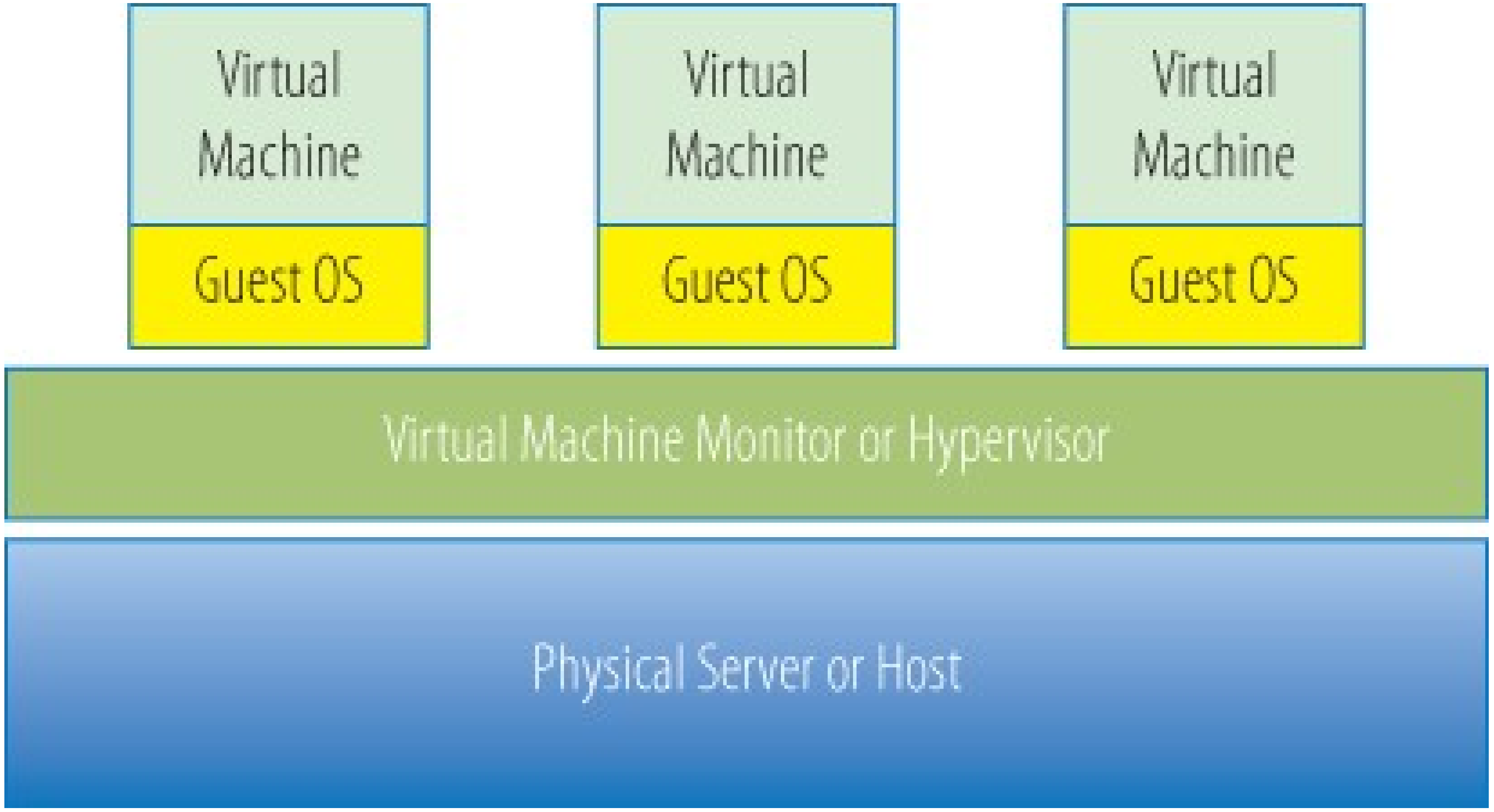
Virtualization

- Virtualization in computing often refers to the abstraction of some physical component into a logical object.
- By virtualizing an object, you can obtain some greater measure of utility from the resource the object provides.
- For example, virtual LANs (local area networks), or VLANs, provide greater network performance and improved manageability by being separated from the physical hardware.
- Likewise, storage area networks (SANs) provide greater flexibility, improved availability, and more efficient use of storage resources by abstracting the physical devices into logical objects that can be quickly and easily manipulated.

History

- The first mainstream virtualization was done on IBM mainframes in the 1960s, but Gerald J. Popek and Robert P. Goldberg codified the framework that describes the requirements for a computer system to support virtualization. Their 1974 article “Formal Requirements for Virtualizable Third Generation Architectures” describes the roles and properties of virtual machines and virtual machine monitors that we still use today. The article is available for purchase or rent at http://dl.acm.org/citation.cfm?doid=361011.361073.
- By their definition, a virtual machine (VM) can virtualize all of the hardware resources, including processors, memory, storage, and network connectivity. A virtual machine monitor (VMM), which today is commonly called a hypervisor, is the software that provides the environment in which the VMs operate.

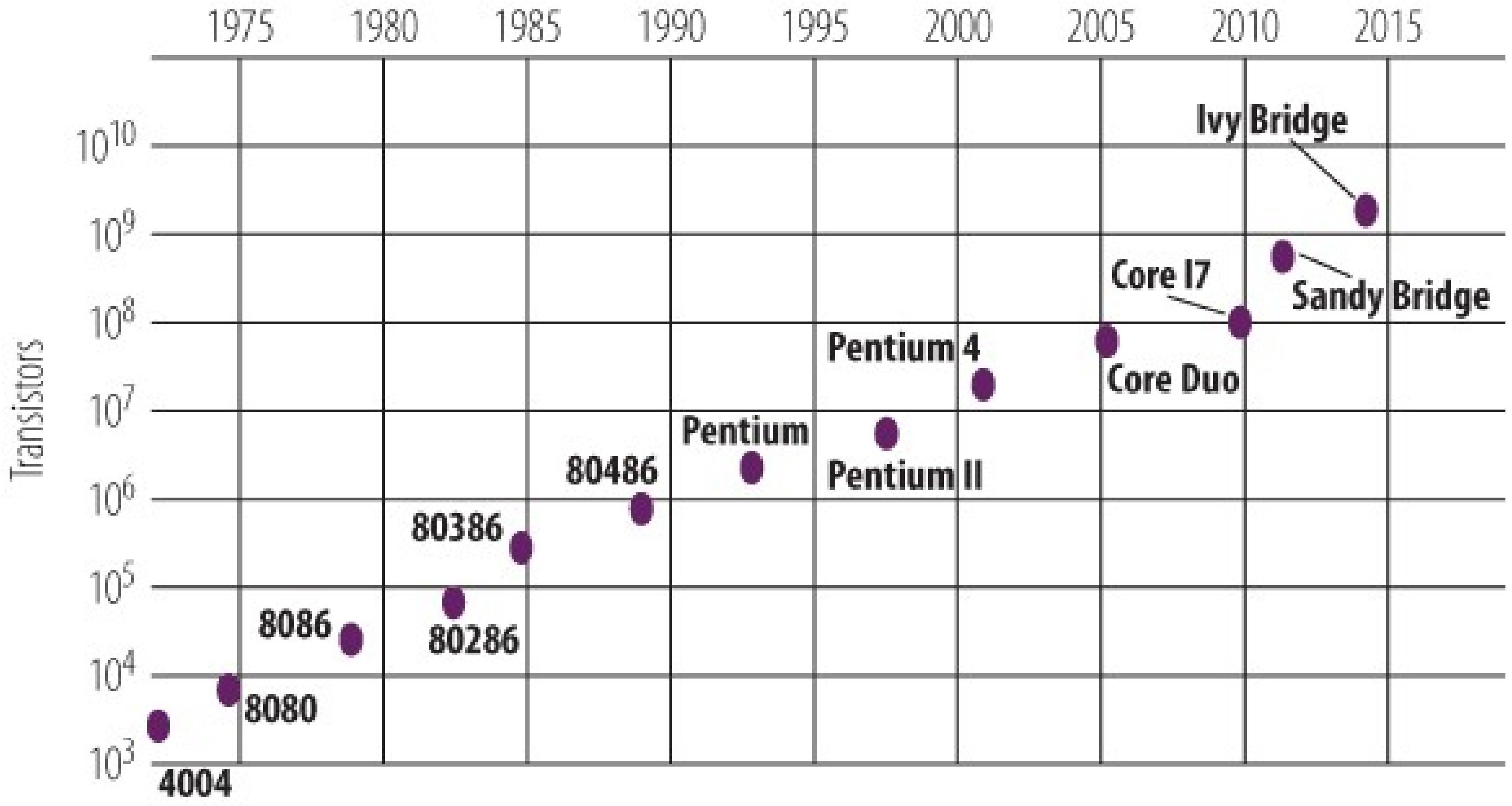

Virtual Machine
Guest OS
Virtual Machine
Guest OS
Virtual Machine
Guest OS
Virtual Machine Monitor or Hypervisor
Physical Server or Host

- According to Popek and Goldberg, a VMM needs to exhibit three properties in order to correctly satisfy their definition:
  - **Fidelity** The environment it creates for the VM is essentially identical to the original (hardware) physical machine.
  - **Isolation or Safety The VMM** must have complete control of the system resources.
  - **Performance** There should be little or no difference in performance between the VM and a physical equivalent.
- Because most VMMs have the first two properties, VMMs that also meet the final criterion are considered efficient VMMs.

