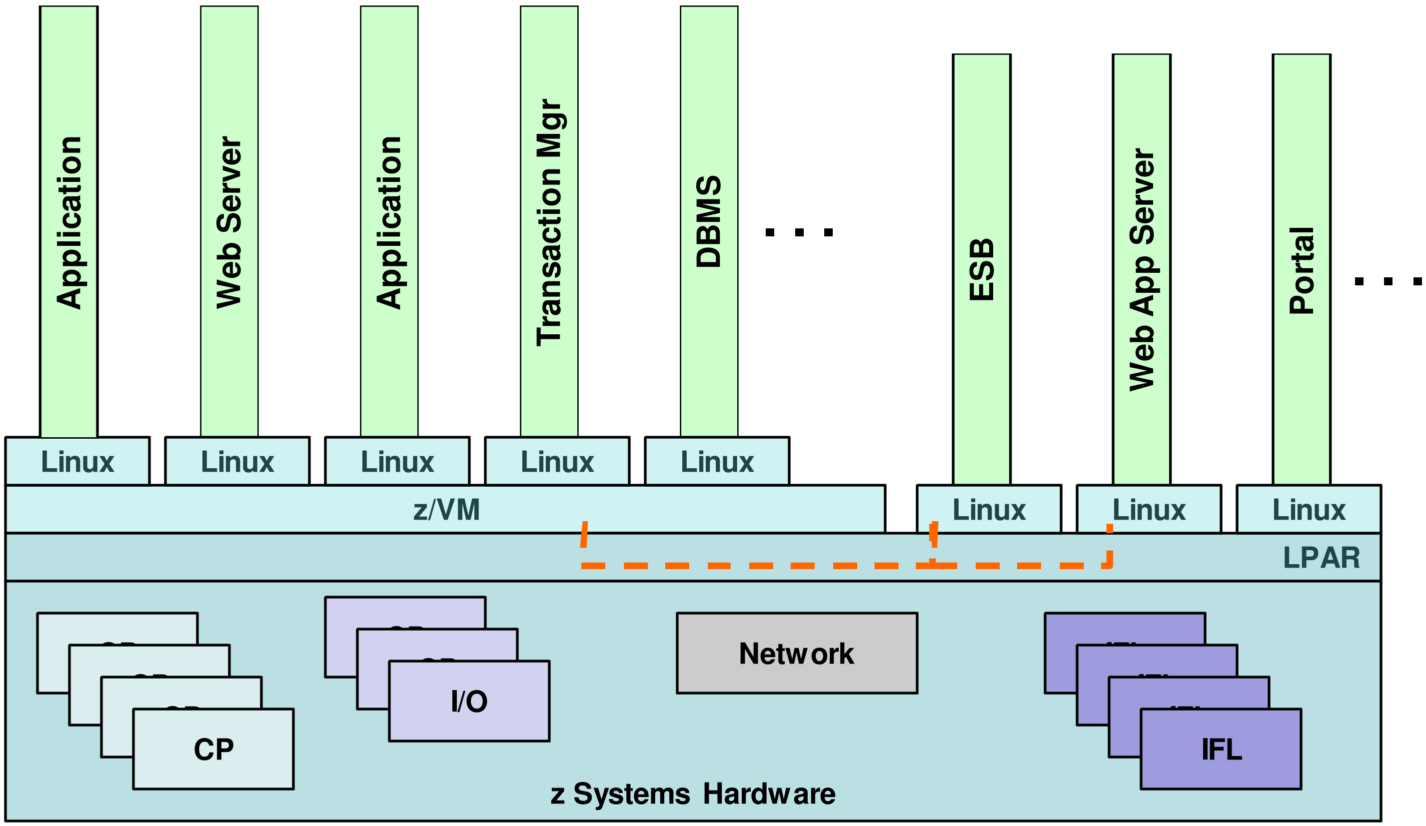


- **"Multiple Virtual Storage" (MVS) was the old name for z/OS**
- **Multiple applications and middleware instances per z/OS system**
  - Benefits from proximity between components – performance, simplicity, reliability
- **Multiple z/OS instances (LPARs) per CEC (box)**
- **Networking between LPARs with Hipersockets (<- - - - - - - ->)**

z/VM Virtual Machines



Linux on z Systems – Server Virtualization



- **A distributed architecture implemented in a z Systems frame**
  - Generally ***one function per Linux instance***, like most distributed server implementations
- **Benefits derived from drastically lower environmental, floor space expense, network efficiency & performance**
  - Networking between Linux instances with Hipersockets (< - - - - - - > above) or z/VM VLAN,
- **z/VM virtualization flexibility, ease of instance management (provisioning, monitoring), security**

