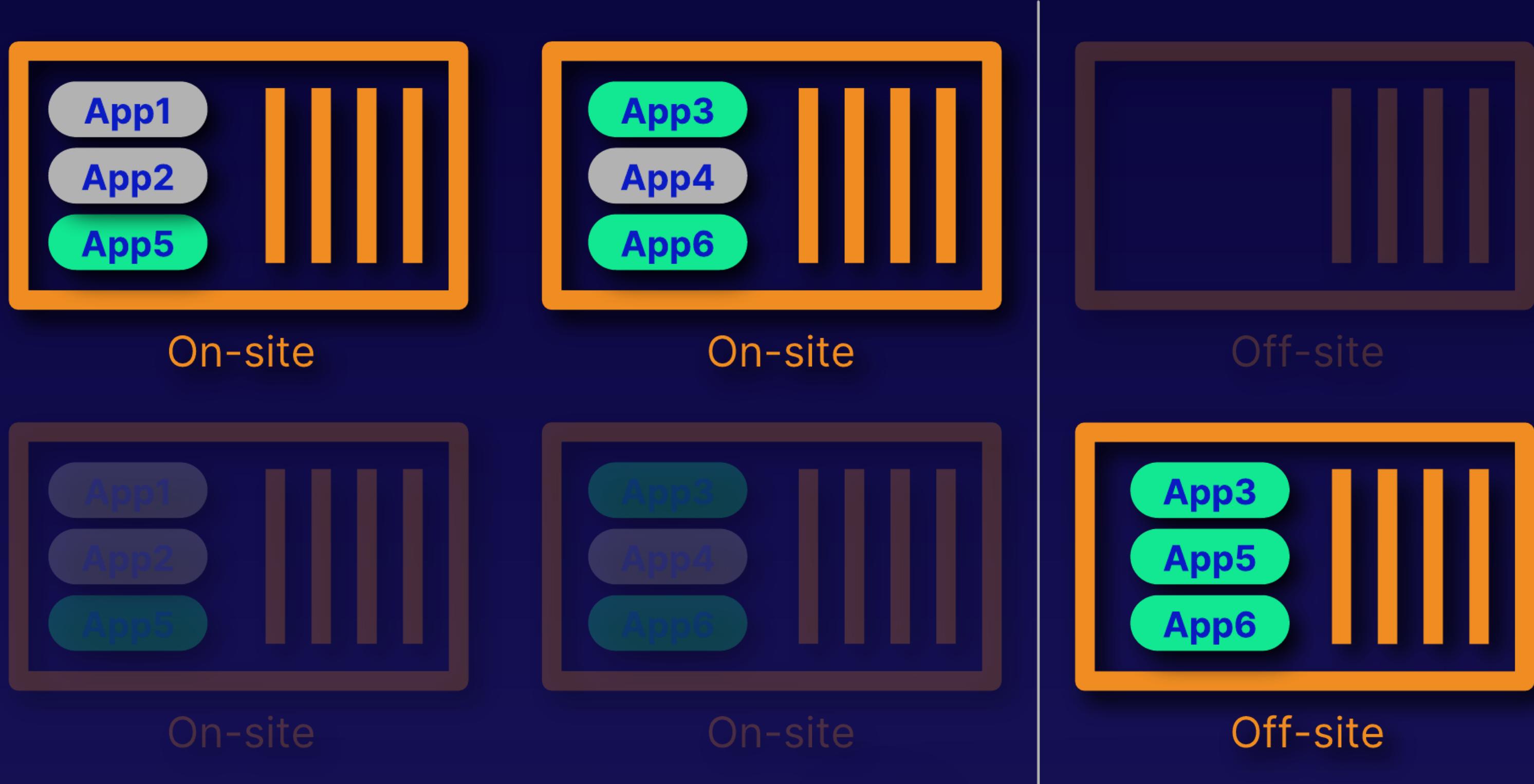




The ability to take snapshots of guests prior to patching, application upgrades, or other **potentially dangerous processes** provides a safety net and a quick mechanism to back out changes that might have been *less than successful*.

Disaster Recovery/ Migrations



An enormous benefit of virtualization is the ***ease with which guests can move between hosts***. This is not only possible within a *single* data center but *across* data centers and ***in the cloud***. Many enterprise-level solutions offer disaster recovery solutions so you don't have to reinvent the wheel.

This same concept applies when it comes to migrating guests to *new hardware* or a *new data center*.

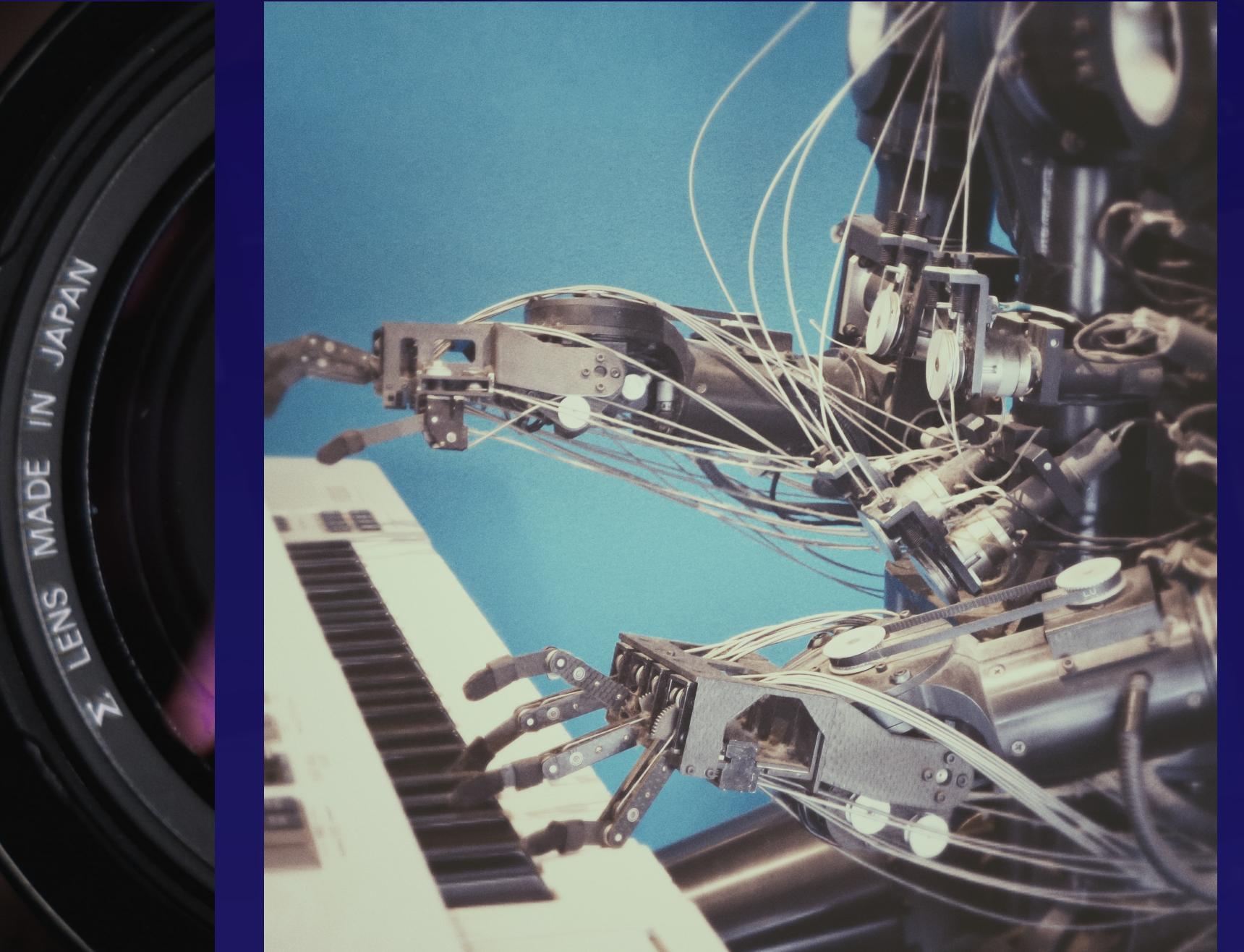
Backups



With the ability to take snapshots of virtual disks comes the ability to back up these snapshots.

Since this happens outside of the guest operating system, we can use a **system-agnostic backup methodology**. We can also leverage advanced backup and recovery options, where available, that were not available or were difficult to implement with traditional hardware-based infrastructures.

Standardization/ Automation

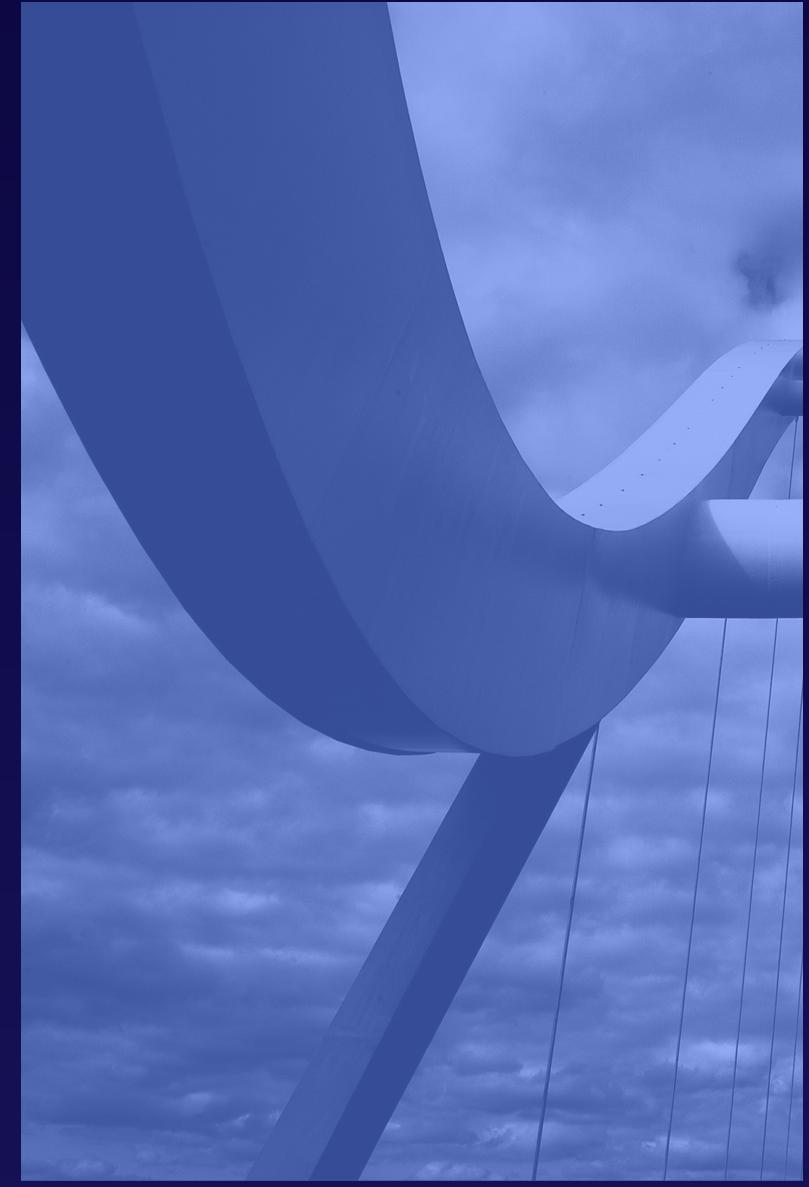


In addition to moving to a homogeneous hardware platform, virtualization provides a clean, predictable environment for guests. This standardization reduces complexity in the data center stack.