Moore’s Law

- So far you have seen how a combination of events—the rise of Windows, corporations increasing their reliance on server technology, and the appearance and mushrooming of the Internet and other content-driven channels—all contributed to accelerated growth of the worldwide server population. One 2006 study estimated that the 16 million servers in use in 2000 had grown to almost 30 million by 2005. This trend continues today. Companies like Microsoft, Amazon, and Google each have hundreds of thousands of servers to run their businesses. Think about all of the many ways you can pull information from the world around you; computers, mobile devices, gaming platforms, and television set tops are only some of the methods, and new ones appear every day. Each of them has a wide and deep infrastructure to support those services, but this is only part of the story. The other piece of the tale has to do with how efficient those computers were becoming.

- If you are reading an electronic copy of this text on a traditional computer, or maybe on a smart phone or even a tablet, you probably have already gone through the process of replacing that device at least once. Phone companies typically give their customers the ability to swap out older smart phones every couple of years for newer, more up-to-date models, assuming you opt for another contract extension. A computer that you bought in 2010 has probably been supplanted by one you purchased in the last three to five years, and if it is closer to five years, you are probably thinking about replacing that one as well. This has little to do with obsolescence, although electronic devices today are rarely engineered to outlive their useful lifespan. It has more to do with the incredible advances that technology constantly makes, packing more and more capability into faster, smaller, and newer packages.
- For example, digital cameras first captured images at less than 1 megapixel resolution and now routinely provide more than 12 megapixel resolutions. PCs, and now smart phones, initially offered memory (RAM) measured in kilobytes; today the standard is gigabytes, an increase of two orders of magnitude. Not surprisingly, there is a rule of thumb that governs how fast these increases take place. It is called Moore's Law, and it deals with the rate at which certain technologies improve



1975
1980
1985
1990
1995
2000
2005
2010
2015
Transistors
10^{10}
10^9
10^8
10^7
10^6
10^5
10^4
10^3
4004
8080
8086
80286
80386
80486
Pentium
Pentium II
Pentium 4
Core Duo
Core I7
Sandy Bridge
Ivy Bridge

- Gordon Moore, one of the founders of Intel, gets credit for recognizing and describing the phenomenon that bears his name. His original thought was publicized back in 1965, and although it has been refined a few times along the way, it is still very true today. Simply stated, Moore's Law says that processing power roughly doubles every 18 months. That means a computer you buy 18 months from now will be twice as powerful as one you buy today. As it turns out, Moore's Law applies not just to processing power (the speed and capacity of computer chips) but to many other related technologies as well (such as memory capacity and the megapixel count in digital cameras). You might think that after almost 50 years, we would be hitting some type of technological barrier that would prevent this exponential growth from continuing, but scientists believe that it will hold true for somewhere between 20 years on the low side and centuries on the high. But what does this have to do with straining data centers and ballooning server growth?

- Processor Speed Increases Over Six Years

Year	2015	2016	2017	2018	2019	2020
Processor Speed	1x	2x	4x	4x	8x	16x
Three-year plan	purchase	purchase				
Five-year plan	purchase					

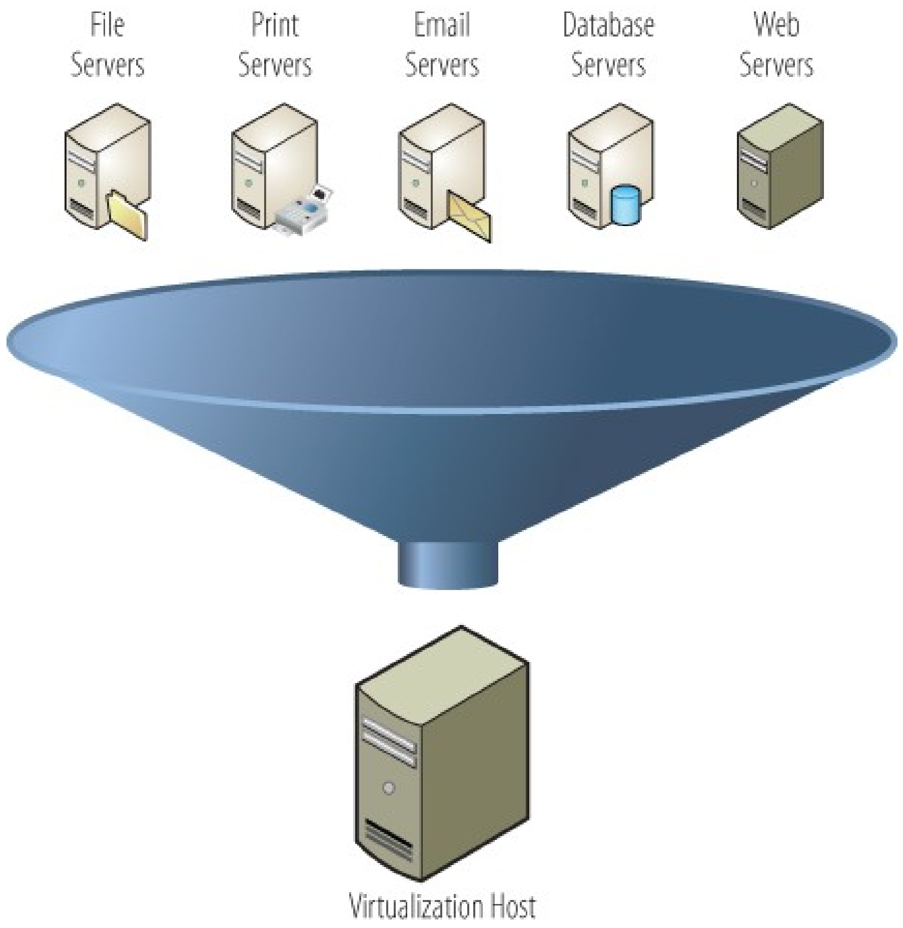
- In addition to faster CPUs and faster processing, newer servers usually have more memory, another benefit of Moore's Law. The bottom line is that the replacement servers are considerably larger and much more powerful than the original server, which was already oversized for the workload it was handling.

Importance of Virtualization

- This is where the two stories come together. There was a wild explosion of data centers overfilled with servers; but as time passed, in a combination of the effect of Moore's Law and the “one server, one application” model, those servers did less and less work. Fortunately, help was on the way in the form of virtualization. The idea and execution of virtualization was not new. It ran on IBM mainframes back in the 1960s but was updated for modern computer systems. We'll come back to the specifics of virtualization in a moment, but in keeping with Popek and Goldberg's definition, virtualization allows many operating system workloads to run on the same server hardware at the same time, while keeping each virtual machine functionally isolated from all the others. The first commercially available solution to provide virtualization for x86 computers came from VMware in 2001.

- A parallel open-source offering called Xen arrived two years later. These solutions (VMMs, or hypervisors) took the form of a layer of software that lived either between an operating system and the virtual machines (VMs) or was installed directly onto the hardware, or “bare-metal,” just like a traditional operating system such as Windows or Linux.

- What virtualization brought to those overfull data centers and underutilized servers was the ability to condense multiple physical servers into fewer servers that would run many virtual machines, allowing those physical servers to run at a much higher rate of utilization. This condensing of servers is called consolidation.
- A measure of consolidation is called the consolidation ratio and is calculated by counting the number of VMs on a server—for example, a server that has eight VMs running on it has a consolidation ratio of 8:1.
- Consolidation was a boon to beleaguered data centers and operations managers because it solved a number of crucial problems just when a critical threshold had been reached. Even a modest consolidation ratio of 4:1 could remove three-quarters of the servers in a data center.


File Servers
Print Servers
Email Servers
Database Servers
Web Servers
Virtualization Host

- In larger data centers, where hundreds or even thousands of servers were housed, virtualization provided a way to decommission a large portion of servers. This reduced the overall footprint of a data center, reduced the power and cooling requirements, and removed the necessity to add to or construct additional data centers. By extension, with fewer servers, it reduced a company's hardware maintenance costs and reduced the time system administrators took to perform many other routine tasks.

Consolidation Drives Down Costs

- Many studies show that the total cost of ownership for an individual server is somewhere between 3 and 10 times the cost of the server itself over three years. In other words, if a server costs $5,000, the cost of maintaining that server is at least another $5,000 per year. Over three years, that is $20,000 per server (the initial hardware spend plus three years of maintenance costs). Those ownership costs include software, annual software and hardware maintenance, power, cooling, cables, people costs, and more. So in this example, for every hundred servers the company can consolidate, it can save two million dollars the first year and every year afterward.