KVM – Tech Preview on z Systems

- **KVM on z Systems (s390x)**

  - Allows Linux hosted and Linux based virtual machines in LPARs

- **KVM supports nested virtualization (Intel, AMD)**
- **KVM provides parity to XEN**
- **Include virtio-blk-data-plane (QEMU)**

  - High-performance code path for I/O requests from KVM guests

KVM – Kernel-based Virtual Machine

KVM means "Kernel-based Virtual Machine"

KVM is no emulator itself.
KVM just provide an interface /dev/kvm to set up VMs.

KVM is a Linux kernel module that allows a user space program access to hardware virtualization features of various processor architectures.

When the target architecture is the same as the host architecture, QEMU can make use of KVM particular features such as acceleration, otherwise QEMU is required to perform emulation steps.

KVM – QEMU

QEMU stands for »**Quick EMUlator**« and is a processor emulator that relies on dynamic binary translation to achieve a reasonable speed while being easy to port to new host CPU architectures.

Wikipedia at http://en.wikipedia.org/wiki/Qemu

QEMU emulates:

- CPUs, even for different architectures.
- various hardware components needed to create a VM (network card, storage, …)

QEMU does I/O, KVM does CPU, memory and interrupt controller.
QEMU uses KVM as an accelerator to access hardware features.

KVM – libvirt

libvirt is an open source API, daemon and management tool for managing platform virtualization. It can be used to manage Linux KVM, Xen, VMware ESX, QEMU and other virtualization technologies. These APIs are widely used in Orchestration Layer for Hypervisors in the development of a cloud based solution.

Wikipedia at http://en.wikipedia.org/wiki/Libvirt

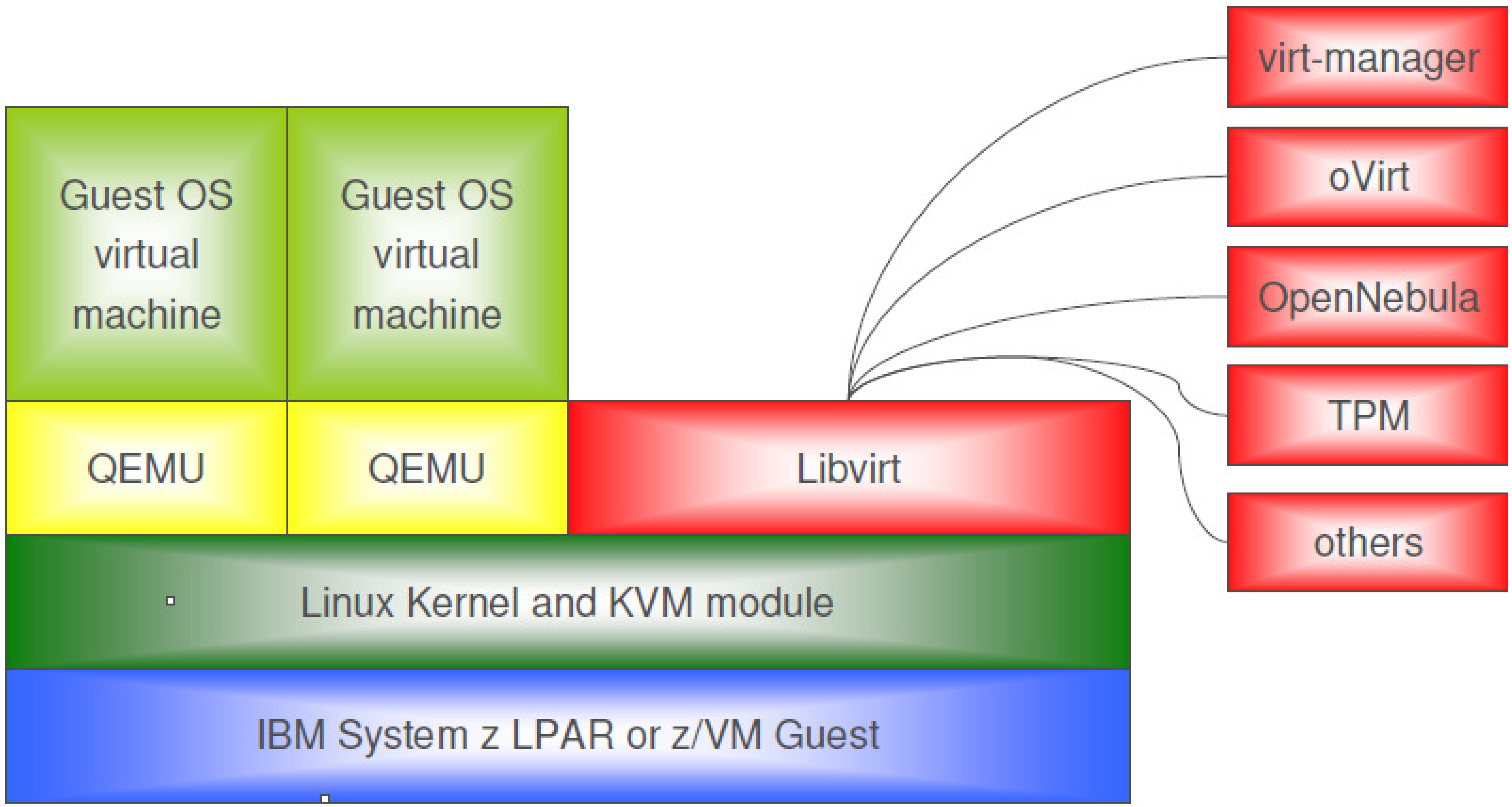
libvirt provides:

A directory for configuration data and operational state of VMs

The **libvirtd deamon** is the server side daemon component of the libvirt virtualization management system.
It runs on host servers and provides remote management "access to libvirt"

The **virsh command shell** is the command line interface for managing guest domains. It is an interactive shell and batch scriptable tool.

KVM – The big picture



Kernel parmline needs an additional parameter

Add 'switch_amode=on' to kernel cmdline

Linux on z Systems – z/VM

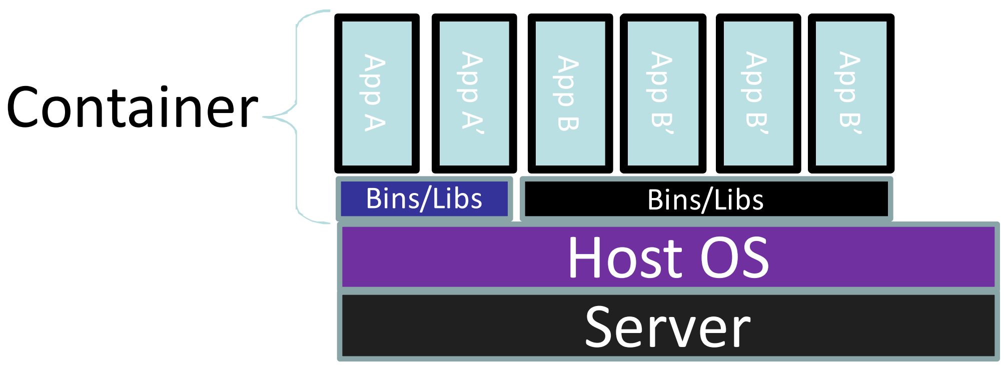
Container

Containers

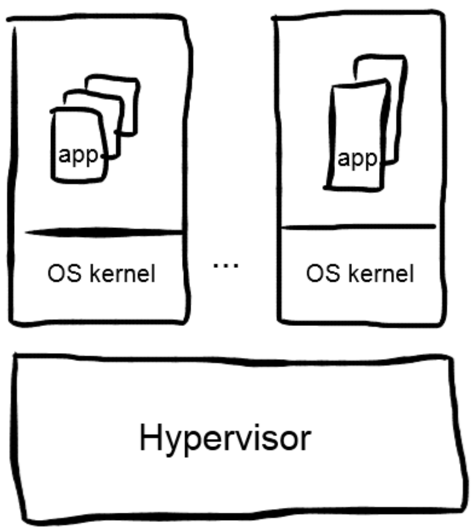
Run

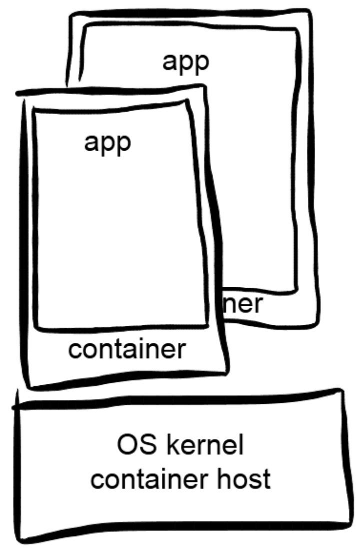
What is a Container

