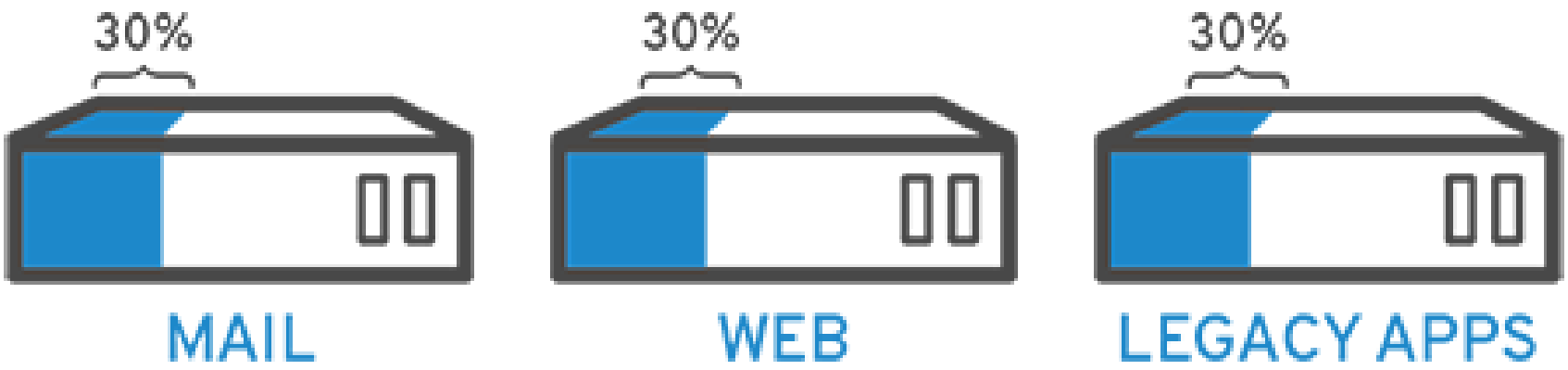
About Hypervisors and Virtualization

Musa Sadık Ünal

2021-02-16

Virtualization

Virtualization enables you to use the full power of a physical computer by distributing its resources to multiple users or environments. Let's give an example about virtualization. There are three different tasks we want to do which run on separate servers. However, this will be both costly and more difficult to control. Thanks to virtualization, we can do these things on a single machine without affecting each other.