- side from consolidation, a second development took place. As companies began to see the benefits of virtualization, they no longer purchased new hardware when their leases were over, or if they owned the equipment, when their hardware maintenance licenses expired. Instead, they virtualized those server workloads. In other words, they staged these application workloads on their existing virtual infrastructures. This is called containment.
- Containment benefited corporations in multiple ways. They no longer had to refresh large amounts of hardware year after year; and all the costs of managing and maintaining those servers—power, cooling, etc.—were removed from their bottom line from that time on. Until the time when virtualization became commercially viable, Moore's Law worked against the existing application/server/data center model; after it became feasible, it actually helped.
- The consolidation ratios of the first generation of x86 hypervisors were in the range of 5:1. As time continued to pass, more powerful chips and larger memory enabled much higher consolidation ratios, where a single physical server could host dozens or hundreds of VMs. Instead of removing three out of four servers, virtualization today can comfortably remove nine out of ten; or with sufficiently configured servers, ninety-nine out of a hundred. As a result, most corporate data centers have reclaimed much of the space that they had lost before virtualization.

Virtualization and Cloud Computing

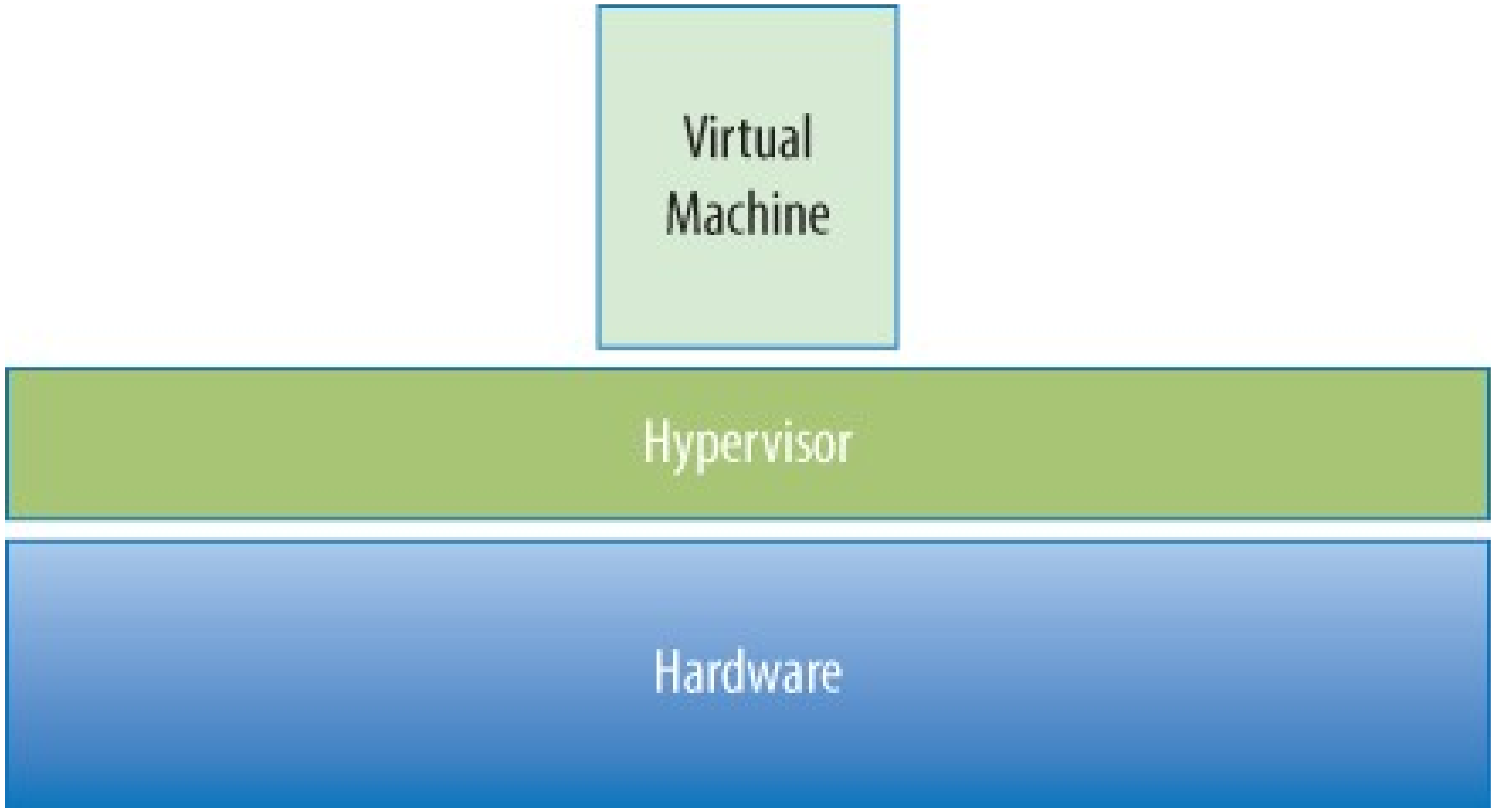
- Virtualization is the engine that drives cloud computing by transforming the data center—what used to be a hands-on, people-intensive process—into a self-managing, highly scalable, highly available pool of easily consumable resources. Before virtualization, system administrators spent 70 percent or more of their time on routine functions and reacting to problems, which left little time for innovation or growth. Virtualization and, by extension, cloud computing provide greater automation opportunities that reduce administrative costs and increase a company's ability to dynamically deploy solutions. By being able to abstract the physical layer away from the actual hardware, cloud computing creates the concept of a virtual data center, a construct that contains everything a physical data center would.
- This virtual data center, deployed in the cloud, offers resources on an as-needed basis, much as a power company provides electricity. In short, these models of computing dramatically simplify the delivery of new applications and allow companies to accelerate their deployments without sacrificing scalability, resiliency, or availability.

Virtualizing Server

- The model for server virtualization, as you saw earlier, is composed of physical hardware augmented by two key software solutions. The hypervisor abstracts the physical layer and presents this abstraction for virtualized servers or virtual machines to use. A hypervisor is installed directly onto a server, without any operating system between it and the physical devices. Virtual machines are then instantiated, or booted. From the virtual machine's view, it can see and work with a number of hardware resources. The hypervisor becomes the interface between the hardware devices on the physical server and the virtual devices of the virtual machines. The hypervisor presents only some subset of the physical resources to each individual virtual machine and handles the actual I/O from VM to physical device and back again. Hypervisors do more than just provide a platform for running VMs; they enable enhanced availability features and create new and better ways for provisioning and management as well.
- While hypervisors are the foundations of virtual environments, virtual machines are the engines that power the applications. Virtual machines contain everything that their physical counterparts do (operating systems, applications, network connections, access to storage, and other necessary resources) but packaged in a set of data files. This packaging makes virtual machines much more flexible and manageable through the use of the traditional file properties in a new way. Virtual machines can be cloned, upgraded, and even moved from place to place, without ever having to disrupt the user applications.

Hypervisor

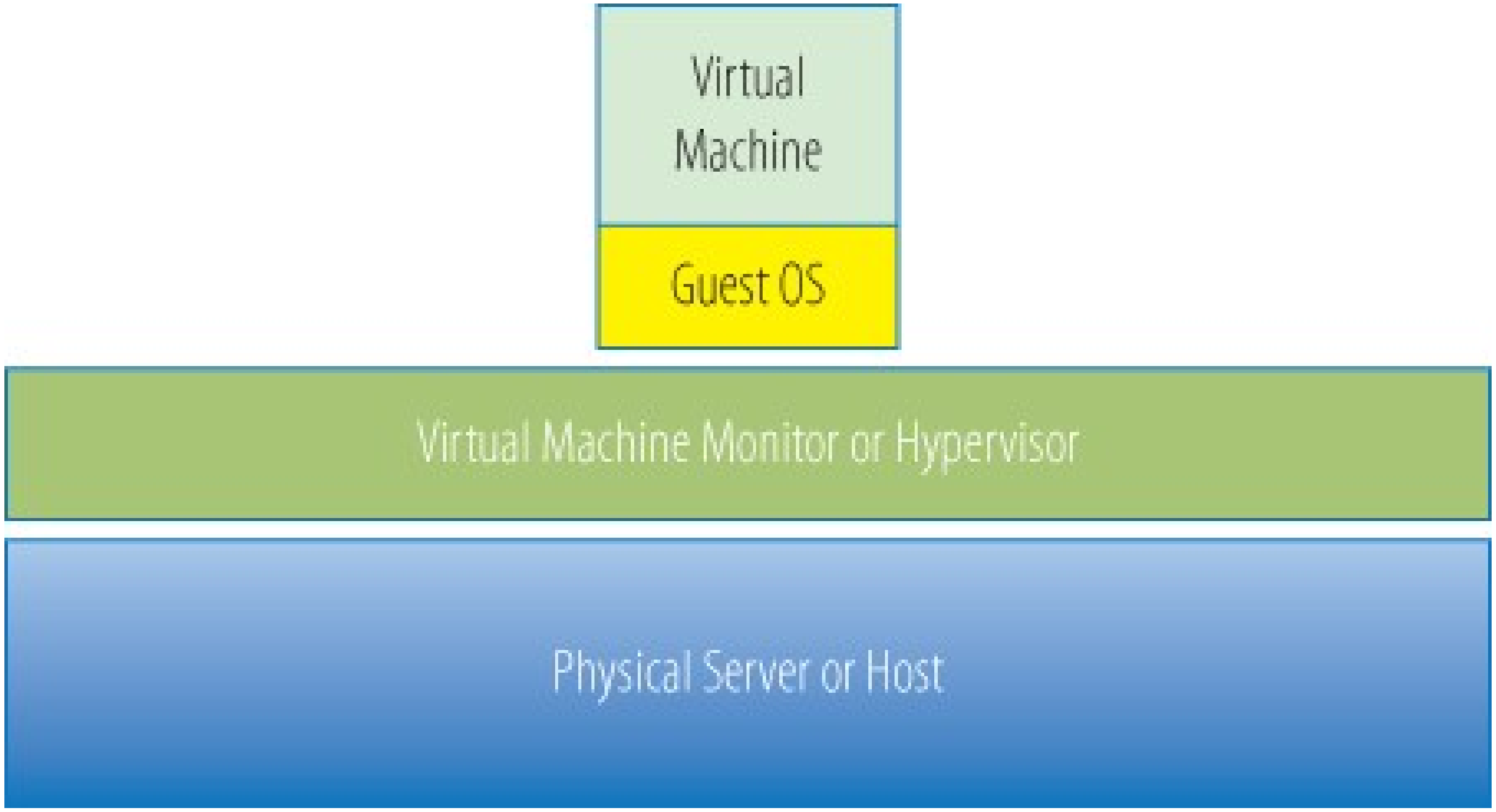
- At the highest level, a hypervisor is an arbiter of resources. It is software that sits between the physical resources on a physical server and the virtual machines that run on that server. In addition to resource allocation, hypervisors provide a virtual environment for those workloads, enable virtual networks for communication between workloads and to the outside world, and offer various forms of clustering for high availability. This is just a small piece of what a hypervisor is; but before looking closer, we need to look at how this all started.
- The original virtual machine monitor (VMM) was created to solve a specific problem. However, VMMs have evolved into something quite different, so much so that the term virtual machine manager has fallen out of favor and has been replaced with the term hypervisor. Today's hypervisors allow us to make better use of the ever-faster processors that regularly appear in the commercial market and to more efficiently use the larger and denser memory offerings that come along with those newer processors. Again, the hypervisor is a layer of software that resides below the virtual machines and above the hardware.