A good **virtualization platform** gives you the tools and automation to ease management.



In this lesson, we covered a number of good reasons to virtualize all or part of your physical workloads:



- Reduce Your **Hardware Costs**
- Reduce Your **Energy Costs**
- Reduce Your **Footprint**
- Workload Management/Fault Tolerance/Maintenance
- Disaster Recovery/Migrations
- Standardization/Automation



Tom Dean
Linux Training Architect

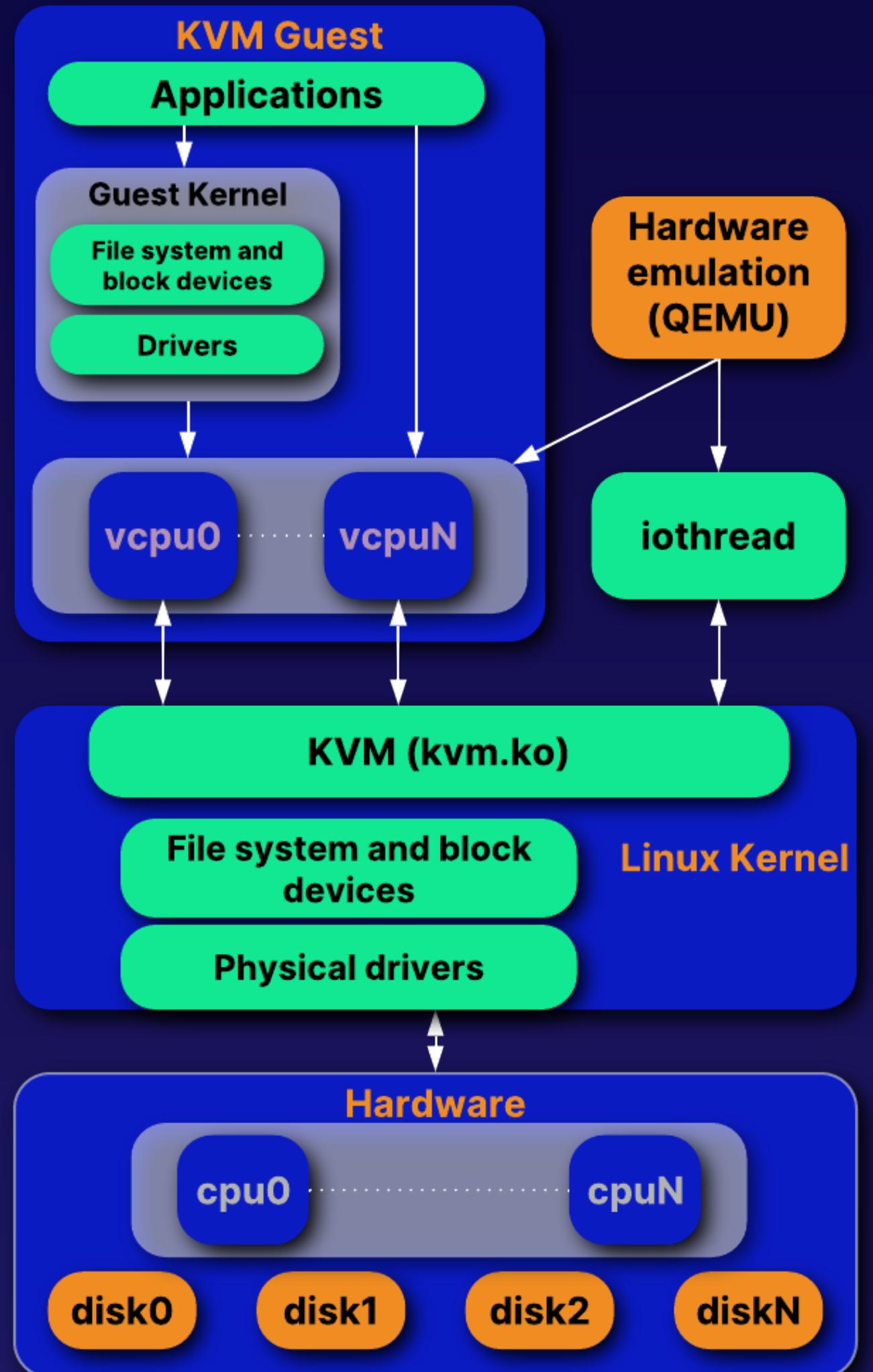
KVM - Overview

What Is KVM?

KVM

KVM: Linux Kernel Module

- Allows the kernel to function as a **type-1** hypervisor
- Was merged into the kernel mainline as of **2.6.20** (in February 2007)
- Provides the **/dev/kvm** interface
- Requires **hardware virtualization extensions**
 - **Intel VT-x**
 - **vmx** extension
 - **AMD-V**
 - **svm** extension



KVM/QEMU/libvirt



KVM

Provides acceleration through access to hardware virtualization extensions



QEMU

Type-2 hypervisor that utilizes KVM for type-1 hypervisor acceleration



libvirt

Toolkit to manage virtualization platforms, like QEMU/KVM

If Linux
virtualization
were a
Corvette...



KVM would be the
powerful engine
under the hood.



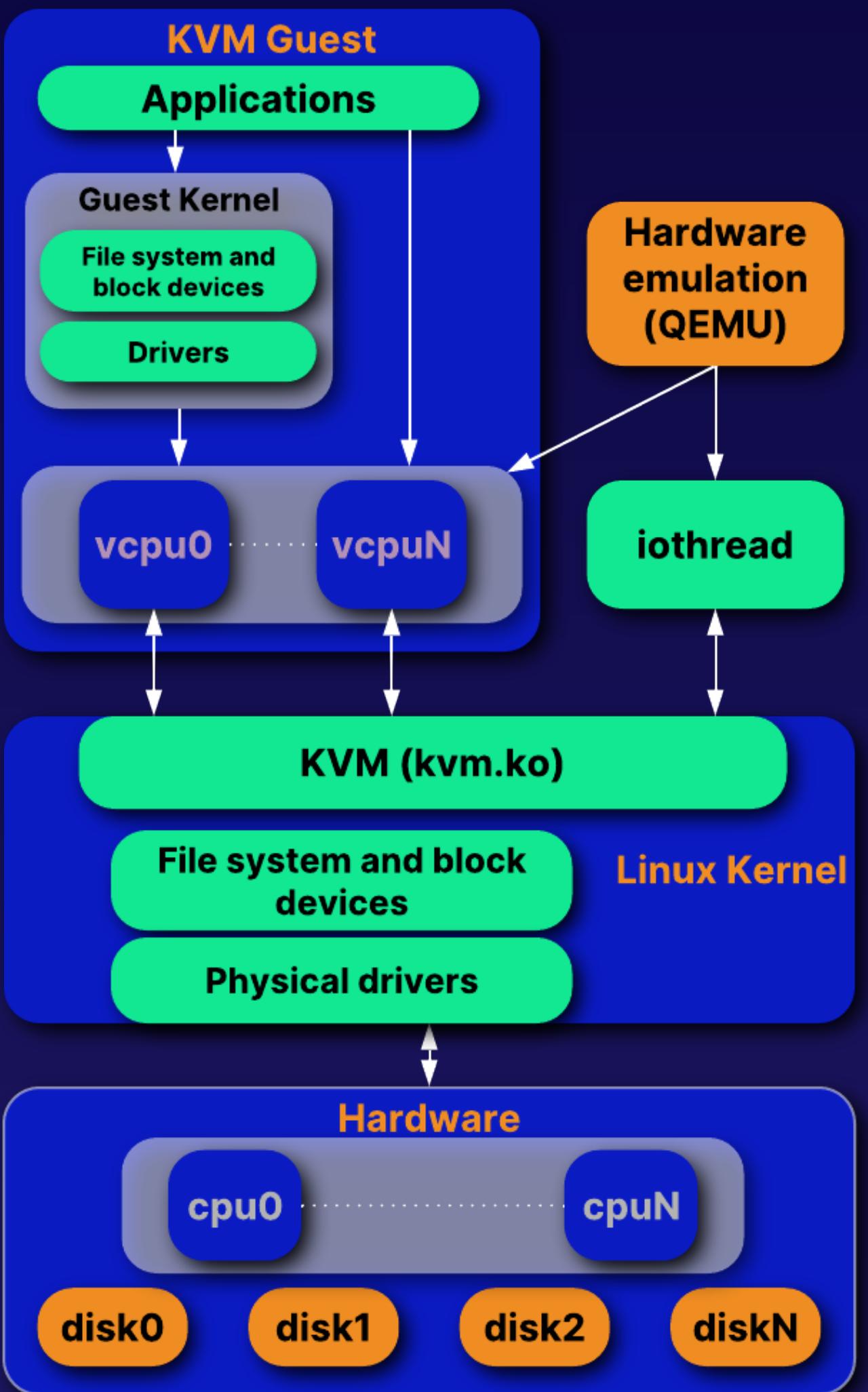
How Does It Work?

“**KVM** (for Kernel-based Virtual Machine) is a full virtualization solution for Linux on **x86** hardware containing virtualization extensions (**Intel VT** or **AMD-V**). It consists of a loadable kernel module, **kvm.ko**, that provides the core virtualization infrastructure and a processor-specific module, **kvm-intel.ko** or **kvm-amd.ko**.

“Using **KVM**, one can run multiple virtual machines running unmodified Linux or Windows images. Each virtual machine has private virtualized hardware: a network card, disk, graphics adapter, etc.

“**KVM** is open source software. The kernel component of **KVM** is included in mainline Linux, as of **2.6.20**. The userspace component of **KVM** is included in mainline **QEMU**, as of **1.3**.”