Virtual vs Traditional Systems

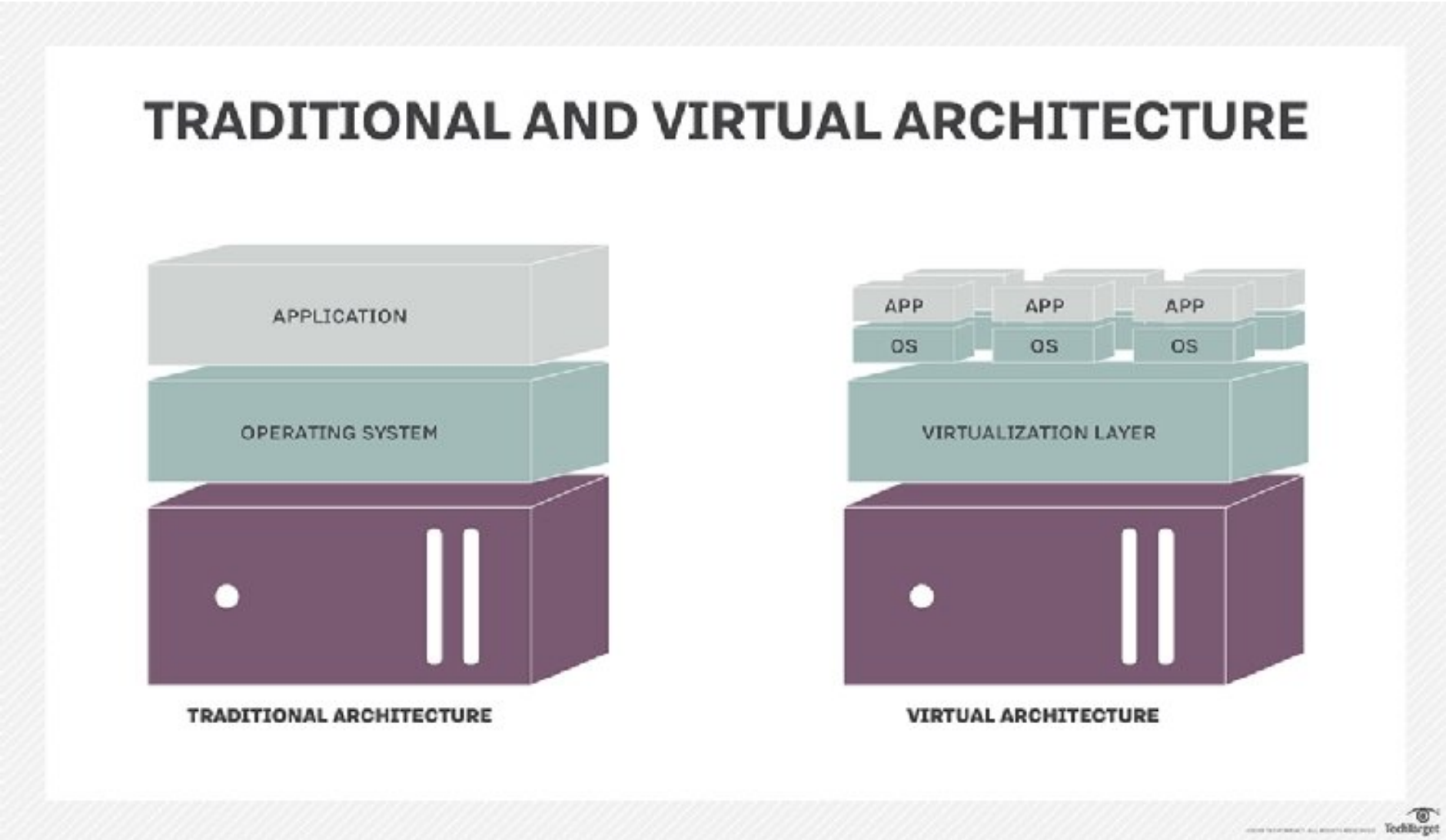
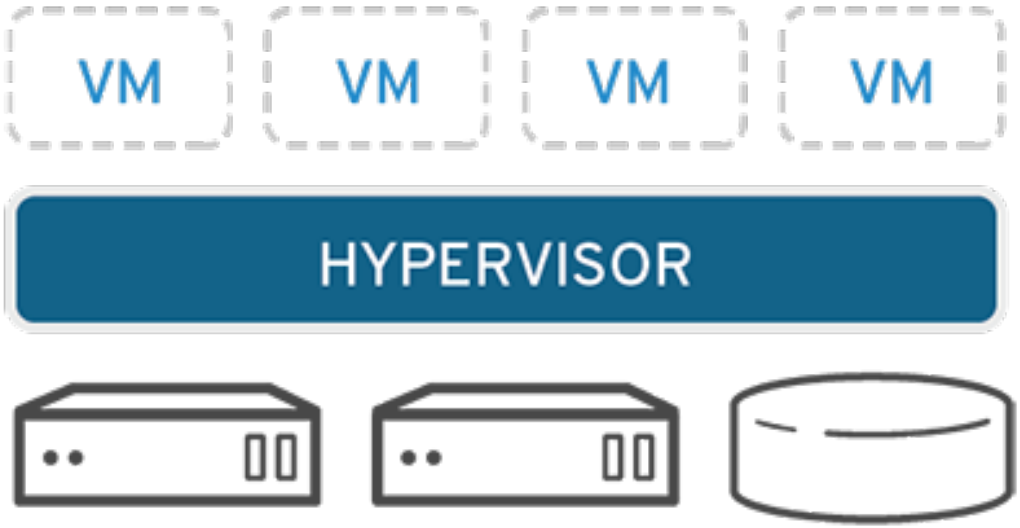


Figure 1: Architectural comparison of Virtual and Traditional Systems

How Does Virtualization Work?

Thanks to hypervisors, it is possible to separate physical resources from virtual environments. Hypervisors take the resources in the system and distribute them to these virtual environments.