- An isolated user space within a running Linux OS
- Shared kernel across containers
- Direct device access
- All packages and data in an isolated run-time, saved as a filesystem.
- Resource management implemented with cgroups
- Resource isolation through namespaces



Virtualization vs. Containers





Infrastructure oriented:
coming from servers, now virtualized
several applications per server
isolation

Service oriented:
application-centric
solution decomposed
DevOps

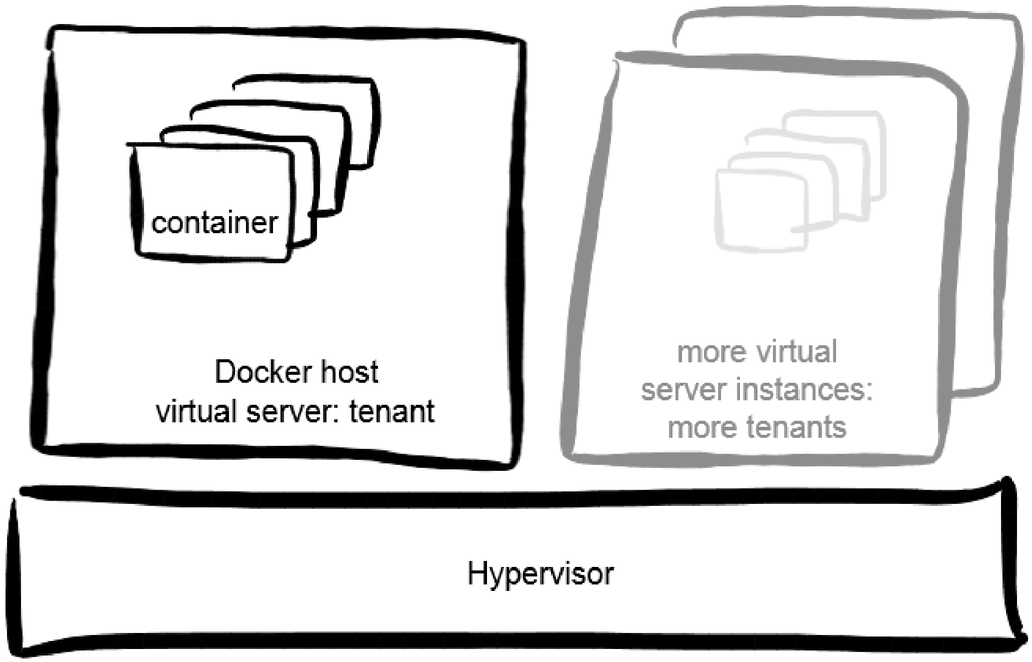
Virtualization and Containers

Virtual machine separation between tenants

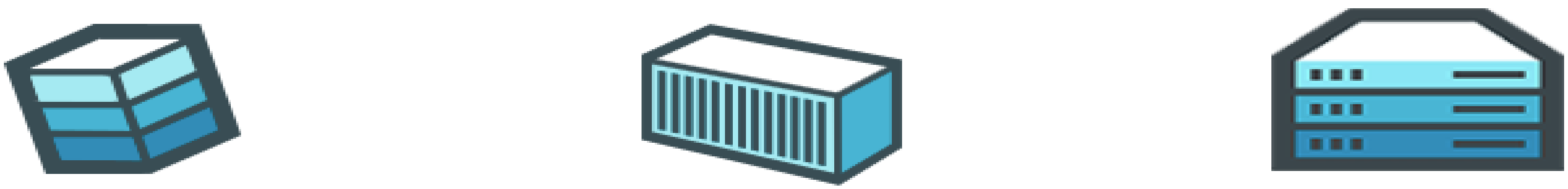
- Virtualization management for infrastructure
- Isolation