— **KVM Project**



The `/dev/kvm` Device



In this
lesson, we
covered:



- What **KVM** is
- How **QEMU / KVM** works
- The **/dev/kvm** interface
- How to determine if a system supports **KVM** virtualization



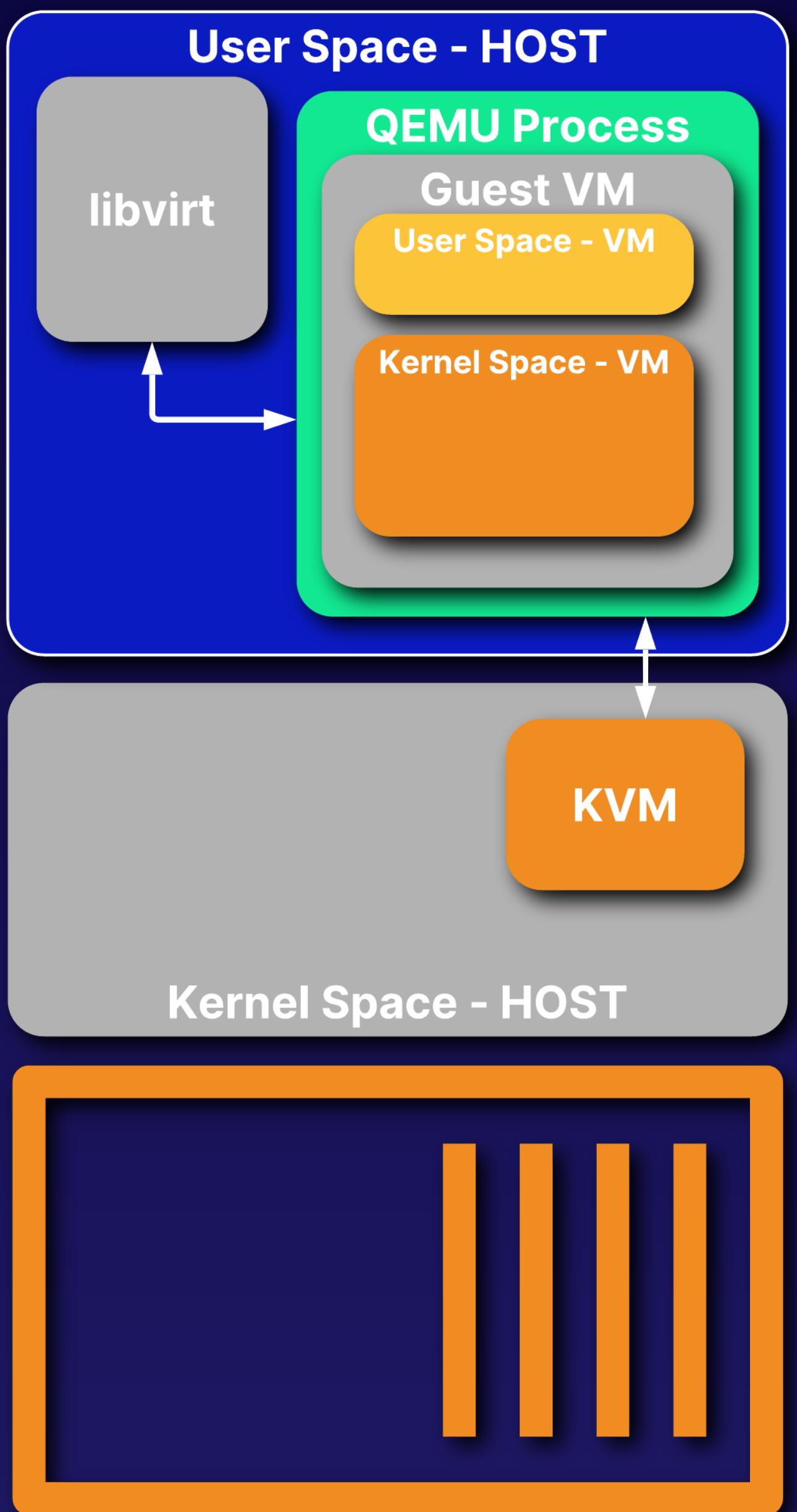
Tom Dean
Linux Training Architect

QEMU - Overview

What Is QEMU?

QEMU

QEMU is a **type-2** hypervisor that runs in **user space**. When used with **KVM**, it accelerates the performance of a **QEMU** guest and the combination becomes a **type-1** hypervisor.



KVM/QEMU/libvirt



KVM

Provides acceleration through access to hardware virtualization extensions



QEMU

Type-2 hypervisor that utilizes KVM for type-1 hypervisor acceleration



libvirt

Toolkit to manage virtualization platforms, like QEMU/KVM

If Linux
virtualization
were a
Corvette...



QEMU would be the chassis and body that channels the power of that engine.



How Does It Work?

QEMU

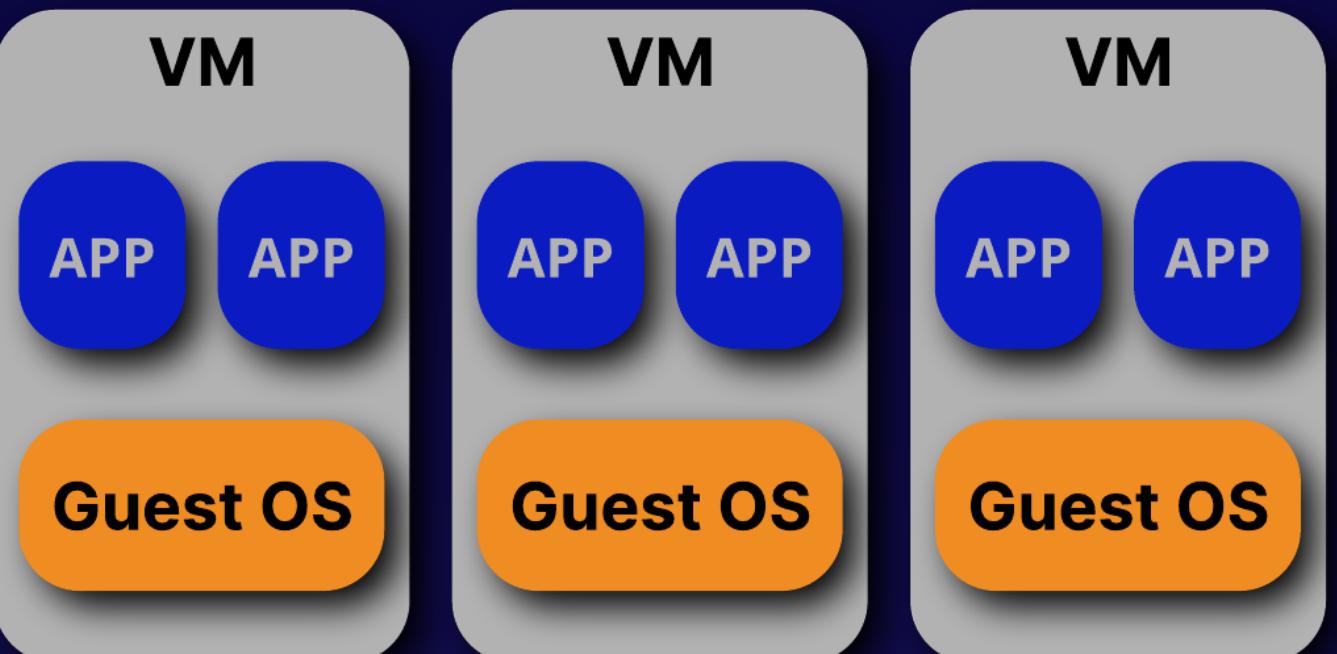
QEMU is a **user space** program that has two modes:

Emulation (software)

- Type-2 hypervisor

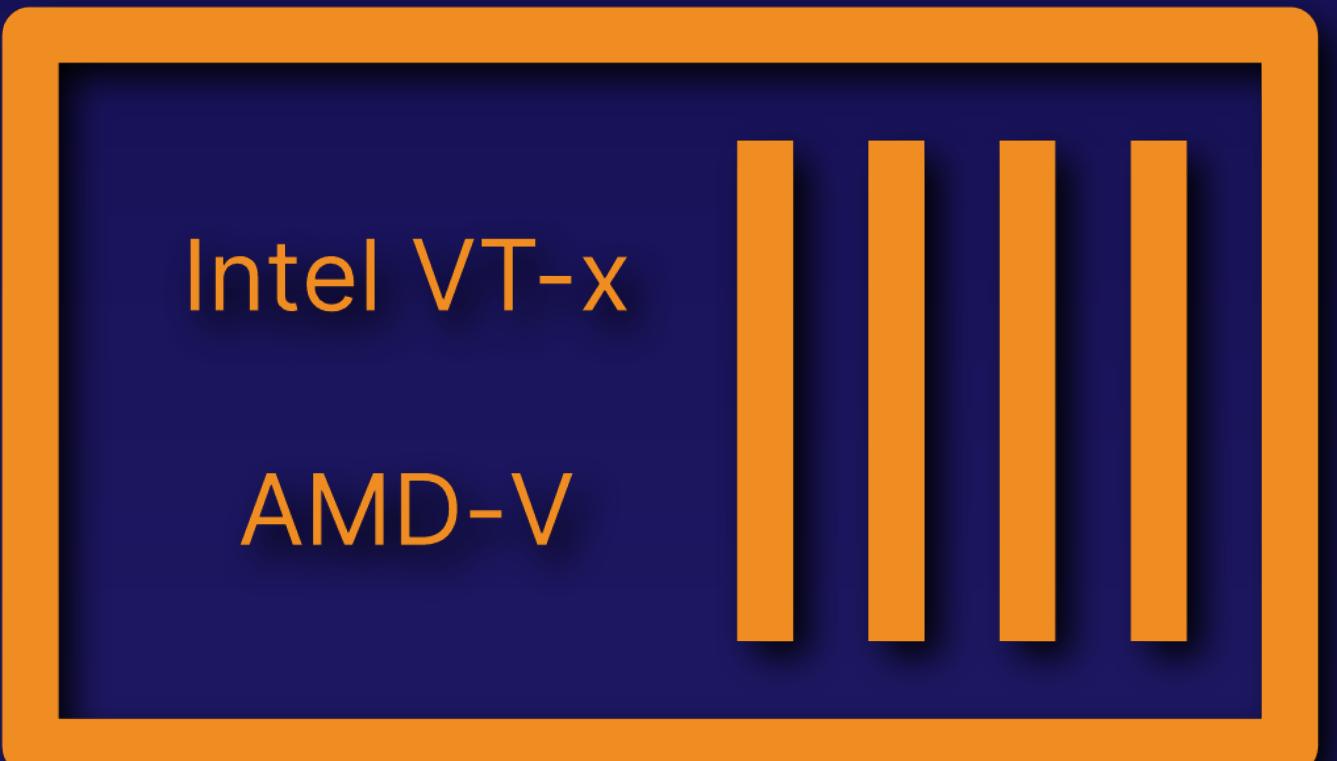
Virtualization (software and hardware)

- Type-1 hypervisor



User Space/QEMU

Kernel/KVM





QEMU creates one process for every **VM** and a thread for each **vCPU**.

These run in the Linux scheduler.

A **virtual machine** is a collection of processes running on the host.