Each virtual machine formed here actually becomes a data file. It is also possible to take these files and open and run them on other systems.

What is Hypervisor?

Also known as Virtual Machine Monitor(VMM) or virtualizer which creates and runs virtual machines. A computer or system which runs one or more virtual machines called as **host** machine. Each virtual machine is called **guest** machine.

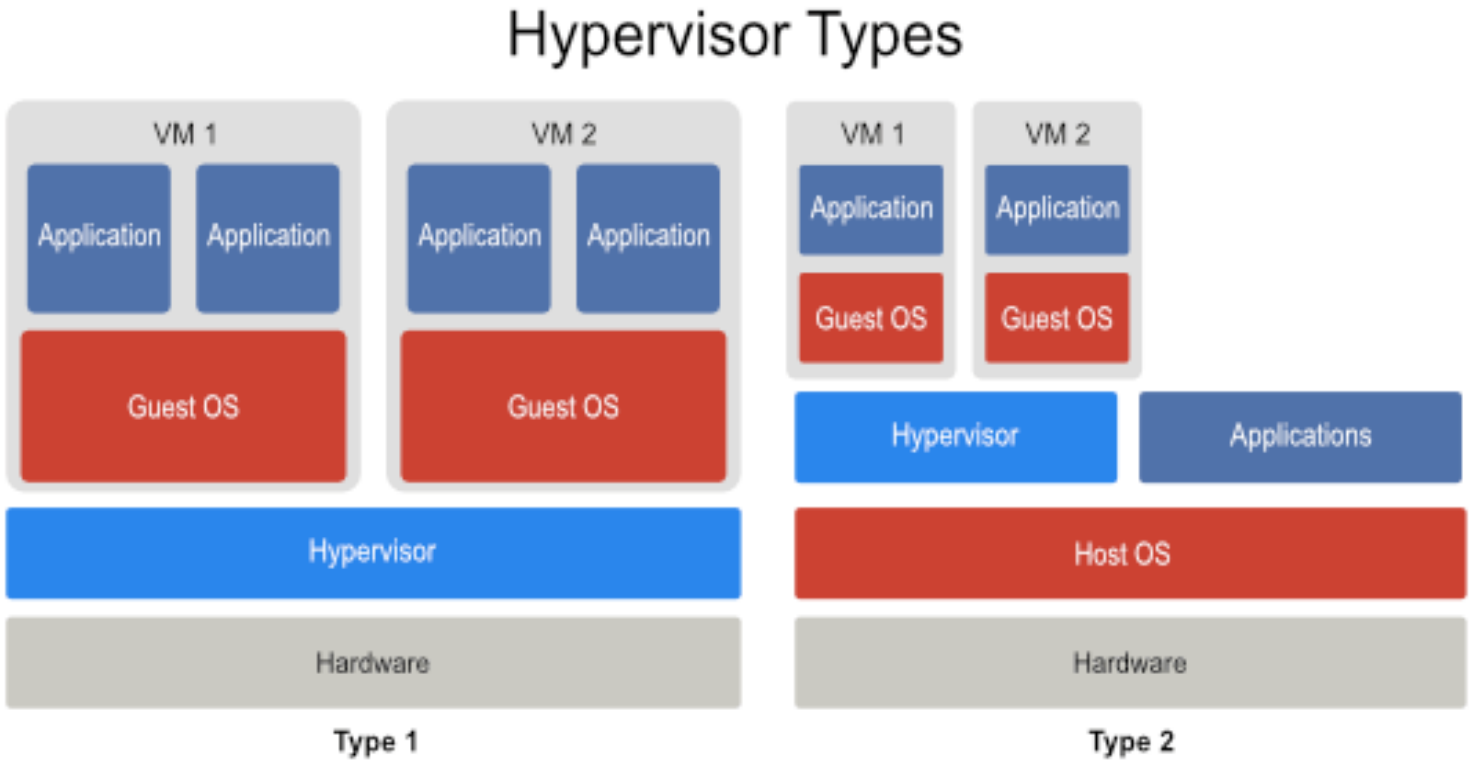


Figure 2: Architectural comparison of Type 1 and Type 2 hypervisors

Types of Hypervisors

Type 1- Bare metal or native hypervisors.