Virtual Machine
Hypervisor
Hardware

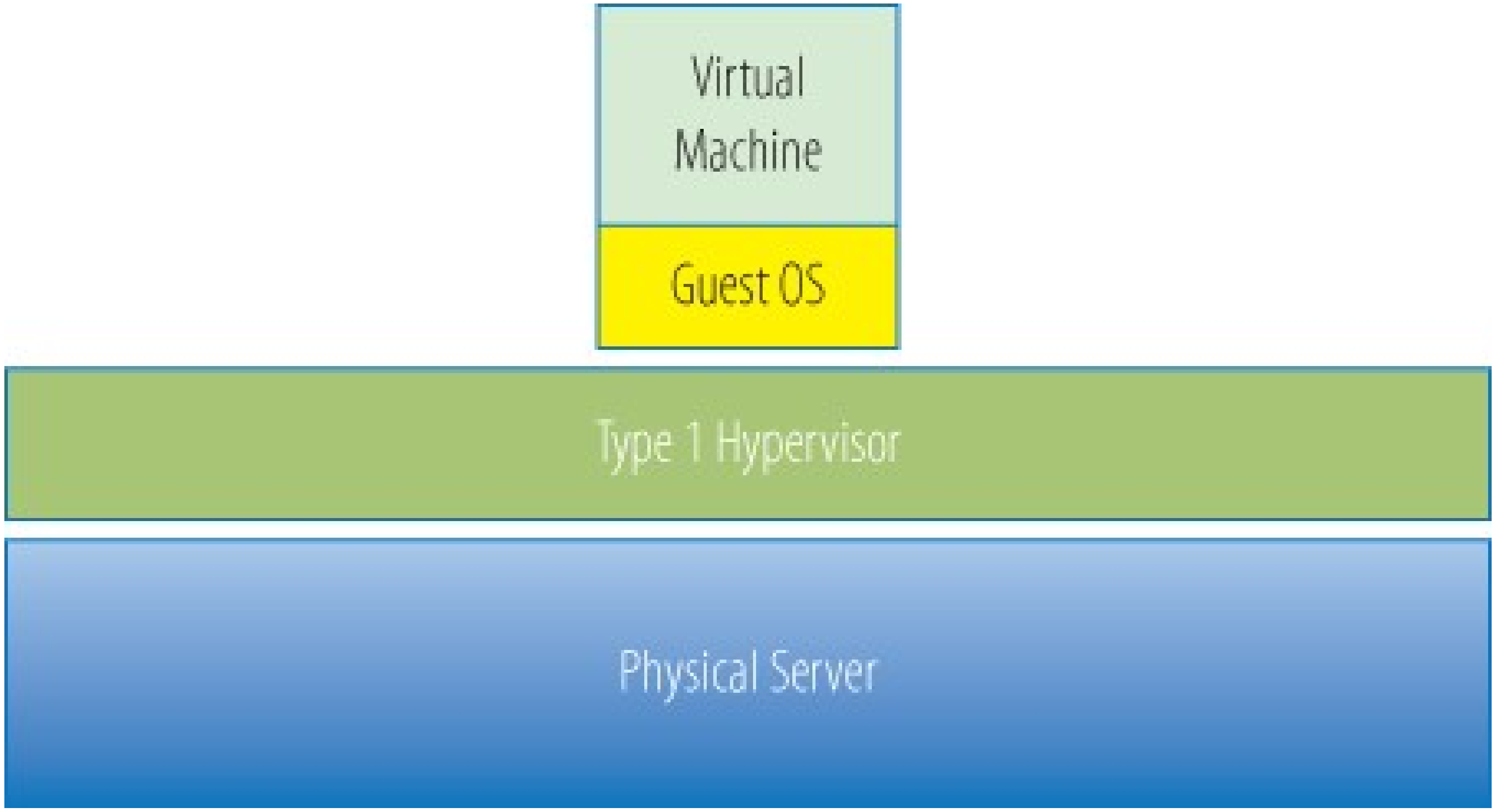
Why Call It a “Hypervisor”?

- Initially, the problem that the engineers were trying to solve was one of resource allocation, trying to utilize areas of memory that were not normally accessible to programmers. The code they produced was successful and was dubbed a hypervisor because, at the time, operating systems were called supervisors and this code could supersede them.

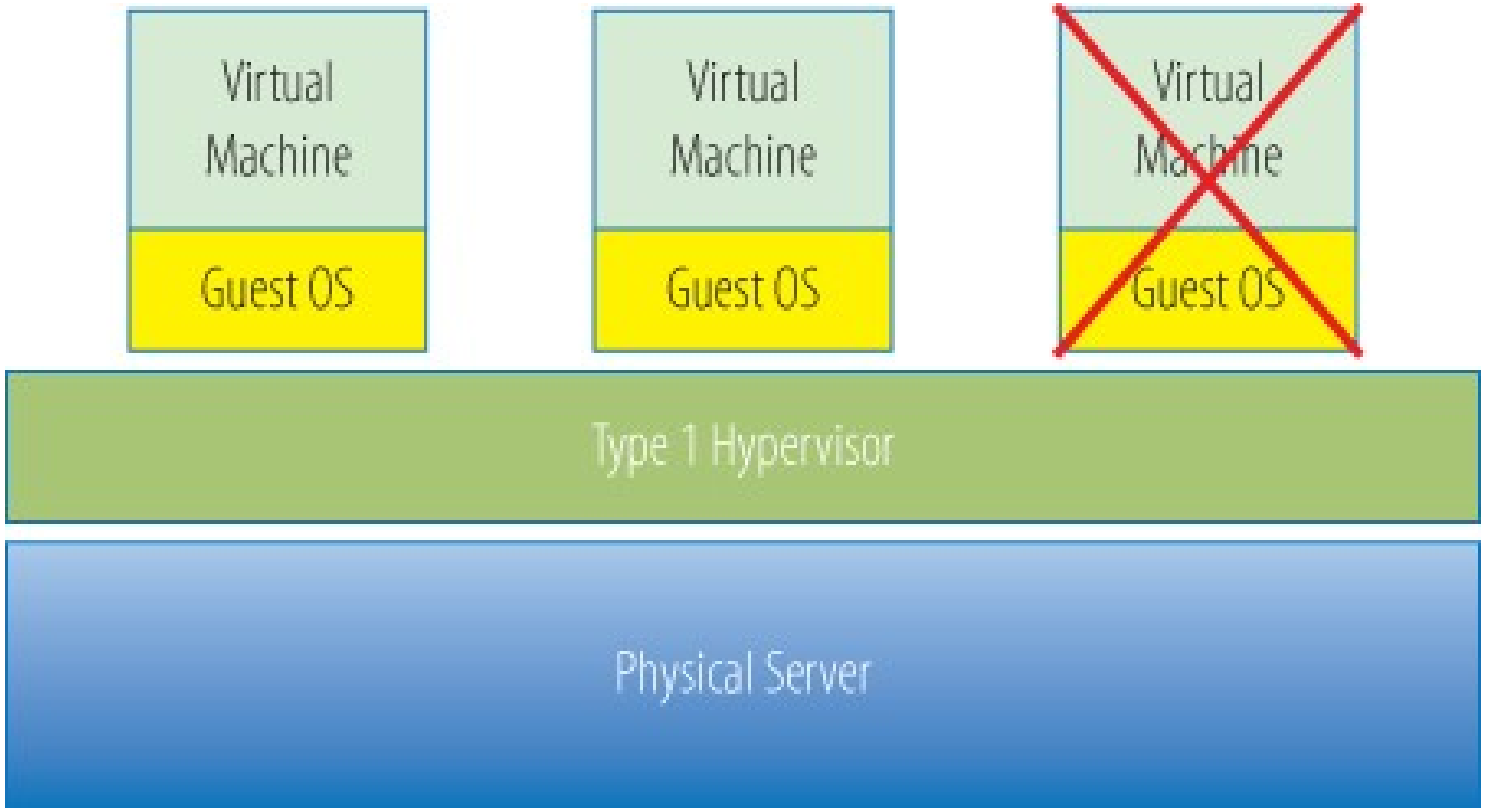
- The structure of a VMM is fairly simple. It consists of a layer of software that lives between the hardware, or host, and the virtual machines that it supports. These virtual machines, or VMs, are also called guests.