QEMU - Devices

QEMU and the `/dev/kvm` interface combine to provide the resources the guest operating system requires:

Key hardware:

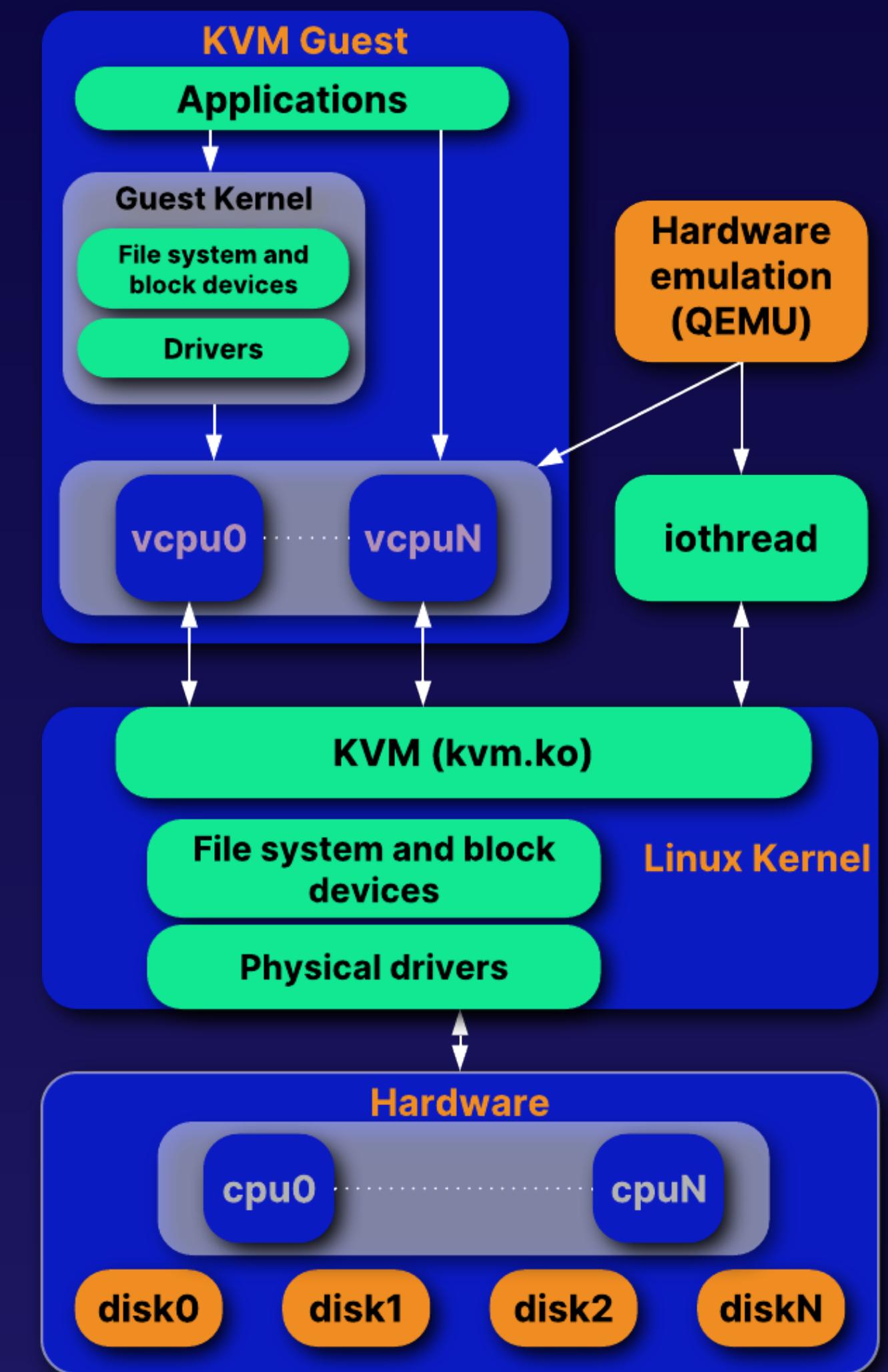
- Virtual CPU(s)
- Virtual memory
- Virtual disk(s)
- Virtual networking

Other resources:

- USB
- Sound
- Virtual media:
 - CD/DVD/ISO images

Mapped physical resources:

- Networking:
 - IP
 - Storage
 - GPU(s)



QEMU and the `/dev/kvm` interface combine to provide the resources the guest operating system requires:

Key hardware:

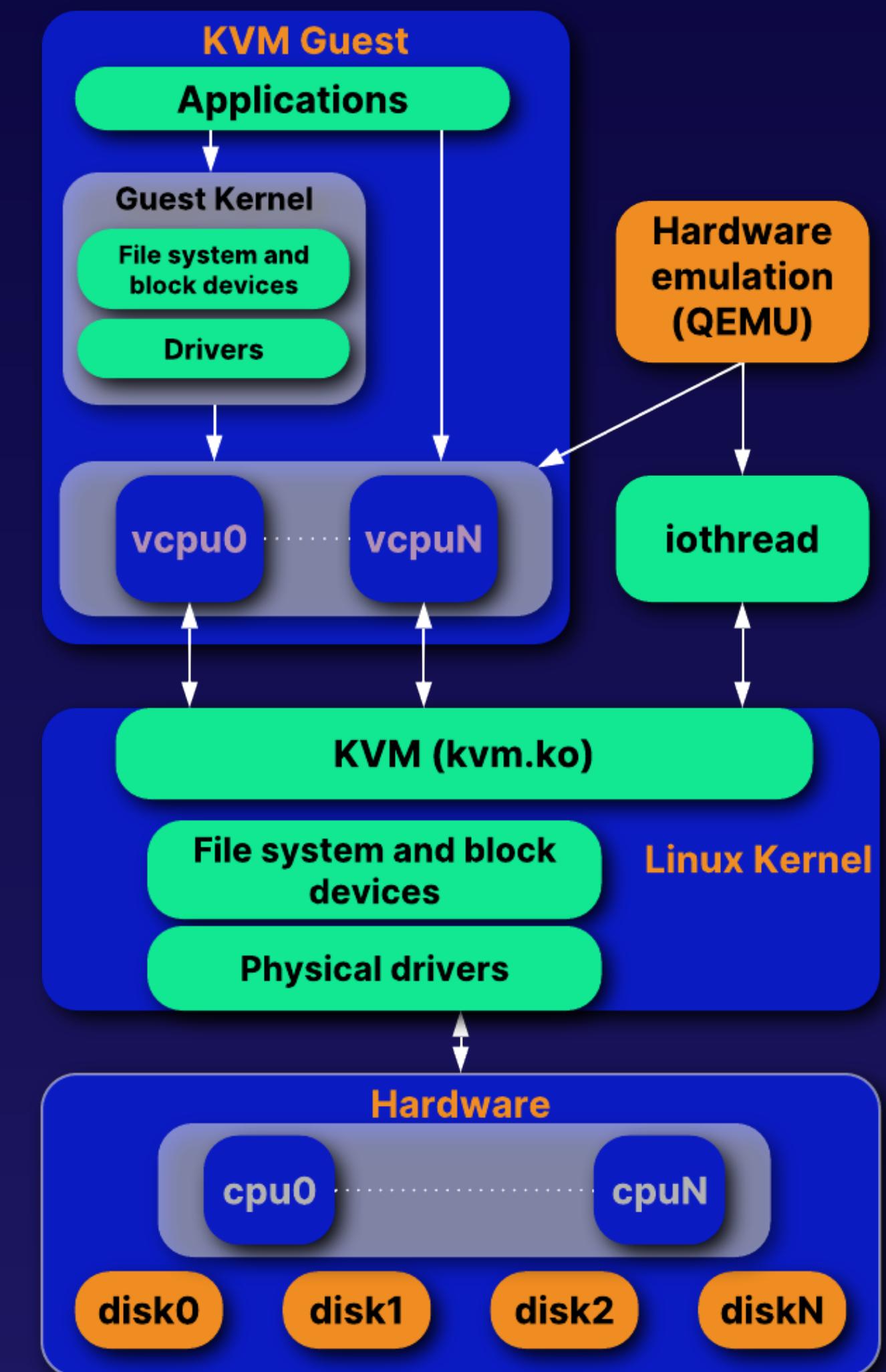
- Virtual CPU(s)
- Virtual memory
- Virtual disk(s)
- Virtual networking

Other resources:

- USB
- Sound
- Virtual media:
 - CD/DVD/ISO images

Mapped physical resources:

- Networking:
 - IP
 - Storage
- GPU(s)



QEMU and the `/dev/kvm` interface combine to provide the resources the guest operating system requires:

Key hardware:

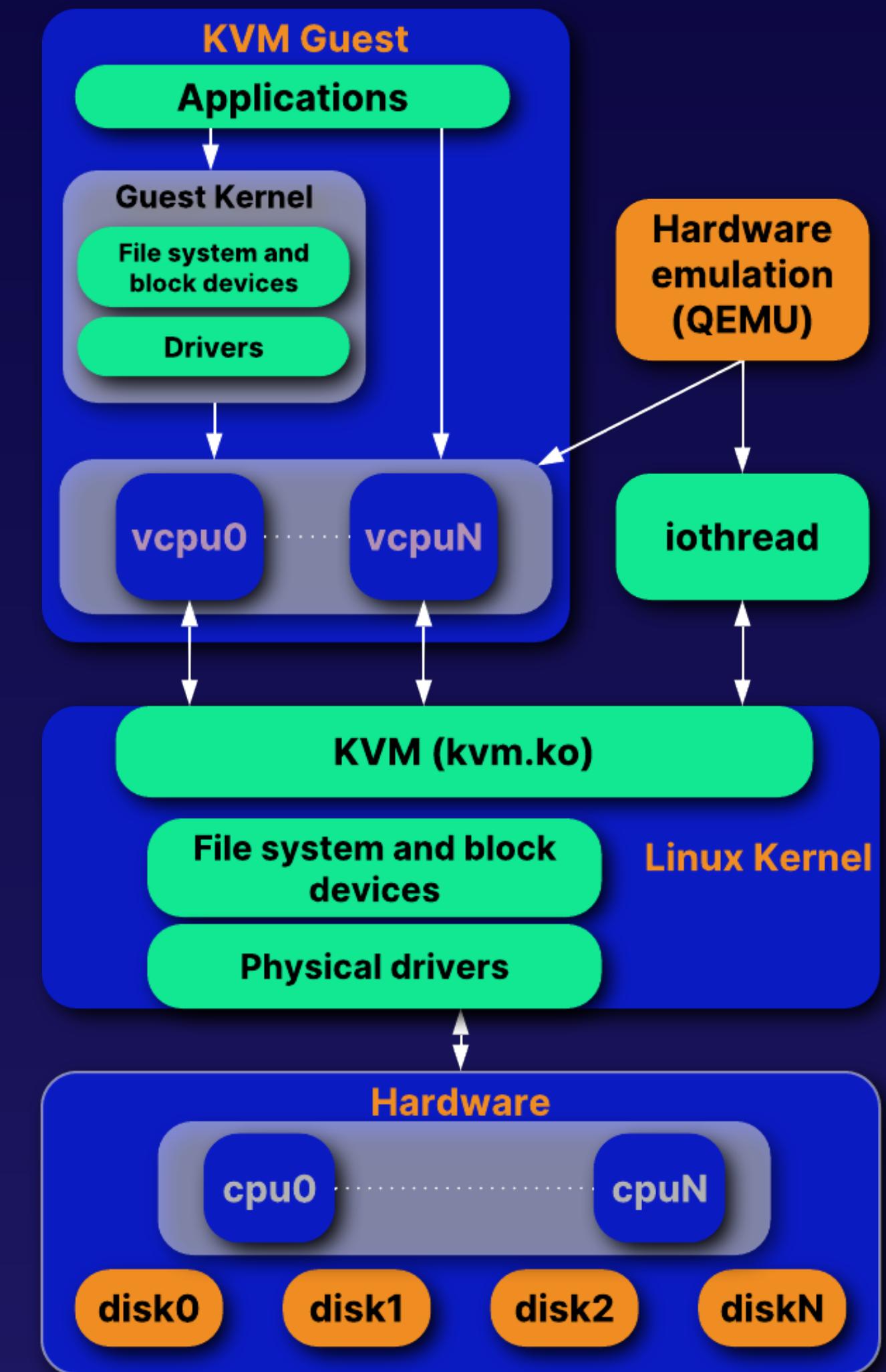
- Virtual CPU(s)
- Virtual memory
- Virtual disk(s)
- Virtual networking

Other resources:

- USB
- Sound
- Virtual media:
 - CD/DVD/ISO images

Mapped physical resources:

- Networking:
 - IP
 - Storage
- GPU(s)

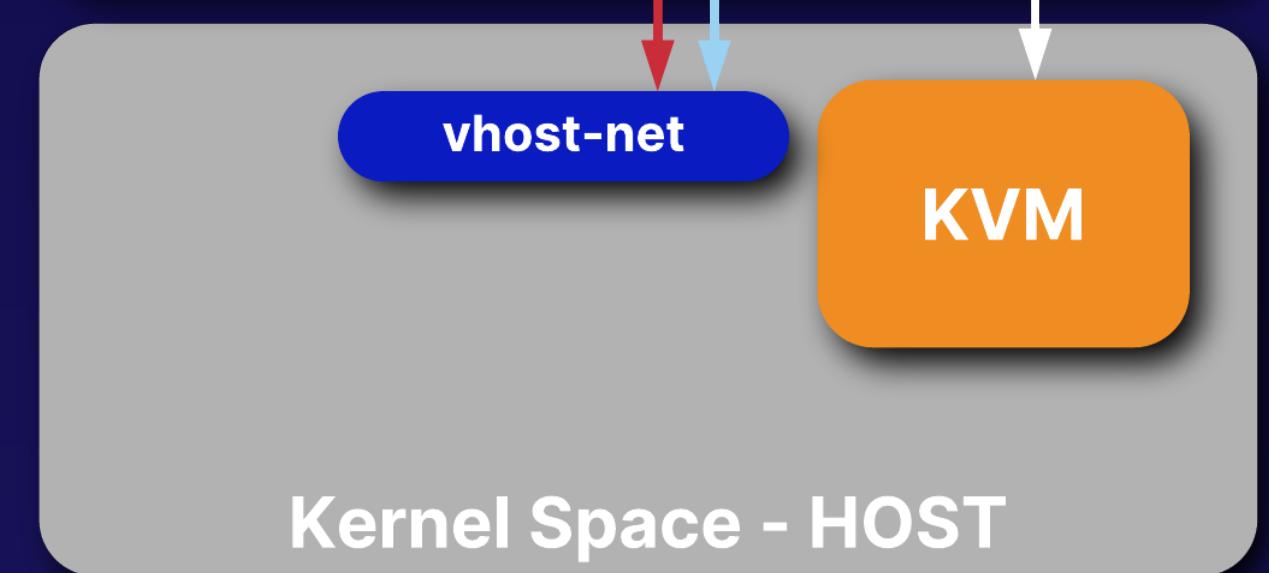
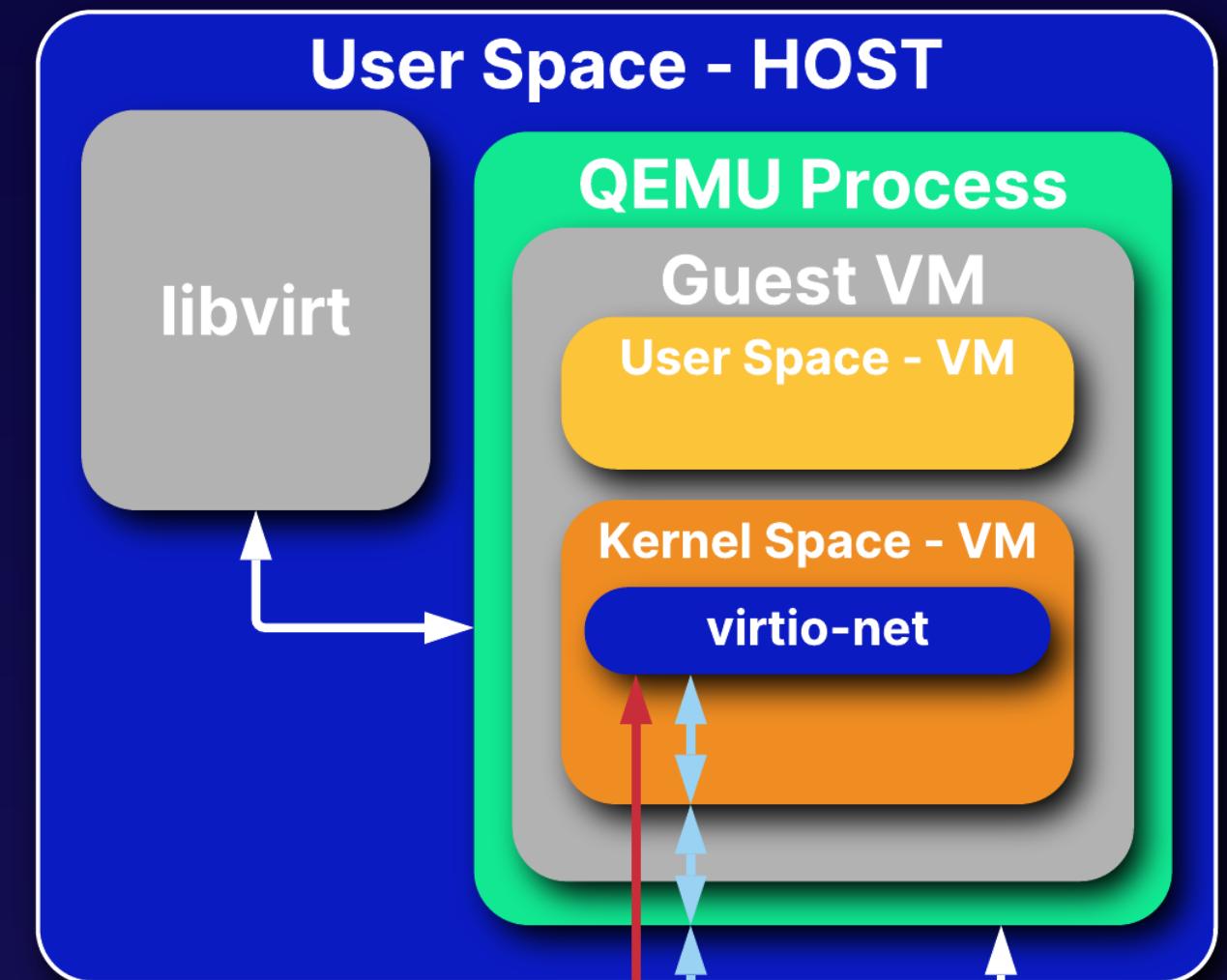


QEMU - Paravirtualization

QEMU/KVM

Provides Paravirtualization Support

- Two parts:
 - Provides a **software** interface that **looks** like hardware
 - **VirtIO** drivers for the guest operating system
 - Offloads some “**challenging**” processes from the **guest** to the **host**
- Uses the **VirtIO API**
 - Ethernet
 - Storage
 - Memory balloon device
 - Display



QEMU - Disk Image Formats

DISK IMAGE FORMATS

Read-Write Formats

- **QEMU** — Copy-On-Write (.qcow2, .qed, .qcow, .cow)
- **VirtualBox** — Virtual Disk Image (.vdi)
- **Virtual PC** — Virtual Hard Disk (.vhdx)
- Virtual **VFAT**
 - A virtual drive with a **FAT** filesystem
 - Quick way to share files **between guest** and **host**
- **VMware** — Virtual Machine Disk (.vmdk)
- **Raw** images (.img)
 - These contain **sector-by-sector** disk contents
- **Bootable** CD/DVD Images (.iso)



DISK IMAGE FORMATS

Read-Only Formats

- **macOS** — Universal Disk Image **Format** (.dmg)
- Bochs
- Linux **loop**
 - Compressed image format
- **Parallels** disk image (.hdd, .hds)

QEMU - Emulation Support

QEMU

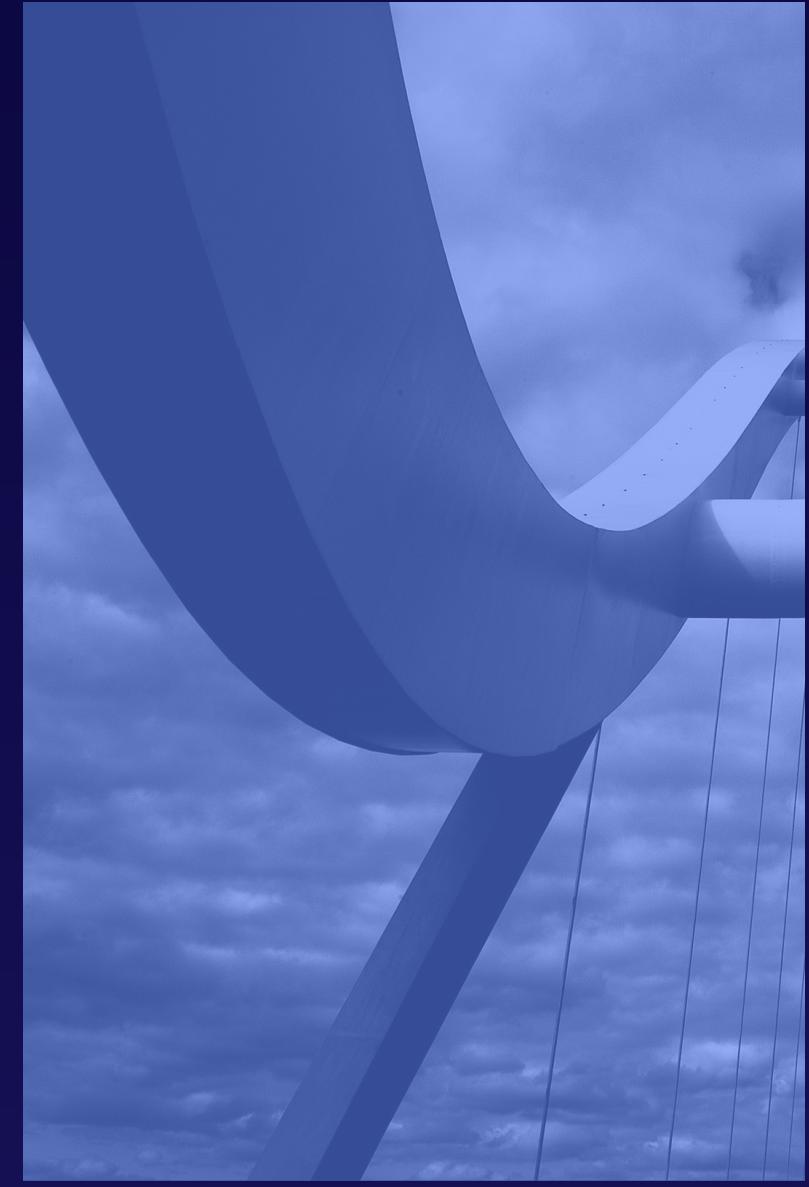
QEMU Emulation Support

- IA-32 (x86) PCs
- x86-64 PCs
- MIPS64 Release 6 and earlier variants
- Sun's SPARC sun4m
- Sun's SPARC sun4u
- ARM development boards (Integrator/CP and Versatile/PB)
- SH4 SHIX board
- PowerPC (PReP and Power Macintosh)
- ETRAX CRIS
- MicroBlaze
- RISC-V





In this
lesson, we
covered:



- What **QEMU** is
- How **QEMU** works
- Devices under **QEMU**
- **Paravirtualization**
- **QEMU** support for virtual disk formats
- **QEMU** emulation support