These type of hypervisors can run directly with host computer hardware and can control guest operating systems. More secure than type 2.

- Xen
- Microsoft Hyper-V
- Xbox One

Type 2 - Embedded or "hosted" hypervisors

This is like an usual computer program which requires an OS to run. So to run a type-2 hypervisor we need an host OS.

- Virtual Box
- Qemu
- VmWare

Hypervisor to Virtualization

Hypervisors can create different types of virtualization. There are two type of virtualization which can be created by hypervisor. These are:

1. Full Virtualization
    1. Software Assisted Full Virtualization
    2. Hardware Assisted Full Virtualization
2. Paravirtualization

Full Virtualization

In full virtualization guest operating system resources will not be modified. Guest's doesn't know they have been virtualized. There is no need to base OS. There are two types of full virtualization.

Software Assisted Full Virtualization

In order to trap and virtualize the execution of sensitive, non-virtualizable instruction sets, it totally relies on binary translation. It emulates the hardware using instruction sets from the program. It is also criticized for performance issues due to binary translation.

Hardware Assisted Full Virtualization

Is also called as native virtualization. Hardware-assisted full virtualization removes binary translation and uses virtualization technology to interrupt hardware directly. It becomes more efficient in terms of performance due to not having binary translation.

Paravirtualization

In paravirtualization guests know that it has been virtualized. Guest's source codes will be changed to communicate with the host in this virtualization method. Paravirtualization system uses hypercalls to communicate. To make API calls to the hypervisor, Guest OS needs extensions. In full virtualization, guests will make a hardware call, but guests will communicate directly with the hypervisor using the drivers in paravirtualization. In most cases paravirtualization has higher performances.

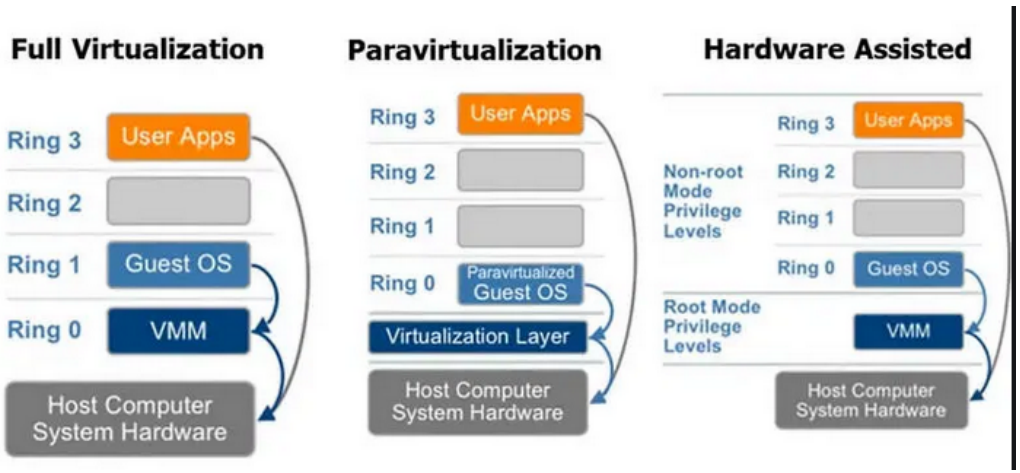


Figure 3: Architectural Comparison

Benefits of Using Hypervisor

- **Speed:** With hypervisors virtual machines can be created instead.
- **Efficiency:** Hypervisor let us to create VM's in single physical machine. Using a single machine is both cheaper and more easier to operate.
- **Portability:** With hypervisors we can create different operating systems in a single host computer.
- **Security:** Problems that may occur as a result of a guest OS corruption will not affect other OS's. In case of a security vulnerability that may occur in this way, other OS will not be affected.

Use Case - Security

- Qubes OS is a **Xen-based** virtualization to allow for the creation and management of isolated vm's.