Virtual Machine
Guest OS
Virtual Machine Monitor or Hypervisor
Physical Server or Host

- There are two classes of hypervisors, and their names, Type 1 and Type 2, give no clue at all to their differences. The only item of note between them is how they are deployed, but it is enough of a variance to point out.
- A Type 1 hypervisor runs directly on the server hardware without an operating system beneath it. Because there is no other intervening layer of software between the hypervisor and the physical hardware, this is also referred to as a bare-metal implementation. Without an intermediary, the Type 1 hypervisor can directly communicate with the hardware resources in the stack below it, making it much more efficient than the Type 2 hypervisor.

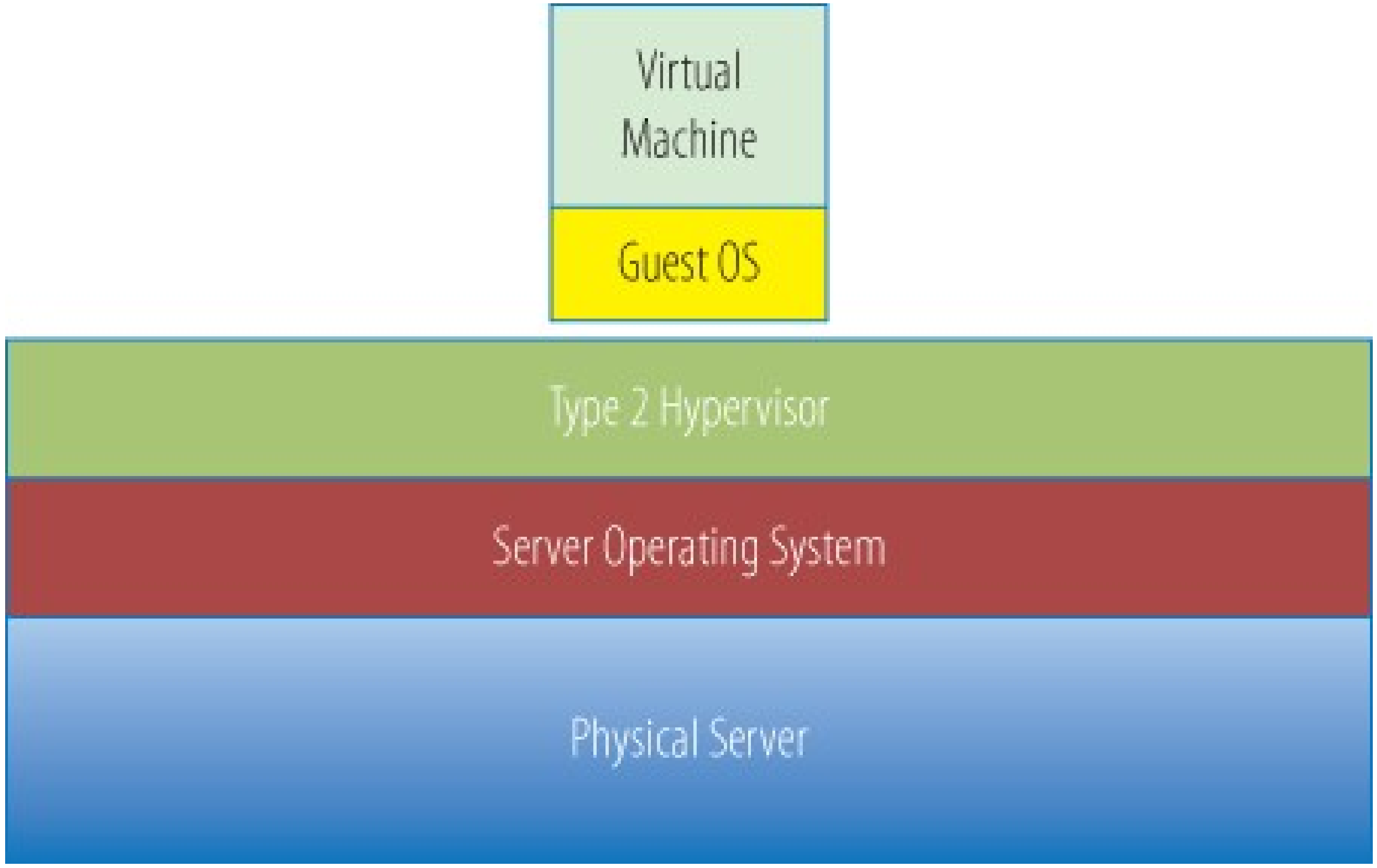

Virtual Machine
Guest OS
Type 1 Hypervisor
Physical Server

- Aside from having better performance characteristics, Type 1 hypervisors are also considered to be more secure than Type 2 hypervisors. Guest operations are handed off and, as such, a guest cannot affect the hypervisor on which it is supported. A virtual machine can damage only itself, causing a single guest crash, but that event does not escape the VM boundaries. Other guests continue processing, and the hypervisor is unaffected as well. A malicious guest, where code is deliberately trying to interfere with the hypervisor or the other guests, would be unable to do so.
- Examples of Type 1 hypervisors include VMware ESX, Microsoft Hyper-V, and the many Xen variants.


Virtual Machine
Guest OS
Virtual Machine
Guest OS
Virtual Machine
Guest OS
Type 1 Hypervisor
Physical Server

- Less processing overhead is required for a Type 1 hypervisor, which means that more virtual machines can be run on each host. From a pure financial standpoint, a Type 1 hypervisor would not require the cost of a host operating system, although from a practical standpoint, the discussion would be much more complex and involve all of the components and facets that comprise a total cost of ownership calculation.

- A Type 2 hypervisor itself is an application that runs atop a traditional operating system. The first x86 offerings were Type 2 because that was the quickest path to market—the actual operating system already handled all of the hardware resources and the hypervisor would leverage that capability.


Virtual Machine
Guest OS
Type 2 Hypervisor
Server Operating System
Physical Server

- One benefit of this model is that it can support a large range of hardware because that is inherited from the operating system it uses. Often Type 2 hypervisors are easy to install and deploy because much of the hardware configuration work, such as networking and storage, has already been covered by the operating system.
- Type 2 hypervisors are not as efficient as Type 1 hypervisors because of this extra layer between the hypervisor itself and the hardware. Every time a virtual machine performs a disk read, a network operation, or any other hardware interaction, it hands that request off to the hypervisor, just as in a Type 1 hypervisor environment. Unlike that environment, the Type 2 hypervisor must then itself hand off the request to the operating system, which handles the I/O requests. The operating system passes the information back to the hypervisor and then back to the guest, adding two additional steps, extra time and more processing overhead, to every transaction.
- VMware Player, VMware Workstation, and Microsoft Virtual Server are examples of Type 2 hypervisors.