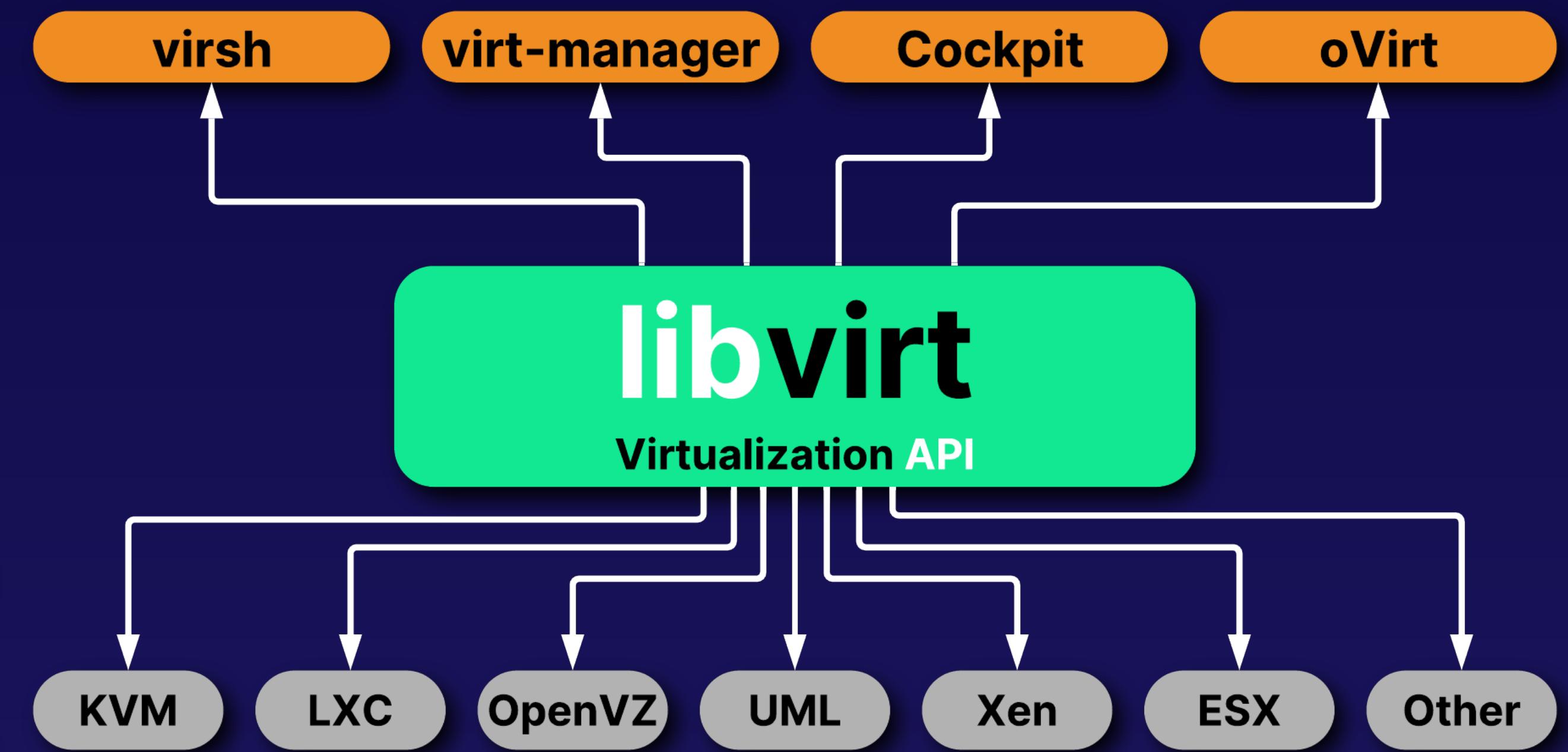
Tom Dean
Linux Training Architect

What Is libvirt?

What's libvirt?

From the **libvirt** project:

- A toolkit to manage **virtualization** platforms
- Accessible from C, Python, Perl, Java, and more
- Licensed under **open-source** licenses
- Supports **KVM**, **QEMU**, **Xen**, **Virtuozzo**, **VMWare ESX**, **LXC**, **bhyve**, and more
- Targets **Linux**, **FreeBSD**, **Windows**, and **OS X**
- Used by many applications

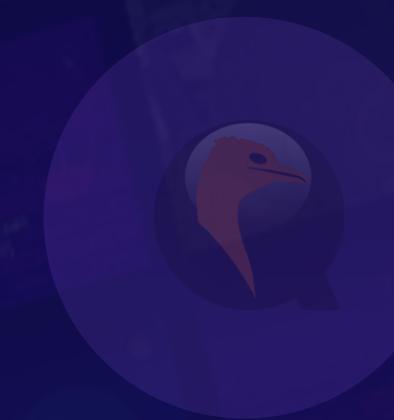


KVM/QEMU/**libvirt**



KVM

Provides acceleration through access to hardware virtualization extensions



QEMU

Type-2 hypervisor that utilizes KVM for type-1 hypervisor acceleration



libvirt

Toolkit to manage virtualization platforms, like QEMU/KVM

If Linux virtualization were a Corvette...



libvirt would be instruments and controls that allow the driver to easily operate the car.



How Does **libvirt** Work?