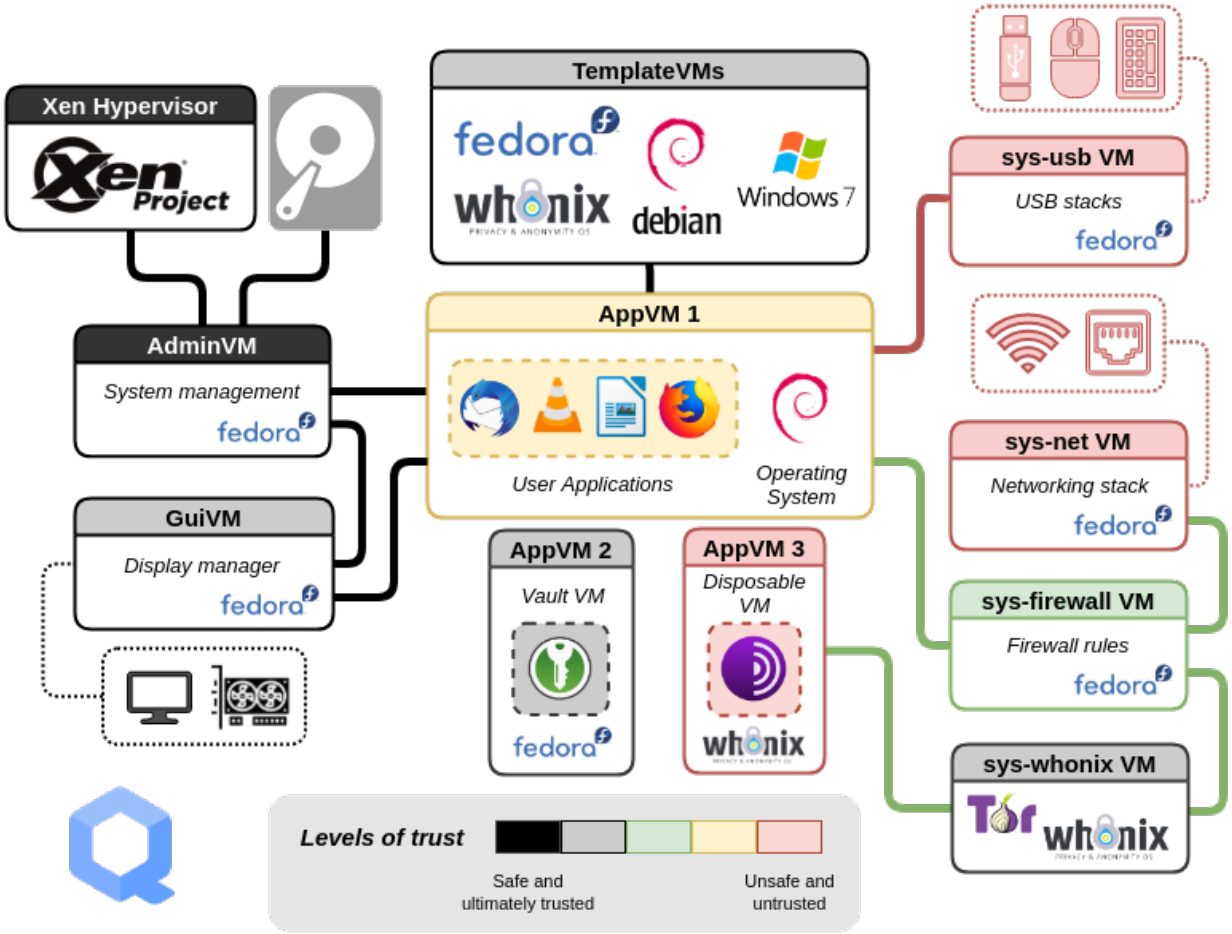


Figure 4: Qubes OS

Important Hypervisors

- XEN => Stable, Type 1 hypervisor
- KVM => Turns Linux Kernel to Type 1 hypervisor