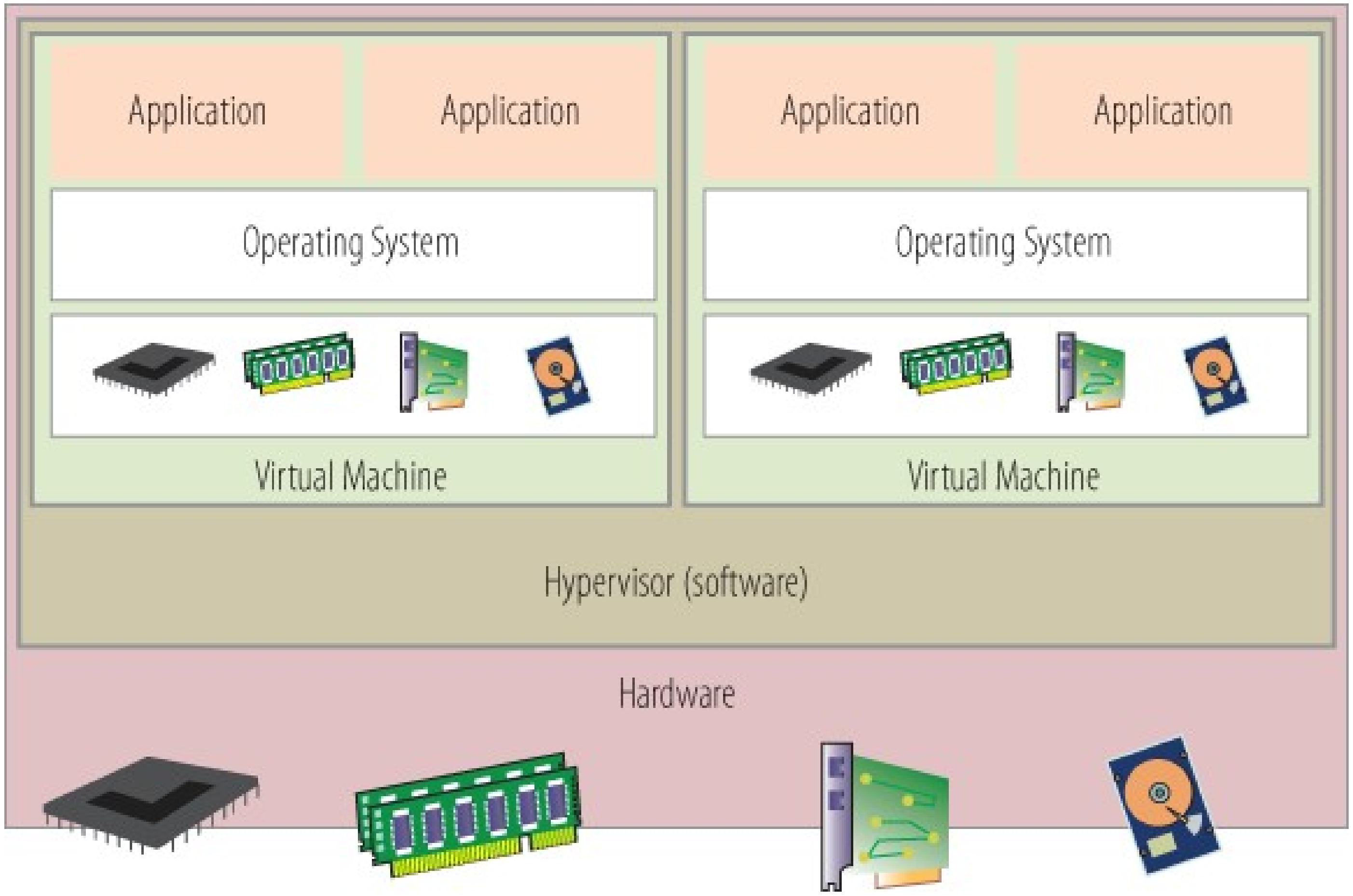
- Type 2 hypervisors are also less reliable because there are more points of failure: anything that affects the availability of the underlying operating system also can impact the hypervisor and the guests it supports. For example, standard operating system patches that require a system reboot would also force reboots of all the virtual machines on that host.
- A Type 2 hypervisor deployment uses more physical resources than a Type 1 hypervisor from the standpoint that the underlying operating system consumes system resources in addition to those consumed by the hypervisor's activities.
- Type 2 hypervisors are typically used in desktop development environments when a single developer is working on one or more virtual machines in building an application. In this case, the hypervisor is actually just an additional desktop application for that developer. Contrast this with a Type 1 hypervisor deployment where the sole function of the physical server is the hosting of virtual machines and no other applications need to run on that hardware.

Rol of hypervisor

- The explanation of a hypervisor up to this point has been fairly simple: it is a layer of software that sits between the hardware and the one or more virtual machines that it supports. Its job is also fairly simple. The three characteristics defined by Popek and Goldberg illustrate these tasks:
  - Provide an environment identical to the physical environment.
  - Provide that environment with minimal performance cost.
  - Retain complete control of the system resources.

Holodecks and Traffic Cops

- In order for many guests to share the physical resources of a host, two things must happen. The first thing is that from the guest's perspective, it has to see and have access to the various hardware resources it needs to function effectively. The operating system in the guest should be able to use disk drives, access memory, and make network calls—or at least believe that it can. This is where the hypervisor steps in.
- Let's use a quick analogy and go back to the virtual reality technology you've seen in films and television. If the technology is sophisticated enough, and can provide the user with a realistic and accurate enough presentation of reality, that user will not be able to distinguish between reality and the virtual reality. In other words, if you were knocked out and you woke up inside of one of the holodecks on the Starship Enterprise, you might not realize that you were actually in a holodeck. From the perspective of a guest operating system, this is what a hypervisor does: it fools the guest into believing that it can actually see and directly interact with the physical devices of the host

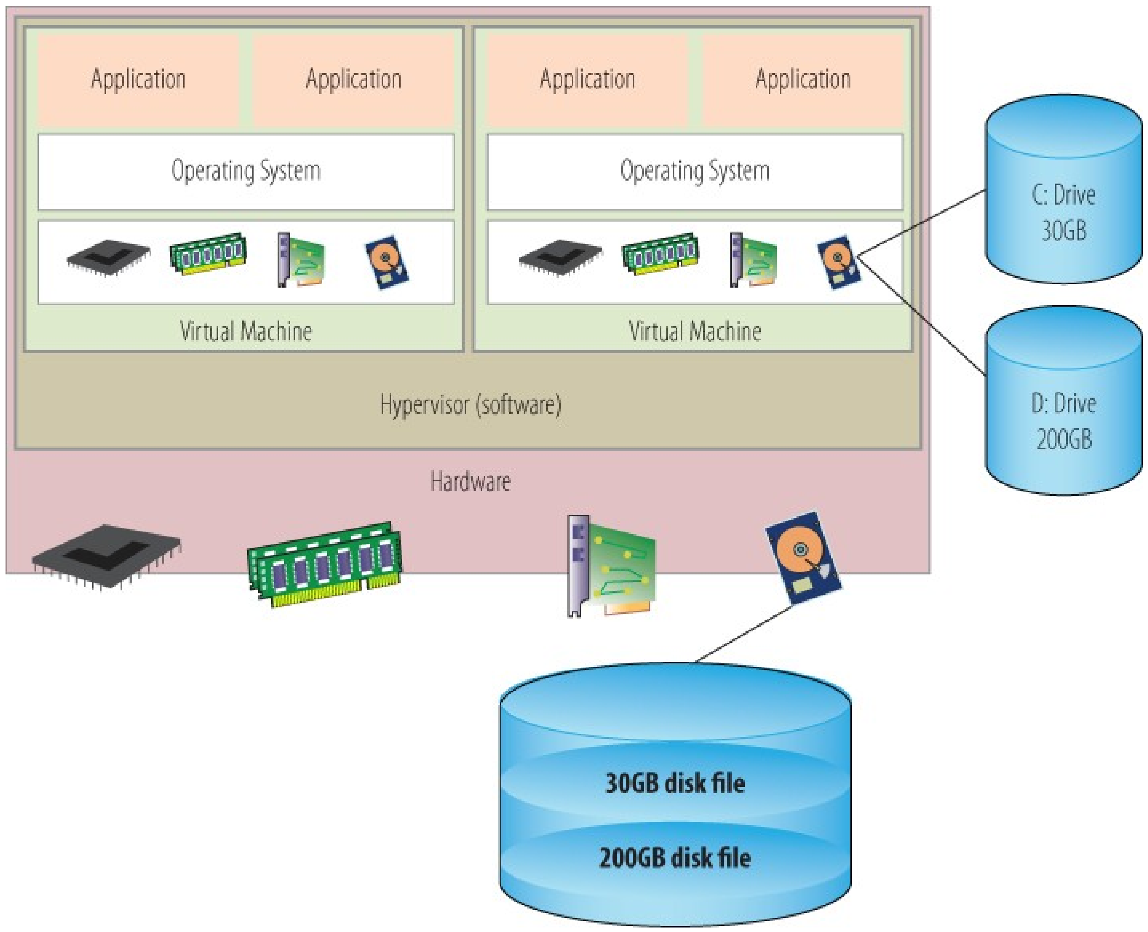

Application
Application
Application
Application
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor (software)
Hardware

- In actuality, each guest is presented with only a fraction of the resources of the physical host. A host may have 256 GB of physical memory installed in its frame, but a guest may believe that it has 4 GB. A guest may be writing files to a 250 GB D: drive, but actually be working with a portion of a file system on a much larger storage area network. Processing and network resources work similarly: a guest may have two virtual CPUs and access to a single Network Interface Card (NIC), but the physical host will have many more of both.

- The second thing that needs to occur is that the hypervisor not only has to abstract the hardware from each of the virtual guests, but it also needs to balance that workload. Each guest makes constant demands on the various resource subsystems. The hypervisor must service all of those demands by acting as an intermediary between each guest and the physical devices, but also do so in a way that provides timely and adequate resources to all. In that way, the hypervisor acts like a traffic cop, controlling the flow of vehicles so that no one has to wait too long in any one direction and all the roads are used fairly.