virsh

virt-manager

Cockpit

oVirt

libvirt

Virtualization API

KVM

LXC

OpenVZ

UML

Xen

ESX

Other

libvirt - Virtualization Platform Support



Virtualization Platform Support

Linux

Windows

macOS

FreeBSD

libvirt - Hypervisor Support

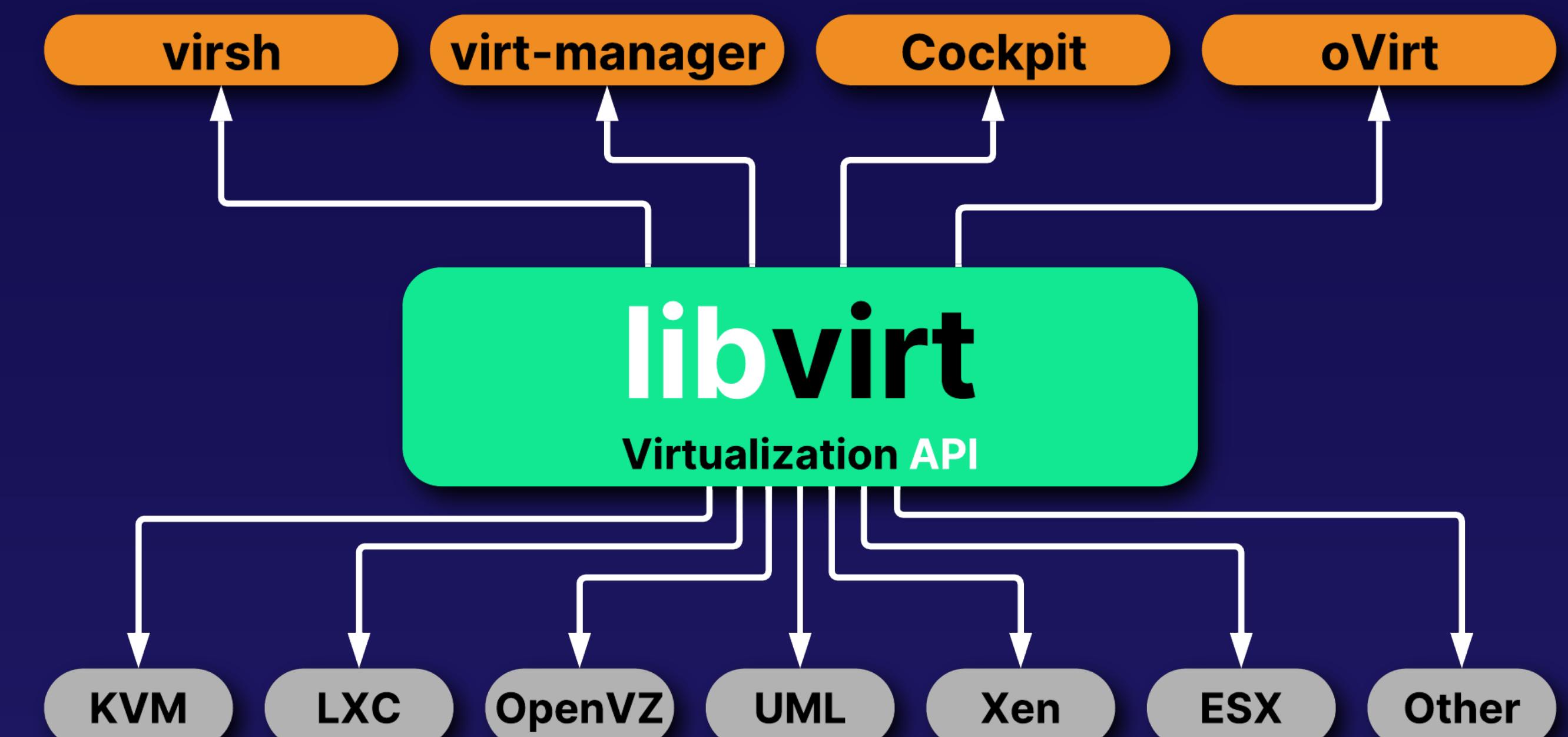


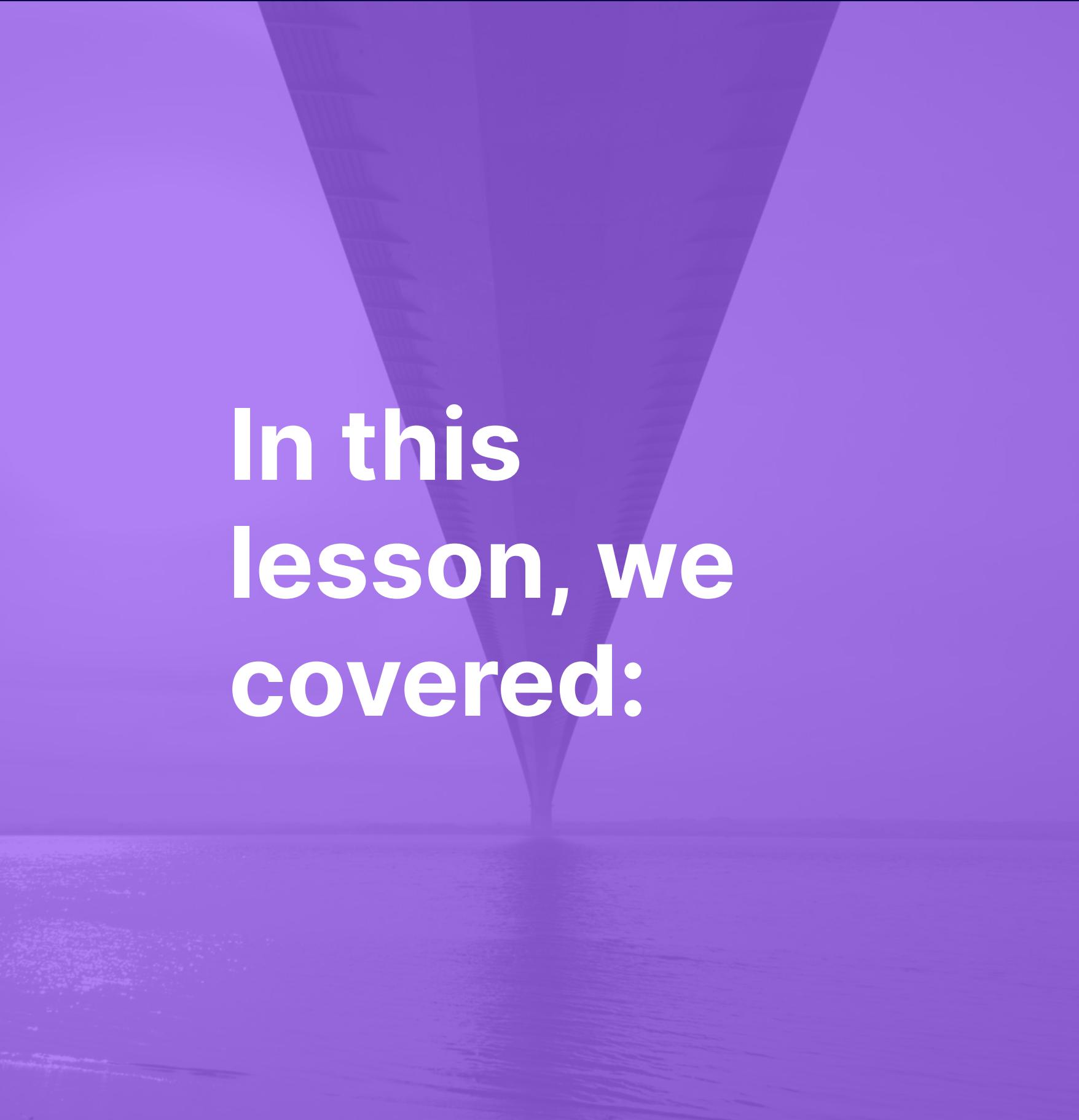
HYPERVISOR SUPPORT

- **LXC**
 - Linux containers
- **OpenVZ**
- **QEMU**
- **Test**
 - Used for testing
- **VirtualBox**
- **VMware ESX**
- **VMware** Workstation Player
- **Xen**
- Microsoft **Hyper-V**
- **Virtuozzo**
- **bhyve** — the BSD hypervisor

We will take a look at these end-user management tools supported by **libvirt** on Linux:

- **Virtual Machine Manager**
 - **virt-manager**
- **Cockpit/Web Console**
 - Via plugin
- **CLI: virsh**
- **oVirt/RHEV**





In this
lesson, we
covered:



- What **libvirt** is
- What function **libvirt** plays in **QEMU/KVM** virtualization
- Which virtualization platforms **libvirt** supports
- Which hypervisor drivers are supported by **libvirt**



Tom Dean
Linux Training Architect

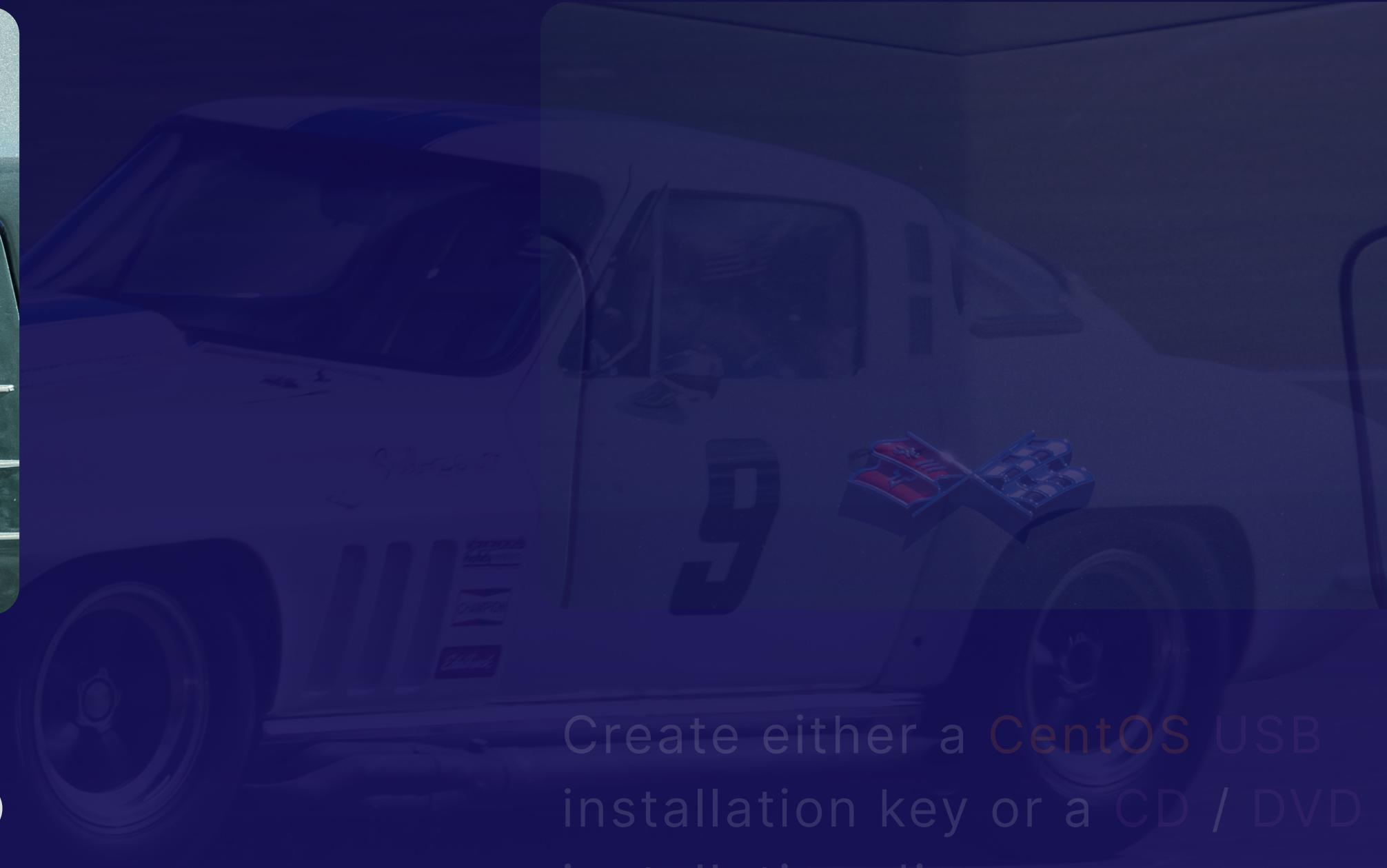
How to Create the **CentOS 8 Workstation** Lesson Environment from Scratch

We'll need two things to start:



Download the **CentOS 8** ISO

- Use the **CentOS 8** Boot ISO
(...-boot.iso)



Create either a **CentOS** USB
installation key or a **CD / DVD**
installation disc.

We'll need two things to start:



Download the CentOS 8 ISO

- Use the [CentOS 8 Boot ISO](#)
(...-boot.iso)



Create either a [CentOS USB](#)
installation key or a [CD/DVD](#)
installation disc.

CentOS 8 Installation

- Configure the **BIOS** to support hardware virtualization.
 - Enable **Intel VT-x/AMD-V**.
- Boot from **USB/optical media**.
- Configure the installation to install the **Workstation** package group.
- Create **cloud_user** account and set the **root** password.





Optional: Install the **xrdp** service for **RDP** access to the **CentOS Workstation**

- Install the **xrdp** service
- Configure the firewall to allow access to the **xrdp** service



We now have a CentOS 8
Workstation installation!

 Tom Dean
Linux Training Architect



Introduction to virt-manager

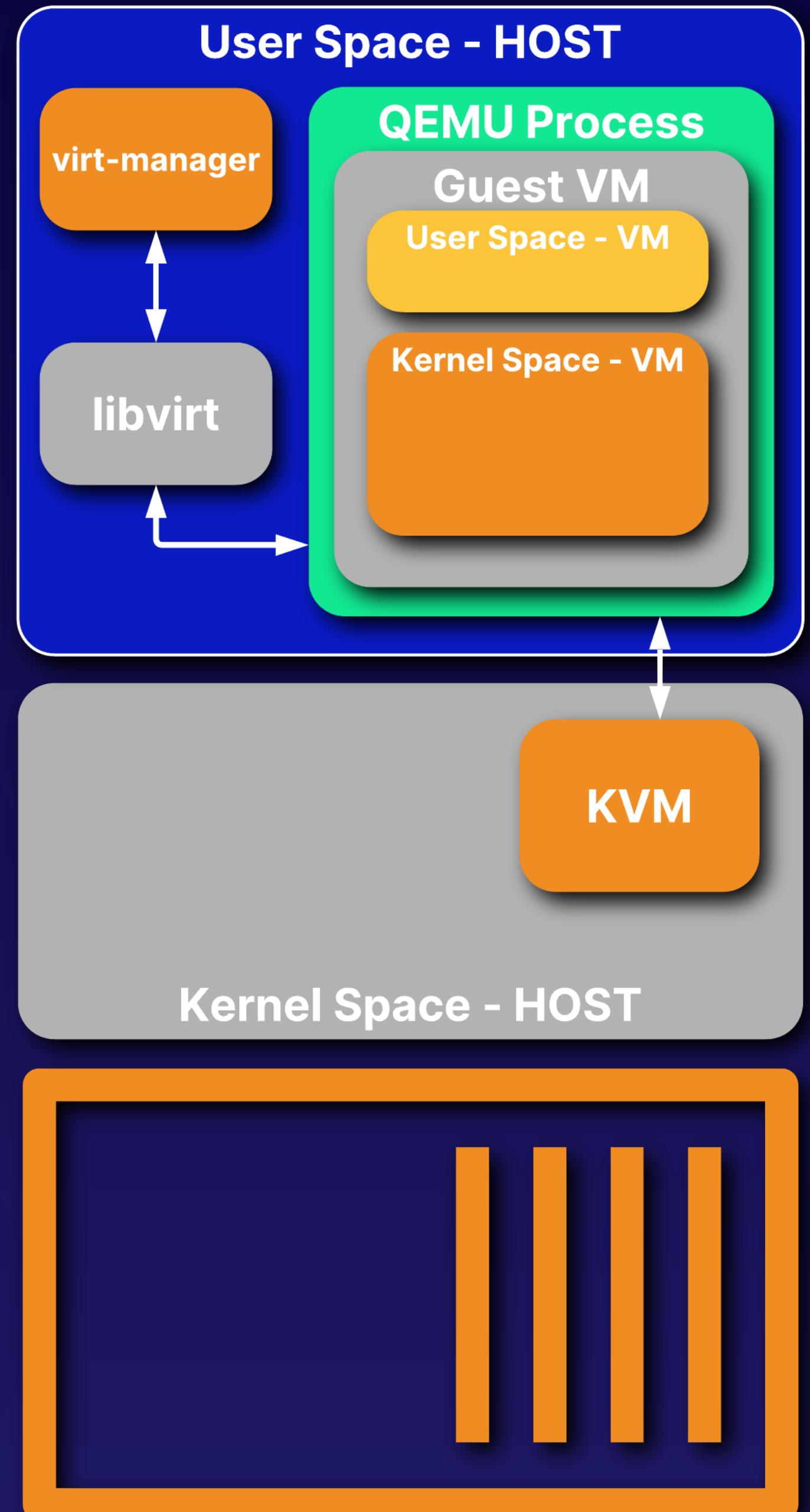
What Is `virt-manager`?