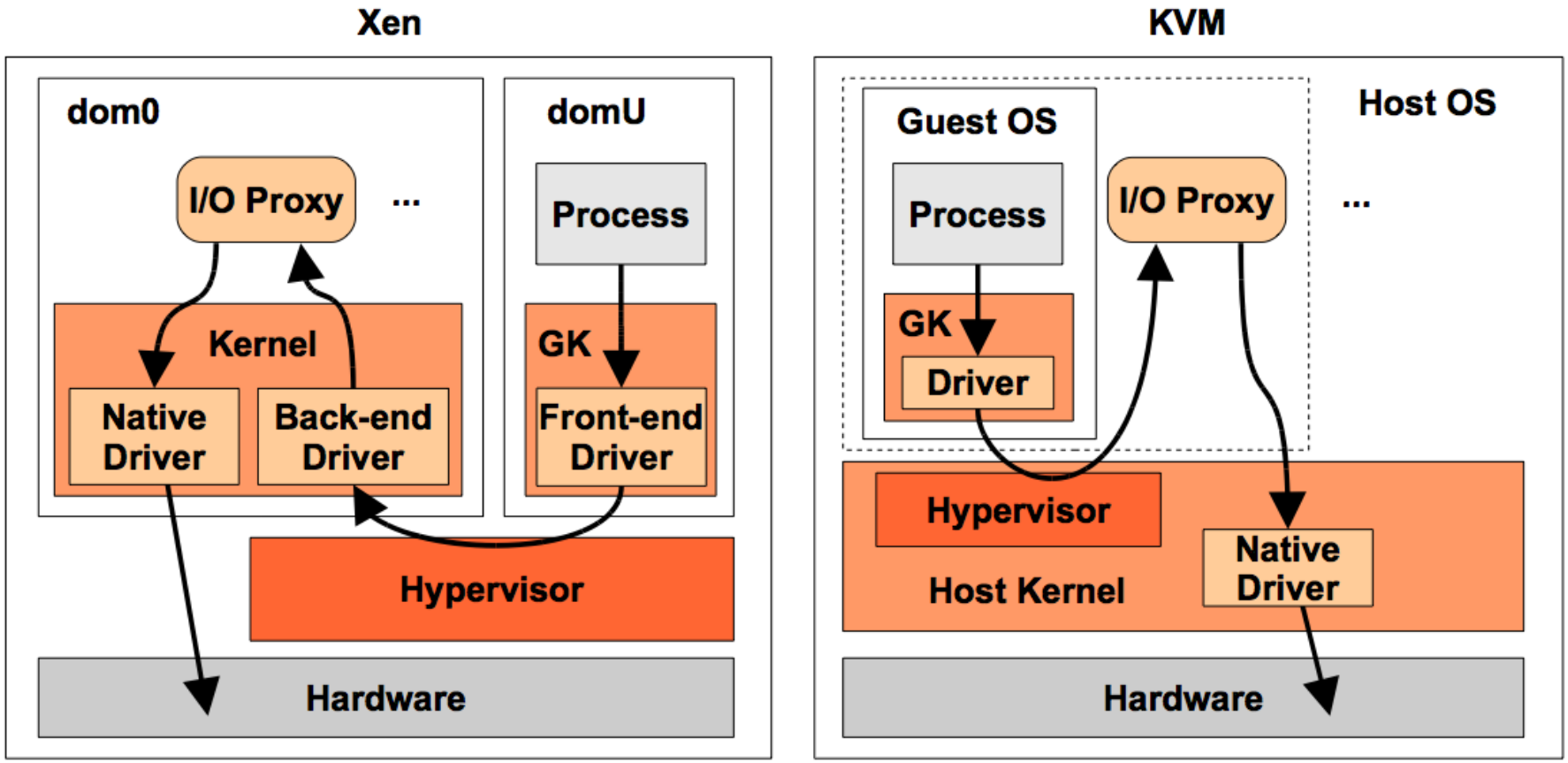


Figure 5: Architectural Comparison

What is KVM (Kernel-based Virtual Machine)?

KVM is a module that convert Linux kernel to type-1 hypervisor. It's designed for x86 processors. KVM requires special hardware such as Intel VT-x or AMD-V to virtualize.