Containers within one tenant

- Container efficiency
- Docker management and ecosystem



Docker Engine



Build **Ship** **Run**

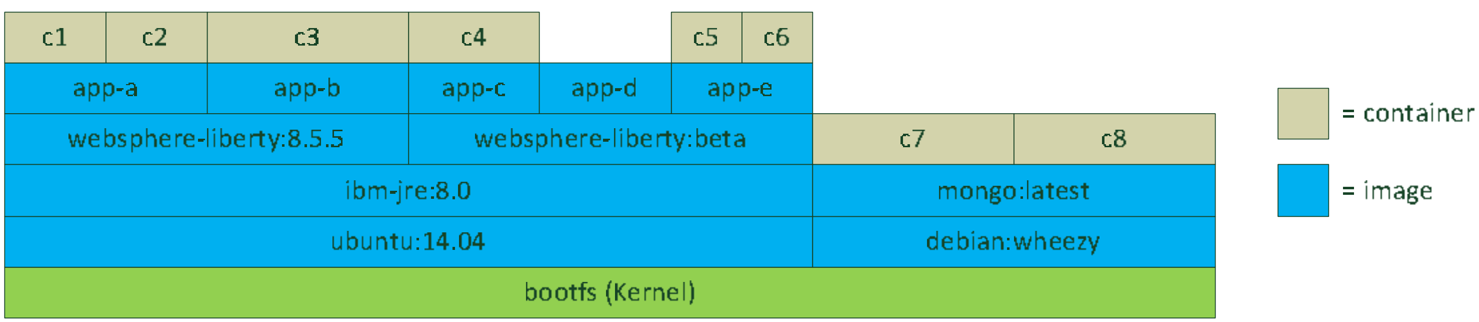
A portable, lightweight application runtime and packaging tool built on top of kernel container primitives

Docker Engine

- **Open source project**
- **Supported on every major Linux distro (MS Windows in 2015)**
- **Client-server architecture with daemon deployed on physical or virtual host**
- **Uses Linux kernel cgroups and namespaces for process resource management and isolation**
- **Uses copy-on-write filesystem for git-like image change management**

Docker Terminology

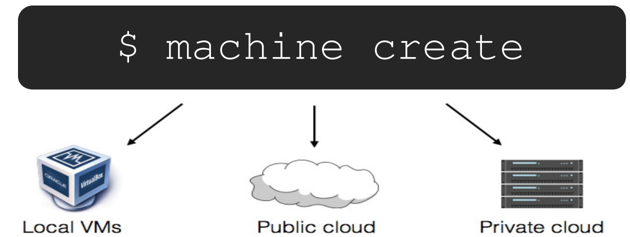
- **Image – layered file system where each layer references the layer below**
- **Dockerfile – build script that defines:**
  - an existing image as the starting point
  - a set of instructions to augment that image (each of which results in a new layer in the file system)
  - meta-data such as the ports exposed
  - the command to execute when the image is run
- **Container – runtime instance of an image plus a read/write layer**



Docker Orchestration

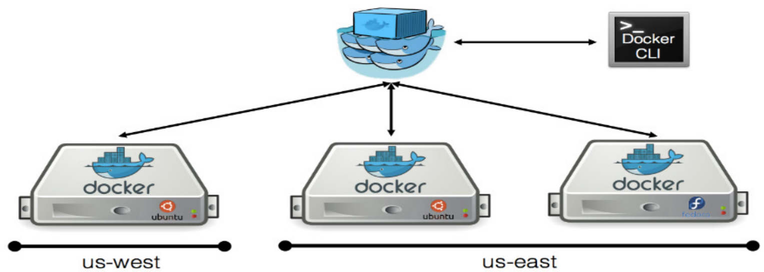
Docker Machine

- Provision Docker daemon onto hosts
- Common CLI for all Docker hosts
- 10 integrations, including AWS, VMware…



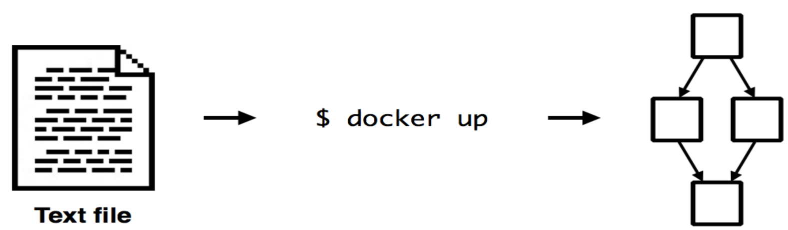
Docker Swarm

- Cluster Docker hosts into a single pool
- Schedule Docker container workloads based on resource availability