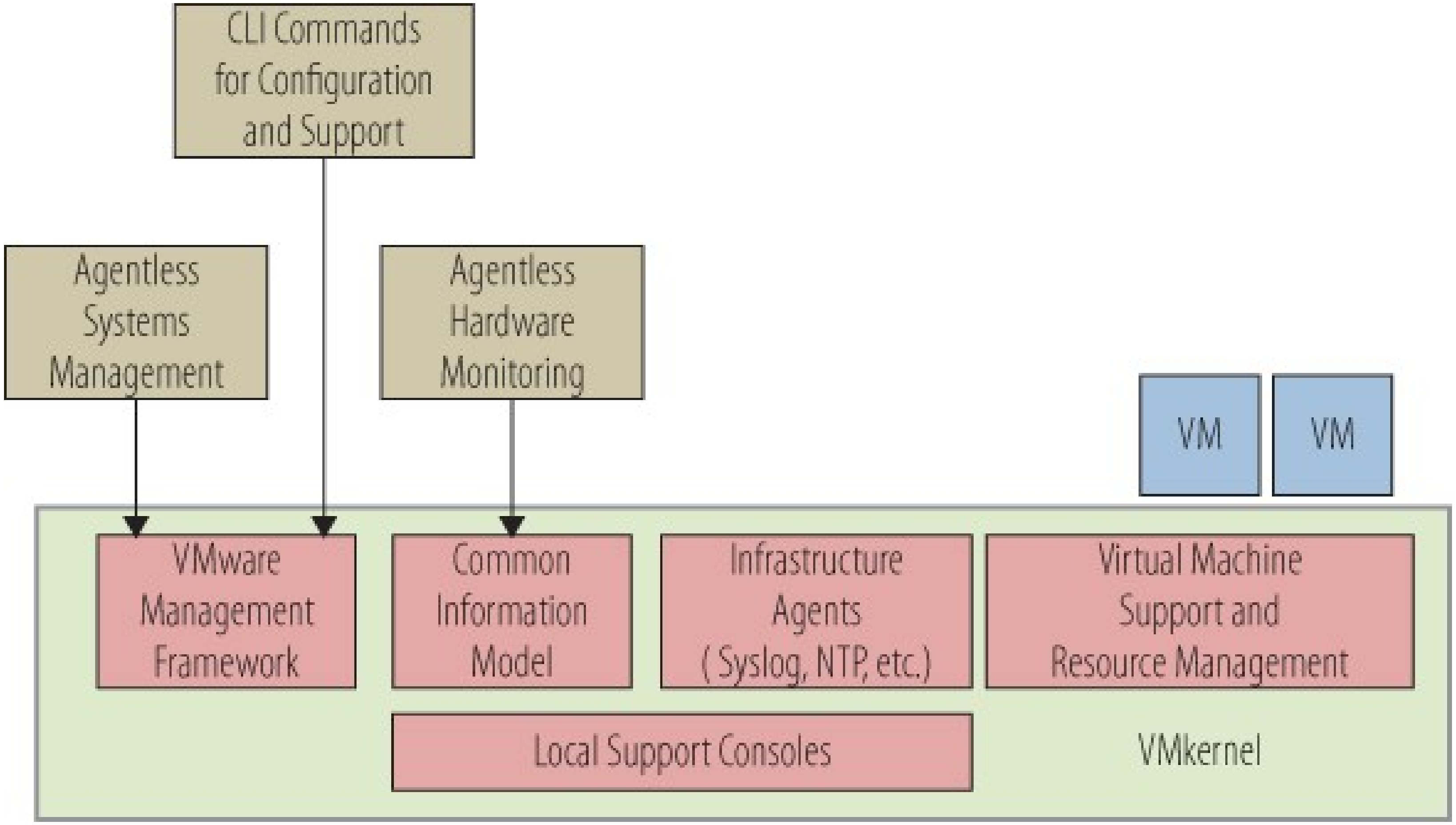
Resource Allocation

- In a way, a hypervisor has become an operating system of sorts for the hardware, but instead of dealing with application/program requests, the hypervisor services entire (virtual) servers.
- A guest application calls for a disk read and passes that request to the guest operating system. The guest operating system makes a read to the disk that it sees, shown in the illustration as the C: or D: drive. Here, the hypervisor steps in and traps that call and translates it into a real-world physical equivalent and passes it to the storage subsystem. In the illustration the data actually resides on shared storage array, and the hypervisor makes that request to the proper storage device and file system. When the response returns, the hypervisor passes the data back to the guest operating system, which receives it as if it came directly from the physical device.


Application
Application
Application
Application
Operating System
Operating System
Virtual Machine
Virtual Machine
Hypervisor (software)
Hardware
C: Drive
30GB
D: Drive
200GB
30GB disk file
200GB disk file

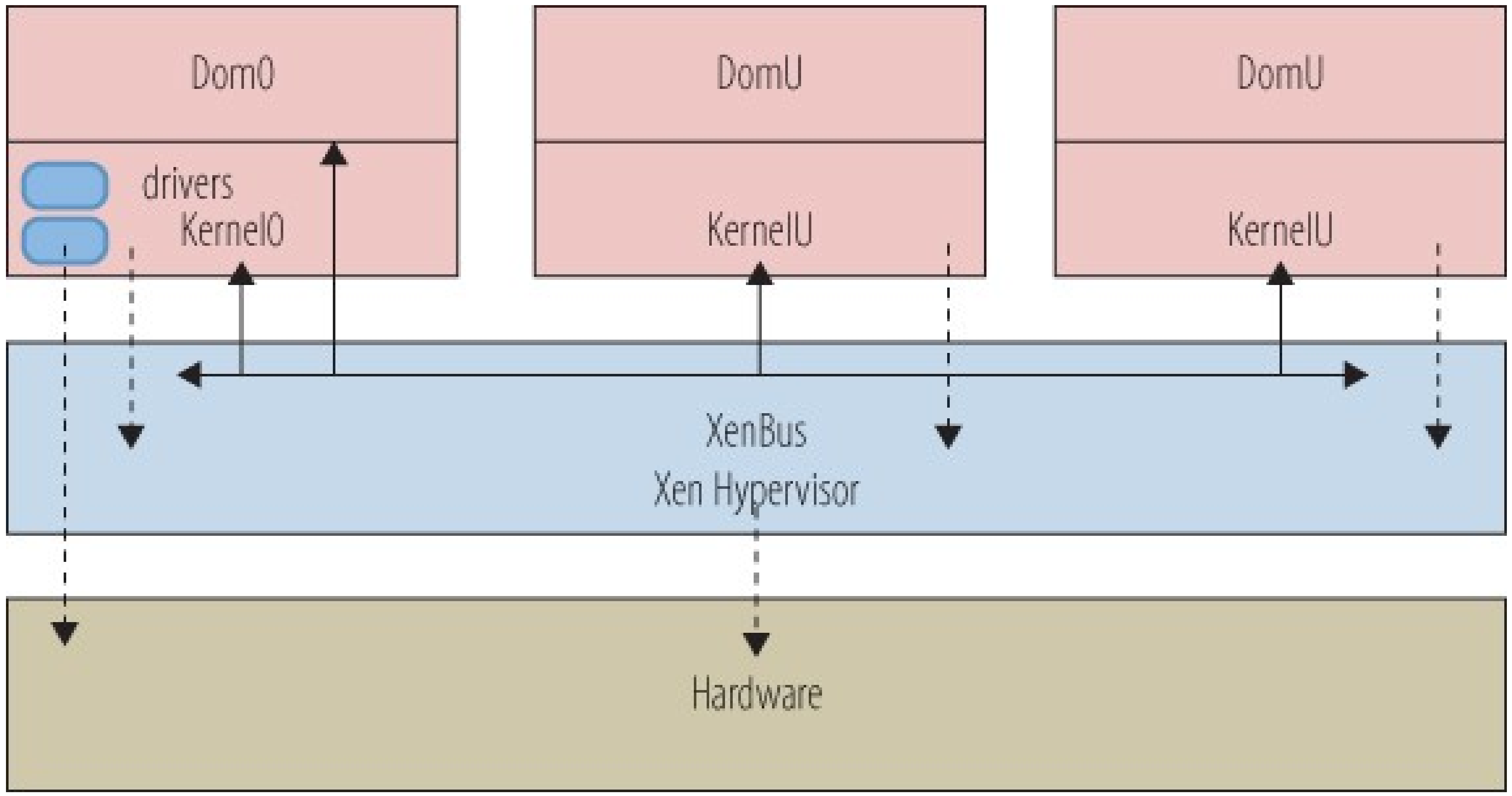
Vmware ESX

- Founded in 1998, VMware was the first company to develop a commercially available x86 virtualization solution. The following year, the company released their first product, Workstation 1.0, which allowed developers to create and work with virtual machines on their Windows or Linux desktops. Two years after that, in 2001, both ESX 1.0 and GSX 1.0 were released. ESX was a Type 1 hypervisor and GSX was a Type 2 hypervisor. Both solutions are still around today and are still being enhanced and updated. GSX, though, has been renamed to VMware Server and is available to download at no cost.
- Which Is It, ESX or ESXi?
- The original architecture of ESX was made up of two parts, the actual hypervisor, which did the virtualization work, and a Linux-based console module that sat alongside the hypervisor and acted as a management interface to the hypervisor. VMware decided that this model was not sustainable for two reasons. The first was that the service console was roughly 30 times the size of the hypervisor—in ESX 3.5, for example, the hypervisor was about 32 MB, while the service console required closer to 900 MB. The second reason was security. Linux is a well-understood environment, and there was concern that the hypervisor could be compromised through the service console. ESXi was developed with the same hypervisor core, but without the service console. The hypervisor is managed through a command-line interface (CLI) and has been re-architected to allow third-party integrations that had been done through agents in the service console. VMware released two versions, classic ESX and ESXi, from version 3.5 in 2007 through version 4.1 in 2010. As of the 2011 release 5, only the ESXi architecture is available.
- Market share is not always the best indicator of a solution's viability and capability; however, 15 years after ESX's first appearance, according to Gartner, VMware still holds close to 70 percent of the market. VMware has done a good job of using their first-to-market advantage to develop features and capabilities that many of the other virtualization vendors are still trying to replicate. We'll cover some of those features in a moment, but first let's take a closer look at ESX.