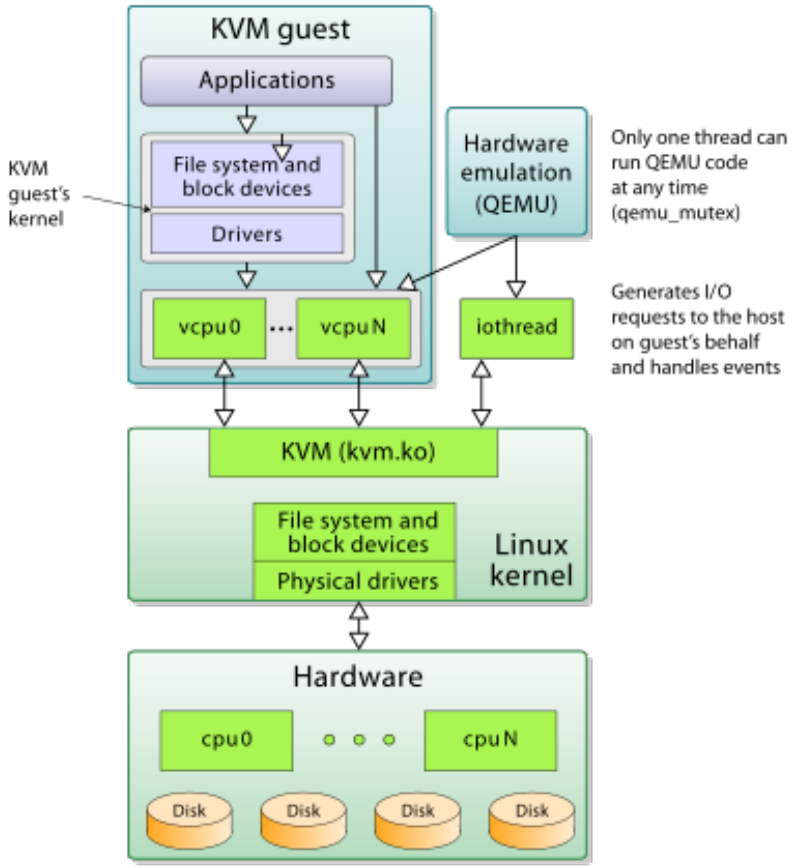


Figure 6: KVM Architecture

Container vs Hypervisor

- "The container's system requires an underlying operating system that provides the basic services to all of the containerized applications using virtual-memory support for isolation."