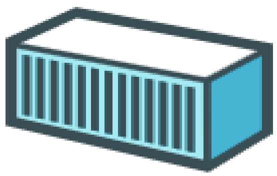


Docker Compose

- Define multi-container distributed apps
- Control all containers via single command

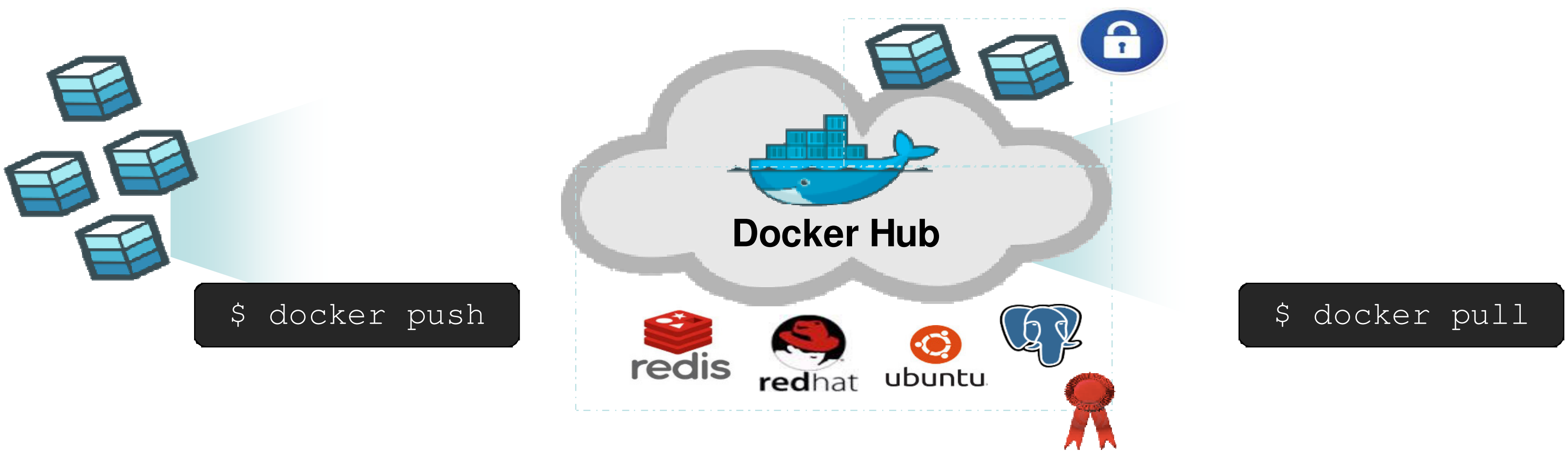