CLI Commands for Configuration and Support
Agentless Systems Management
Agentless Hardware Monitoring
VM
VM
VMware Management Framework
Common Information Model
Infrastructure Agents (Syslog, NTP, etc.)
Virtual Machine Support and Resource Management
Local Support Consoles
VMkernel

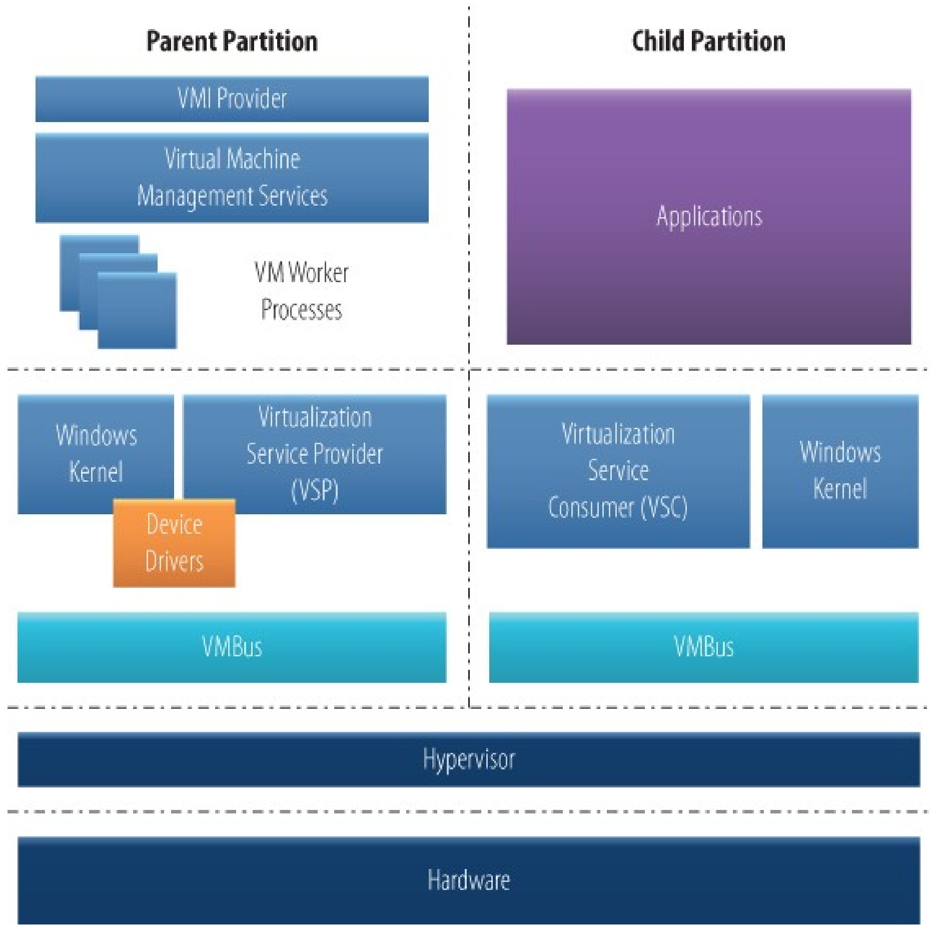
Citrix Xen

- The Xen hypervisor began as a research project in the late 1990s at the University of Cambridge, the goal of which was to create an efficient platform for distributed computing. In 2002, the code was made an open-source project, allowing anyone to contribute to improving the features and capabilities. XenSource was founded in 2004 to bring the Xen hypervisor to market, but the open-source project still remained open, as it does to this day. In 2005, Red Hat, Novell, and Sun all added the Xen hypervisor to their product offerings, bringing it to the mainstream. Other solutions like Oracle VM and Amazon Web Services also use Xen as their virtualization framework. Two years later, Citrix Systems acquired XenSource to complement their application delivery solutions. In 2013, it was announced that the development for Xen would become a Linux Foundation Collaborative Project, returning development to the open-source community.
- The hypervisor is a bare-metal solution and sits directly on the hardware, but the implementation shows some differences from the VMware architecture. The Xen model has a special guest called Domain 0, also referred to as Dom0. This guest gets booted when the hypervisor is booted, and it has management privileges different from the other guests. Because it has direct access to the hardware, it handles all of the I/O for the individual guests. It also handles the hardware device driver support. When additional guests make requests of the underlying hardware resources, those requests go through the hypervisor, up to the Dom0 guest, and then to the resource. Results from those resources reverse that trip to return to the guests.
- Having an operating system in the Dom0 guest can affect availability. When OS patching needs to occur, a reboot of Dom0 will interrupt all of the other guests, even if the patches were not related to the virtualization functions. Because Dom0 is also a guest, it consumes resources and contends for resources with the other guests in the system that could lead to performance issues if Dom0 is either short of resources or using guest resources.


Dom0
drivers
Kernel0
DomU
KernelU
DomU
KernelU
XenBus
Xen Hypervisor
Hardware

Microsoft Hyper-V

- Microsoft began in the virtualization space with Virtual Server in 2005, after they had acquired the solution from Connectix a few years earlier. Like GSX, Virtual Server was a Type-2 hypervisor, but has been discontinued in favor of Hyper-V. Microsoft Hyper-V was released in 2008 as an installable part of the Windows Server 2008 operating system.
- Hyper-V is a Type 1 hypervisor because the hypervisor code lives directly on the hardware. The nomenclature is slightly different, though—rather than guests, the virtualized workloads are called partitions. Similar to the Xen model, it requires a special parent partition that has direct access to the hardware resources. Like Dom0, the parent partition runs an operating system—in this case, Windows Server. This partition creates and manages the child partitions and handles the system management functions and device drivers. Because it utilizes a model similar to XenServer, it is subject to the same availability vulnerabilities regarding patching and contention.


Parent Partition
Child Partition
VMI Provider
Virtual Machine Management Services
VM Worker Processes
Applications
Windows Kernel
Virtualization Service Provider (VSP)
Device Drivers
Virtualization Service Consumer (VSC)
Windows Kernel
VMBus
VMBus
Hypervisor
Hardware

Other Solutions

- In addition to the solutions discussed in the previous sections, there are a large number of other virtualization vendors and solutions that as a group comprise, depending on whose numbers you follow, between 1 percent and 5 percent of the market. Most of the remaining solutions are based on the original open-source Xen code and have been enhanced and updated by the various solution providers.
- **Oracle** offers a number of solutions, both built and acquired. Introduced in 2007, Oracle VM is a bare-metal hypervisor that is based on the open-source Xen code. In 2009, Oracle acquired Virtual Iron, another Xen-based hypervisor solution, with the intent to integrate the technology into the existing Oracle VM. Oracle's acquisition of Sun Microsystems in 2010 brought with it a number of additional virtualization solutions that Sun had developed or acquired as well, including the Solaris-specific Zones and the x86-oriented VirtualBox. VirtualBox has since been rebranded Oracle VM VirtualBox. The Oracle solutions have not gained mainstream traction, mostly due to their later entry into the market and the variety of solutions that is sometimes confusing to users. The users of these solutions are from strong Oracle shops.
- **Red Hat** is another solution that has gone through a few different permutations over time. Initially, they also used the open-source Xen code because it fit nicely with their business model of open-source solutions. In 2008, Red Hat acquired Qumranet and their Kernel-based Virtual Machine (KVM) solution. KVM, like Linux itself, is also based on the open-source project of the same name. For a time, releases of Red Hat Enterprise Linux (RHEL) supported both the KVM and Xen virtualization technologies. In 2012, Red Hat stated that KVM is their future direction, and Xen is no longer supported. KVM is delivered as a Linux kernel module allowing you to leverage many features of Linux, like the scheduler and memory management, without needing to include them as part of the code. Like Oracle, KVM usage has not yet acquired any significant following and is mostly limited to existing users of Red Hat itself. However, the rise of OpenStack, an open-source cloud-computing project, has helped as KVM can be used as one of the hypervisor choices.
- In addition to these, there are about another dozen or so commercial x86-server virtualization solutions available today. The history of this particular software solution area, and other more mature areas in the past, indicates that many of these solution vendors will either be acquired by one of the market leaders for their technical innovations or fail outright because of the lack of sustainable market share or sufficient financial backing. We are in the midst of this consolidation now, while other areas of the data center—networking and storage, for example—are beginning to accelerate their virtualization journeys. Whatever happens, it is an exciting time to be watching the birth, growth, and maturation of a significant technology that has changed and continues to change the IT industry.