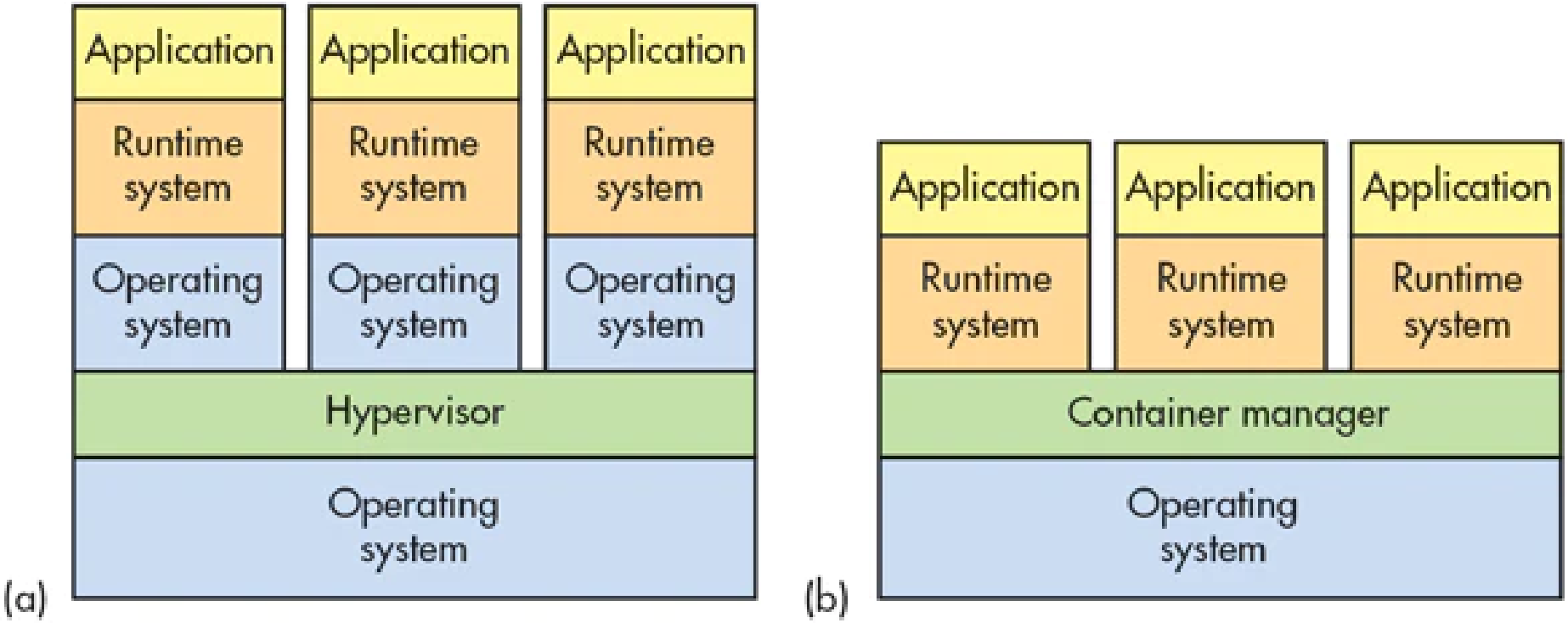


Figure 7: Architectural Comparison Between Containers and Hypervisors

Embedded Virtualization

There are two widely used virtualization technique in embedded systems.

- Full Hypervisor
- Static Partition

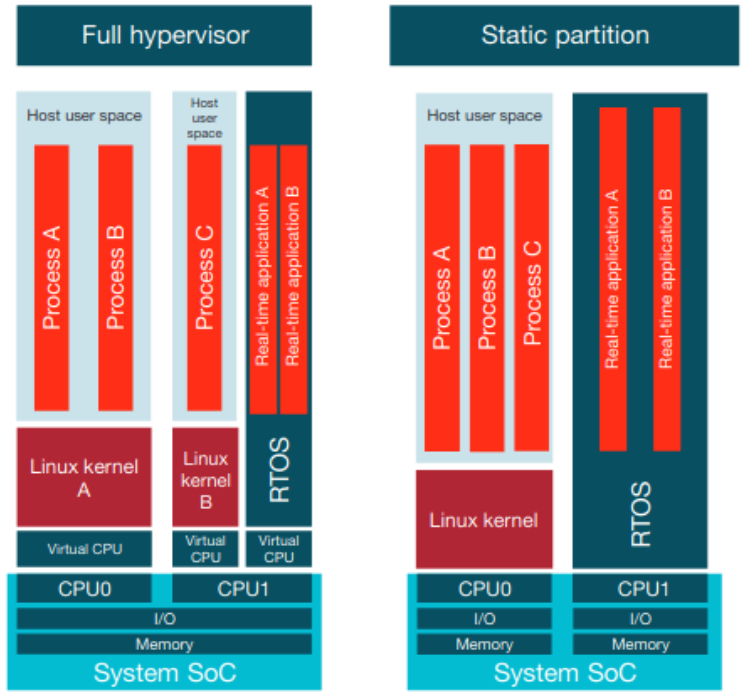


Figure 8: Comparison of the two main types of virtualization

Using Static Partitioning in Embedded Systems

- Static partitioning trades off flexibility of shared memory and I/O for some guarantees of determinism.
- Separating mixed-critically task.
- Timing of the tasks are less affected because physical resources are also divided.

Commercial Embbeded Virtualizers

- WindRiver WxWorks
- LynuxWorks LynxOs
- Jailhouse
- GreenHills Integrity

Keywords

- Hypercall: hypercall is to a syscall what a hypervisor is to an OS.
- Binary Translation: is a type of virtualization technique. It is used to eliminate the problem of different hardwares.
- Dom0: first domain started by the Xen hypervisor on boot.

Resources

0. https://www.redhat.com/en/topics/virtualization/what-is-virtualization
1. https://www.redhat.com/en/topics/virtualization/what-is-KVM
2. https://www.parallels.com/blogs/ras/hypervisor/
3. https://onapp.com/2016/09/06/hypervisor-choice-xen-or-kvm/
4. https://www.sciencedirect.com/topics/computer-science/assisted-virtualization
5. https://www.unixarena.com/2017/12/para-virtualization-full-virtualization-hardware-assisted-virtualization.html/