Docker Hub



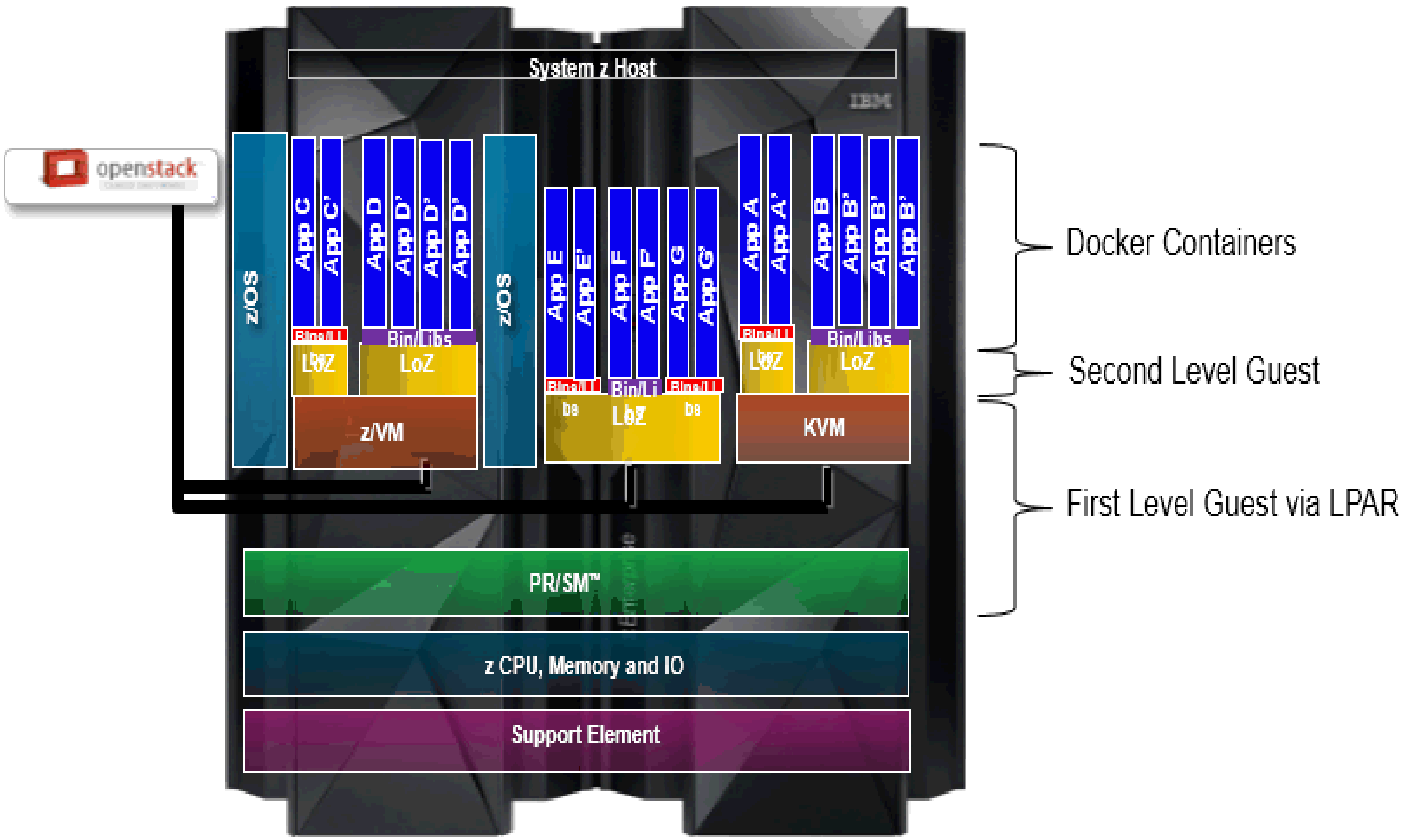
Ship

- **Enable sharing and collab of Docker Images**
- **Private and public repositories of images**
- **Certified base images by ISVs**