Virtual Machine Manager

virt-manager:

From the **Virtual Machine Manager** website:

"The **virt-manager** application is a desktop user interface for managing virtual machines through **libvirt**. It primarily targets **KVM** VMs but also manages **Xen** and **LXC** (Linux containers). It presents a summary view of running domains, their live performance, and resource utilization statistics. Wizards enable the creation of new domains, and configuration and adjustment of a domain's resource allocation and virtual hardware. An embedded **VNC** and **SPICE** client viewer presents a full graphical console to the guest domain."



If Linux
virtualization
were a
Corvette...



virt-manager
would be the fancy
touchscreen stereo
that gives you
easy-to-use
control over your
music and
navigation.



Virtual Machine Manager Functionality

Functionality

Virtual Machine Manager (**virt-manager**) can:

- Create, edit, start, and stop guest VMs
- Provide console access for guest VMs
- Provide operating metrics for guest VMs
- Display the current status of virtualization hosts and guest VMs
- Provide a choice of hypervisors
- Support LXC containers
- More



How Does It Work?

KVM/QEMU/libvirt



KVM

Provides acceleration through access to hardware virtualization extensions



QEMU

Type-2 hypervisor that utilizes KVM for type-1 hypervisor acceleration



libvirt

Toolkit to manage virtualization platforms, like QEMU/KVM

KVM/QEMU/libvirt



KVM

Provides acceleration through access to hardware virtualization extensions



QEMU

Type-2 hypervisor that utilizes KVM for type-1 hypervisor acceleration



libvirt

Toolkit to manage virtualization platforms, like QEMU/KVM

KVM/QEMU/**libvirt**



KVM

Provides acceleration through access to hardware virtualization extensions



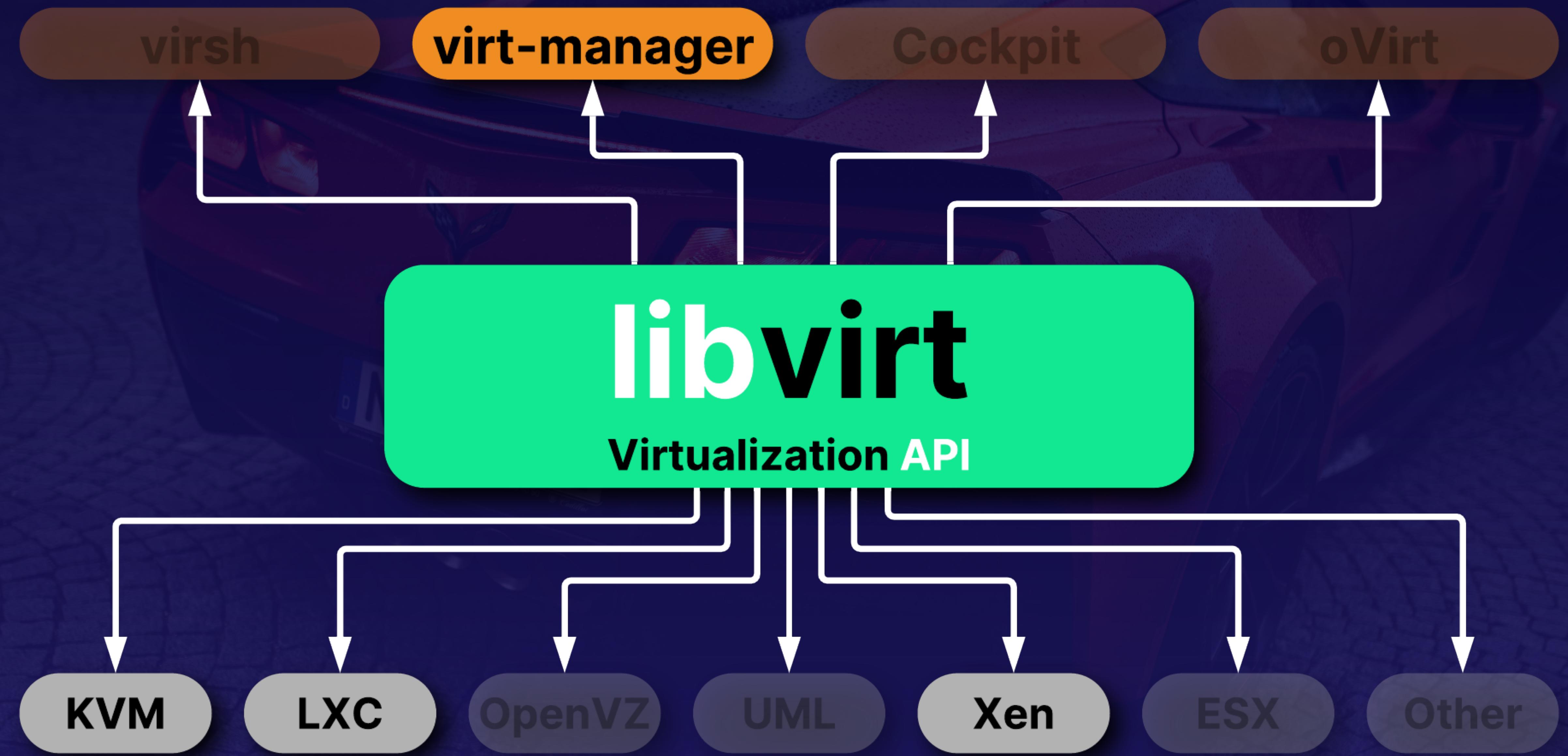
QEMU

Type-2 hypervisor that utilizes KVM for type-1 hypervisor acceleration



libvirt

Toolkit to manage virtualization platforms, like QEMU/KVM

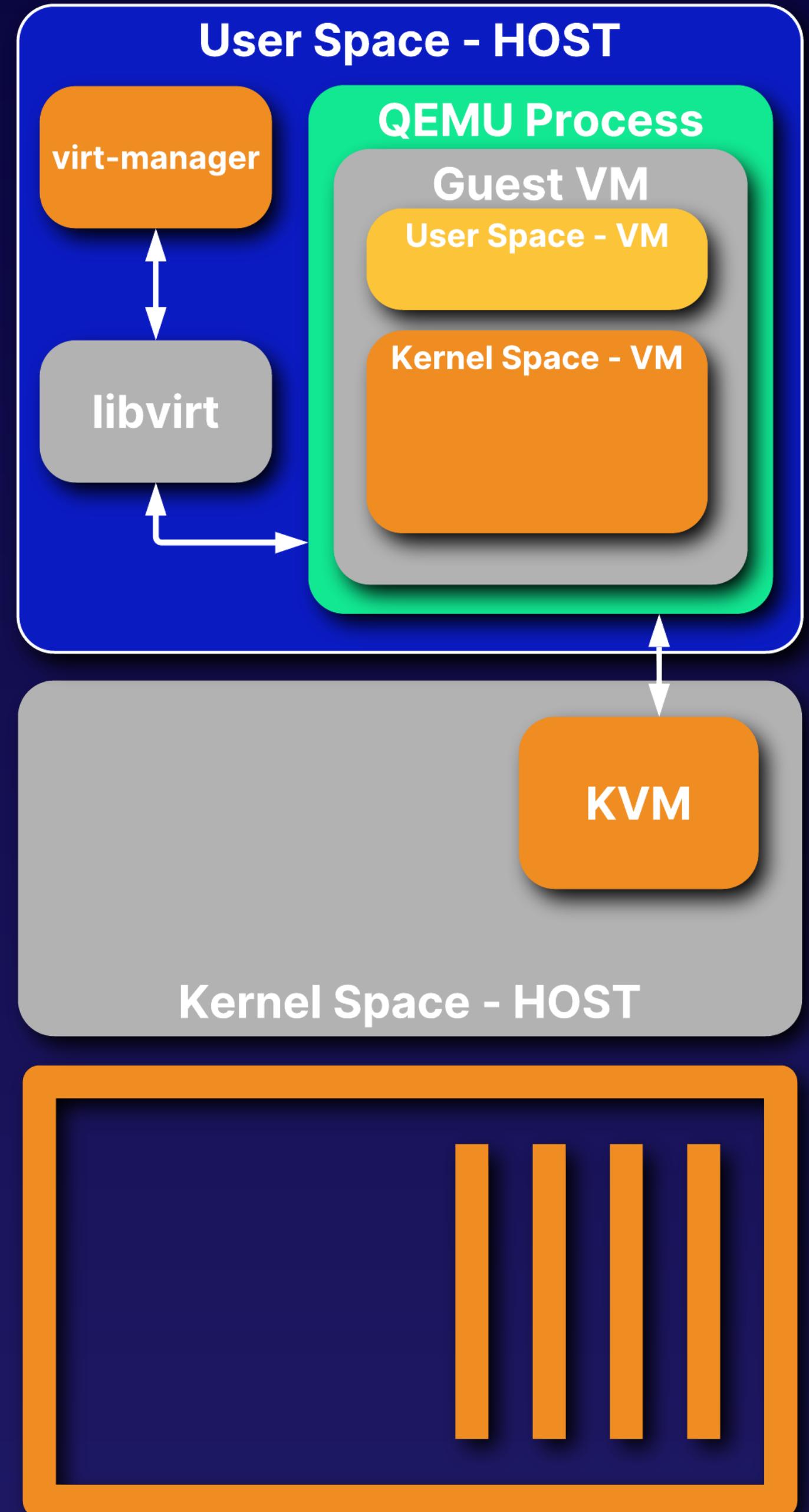


Virtual Machine Manager

virt-manager:

How does **virt-manager** work?

Virtual Machine Manager runs in user space and leverages **libvirt** to access the underlying **hypervisor(s)**. Because of this, **virt-manager** doesn't have to know how to “talk to” each particular **hypervisor** — it only has to communicate with **libvirt** via the **libvirtd** service on each host.



Linux Distribution Support



The **virt-manager** package is available for:

- Arch Linux
 - CentOS
 - Debian
 - Fedora
 - FreeBSD
 - Via Ports collection
 - Frugalware
 - Gentoo
 - Mandriva Linux
 - NetBSD
 - Via pkgsrc
 - OpenBSD
 - Via Ports collection
 - openSUSE
-
- Red Hat Enterprise Linux
 - Deprecated in 8, but still around
 - Web Console will be the replacement
 - Still can't do everything yet
 - Scientific Linux
 - Trisquel
 - TrueOS
 - Ubuntu
 - Void Linux



The **virt-manager** package is available for:

- Arch Linux
 - CentOS
 - Debian
 - Fedora
 - FreeBSD
 - via Ports collection
 - Frugalware
 - Gentoo
 - Mandriva Linux
 - NetBSD
 - via pkgsrc
 - OpenBSD
 - via Ports collection
 - openSUSE
-
- Red Hat Enterprise Linux
 - Deprecated in 8, but still around
 - Web Console will be the replacement
 - Still can't do everything yet
 - Scientific Linux
 - Trisquel
 - TrueOS
 - Ubuntu
 - Void Linux



In this lesson, we covered:



- What the Virtual Machine Manager (**virt-manager**) is
- The functionality of **virt-manager**
- How **virt-manager** works
- Linux distribution support for **virt-manager**



Tom Dean
Linux Training Architect