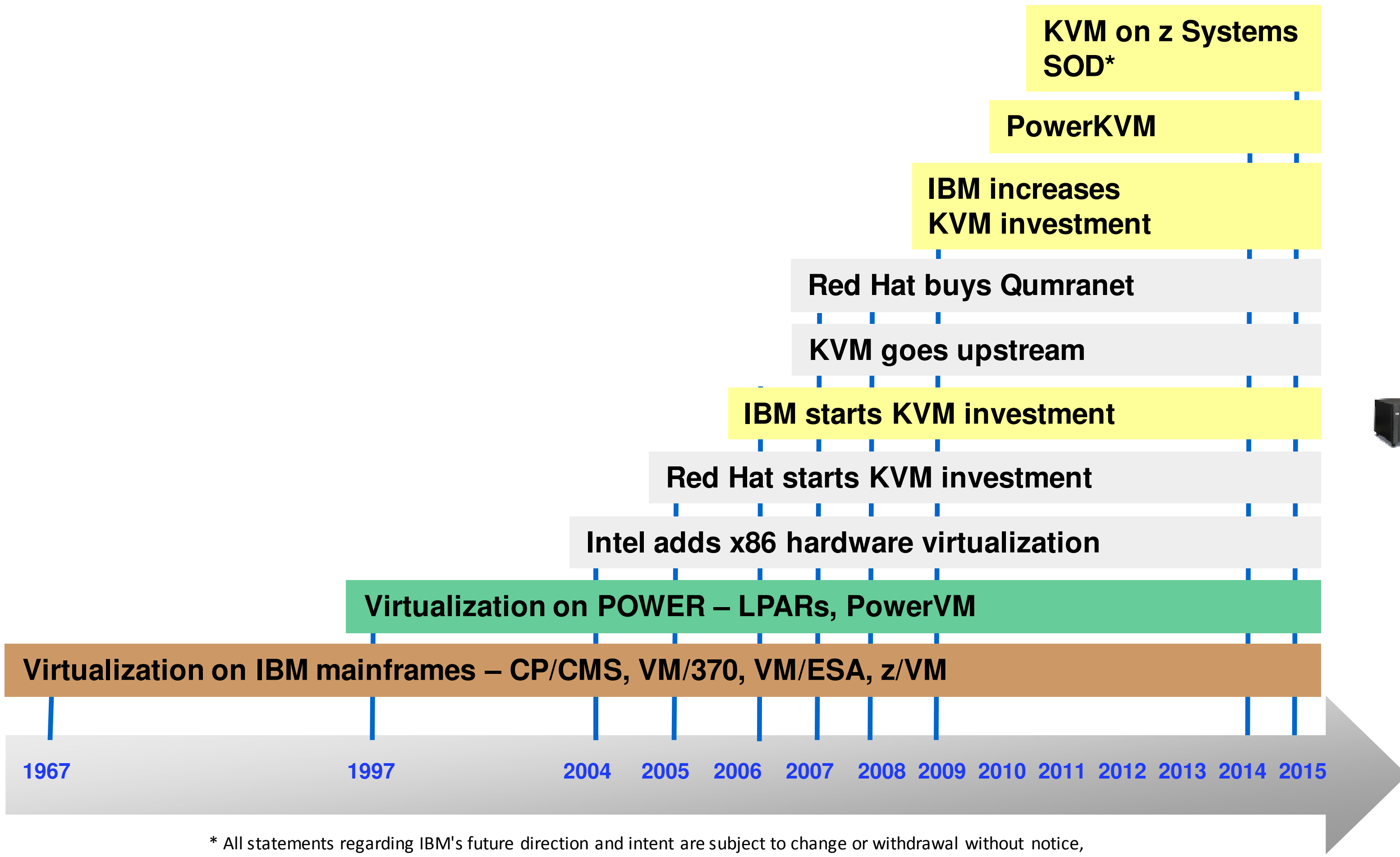
z System's Extreme Virtualization

Built into the architecture not an "add on" feature



A Brief History of IBM and Virtualization

IBM has more than 45 years of experience in server virtualization.
Virtualization was originally developed to make better use of critical hardware.
Hardware support for virtualization has been critical to its adoption.



* All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Any reliance on these Statements of General Direction is at the relying party's sole risk and will not create liability or obligation for IBM.

THANK YOU

A list of terms…

- BCP: Base Control Program
- CBU: Capacity Backup
- CEC: Central Electronics Complex
- CF: Coupling Facility
- CHPID: Channel Path ID
- CICS: Customer Information Control System
- CP: Central Processor
- CSE: Cross System Extensions
- CUoD: Capacity Upgrade on Demand
- DASD: Direct Access Storage Device
- DCSS: Discontiguous Shared Storage
- ESCON: Enterprise System Connection
- ETR: External Time Reference
- FICON: Fibre Connection
- FICON-E: Fibre Connection Express
- FSP: Flexible Service Processor
- GDPS: Geographically Dispersed Parallel Sysplex
- HMC: Hardware Management Console
- HSA: Hardware System Area
- ICB: Integrated Cluster Bus
- ISC: Inter Systems Channel
- ICF: Internal Coupling Facility
- IFL: Integrated Facility for Linux
- IMS: Information Management System
- ISPF: Interactive System Productivity Facility
- JES: Job Entry Subsystem
- LCSS: Logical Channel Sub-system
- LIC: Licensed Internal Code
- LPAR: Logical Partition
- MBA: Memory Bus Adapter
- MCM: Multi-Chip Module
- MIF: Multi-Image Facility
- MQ: Message Queuing
- NIC: Network Interface Card
- OSA: Open Systems Adapter
- PPRC: Peer to Peer Remote Copy
- PR/SM: Processor Resource/Systems Manager
- PU: Processor Unit
- RACF: Resource Access Control Facility
- RMF: Resource Measurement Facility
- SAP: System Assist Processor
- SE: Support Element
- STI: Self Timed Interface
- STP: Server Time Protocol
- TSO: Time Sharing Option
- VIPA: Virtual IP Address
- VM: Virtual machine
- XRC: Extended Remote Copy
- zAAP: zSeries Application Assist Processor
- zIIP: zSeries Integrated Information Processor