

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



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



# KVM Virtualization on Linux - Virtualization Scenarios

---

In this course, we've explored **KVM Virtualization on Linux** from **four** perspectives:

**Workstation virtualization using CentOS 8**

- Virtual Machine Manager (virt-manager)

**Web-based administration using CentOS 8**

- Cockpit/Web Console

**CLI administration using CentOS 8**

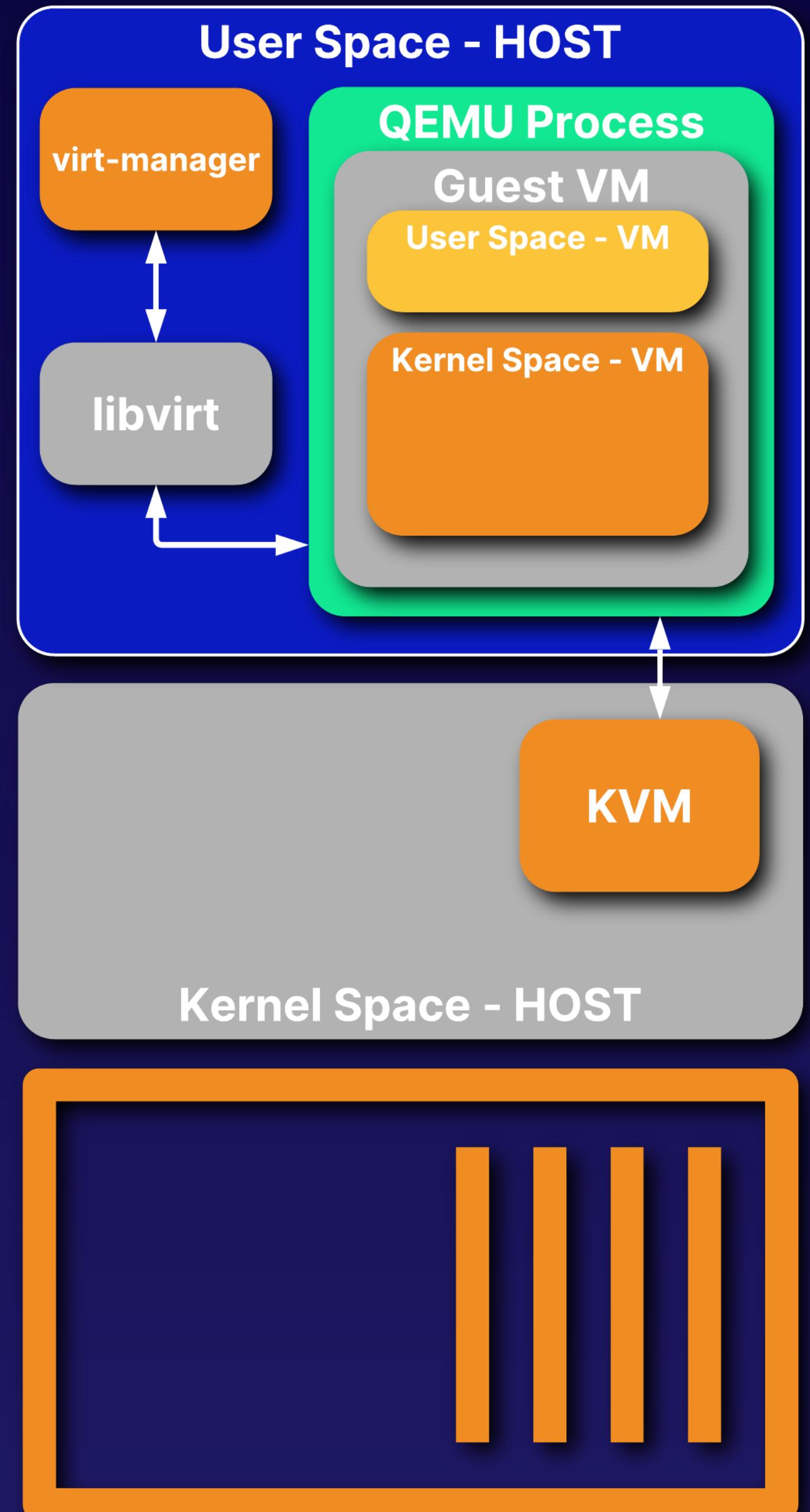
**Explored oVirt on CentOS 7**

# 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.



In this course, we've explored **KVM Virtualization on Linux** from **four** perspectives:

Workstation virtualization using  
**CentOS 8**

- Virtual Machine Manager  
(`virt-manager`)

**Web-based administration**  
using **CentOS 8**

- Cockpit/Web Console

CLI administration using  
**CentOS 8**

Explored oVirt on **CentOS 7**

# Cockpit/Web Console

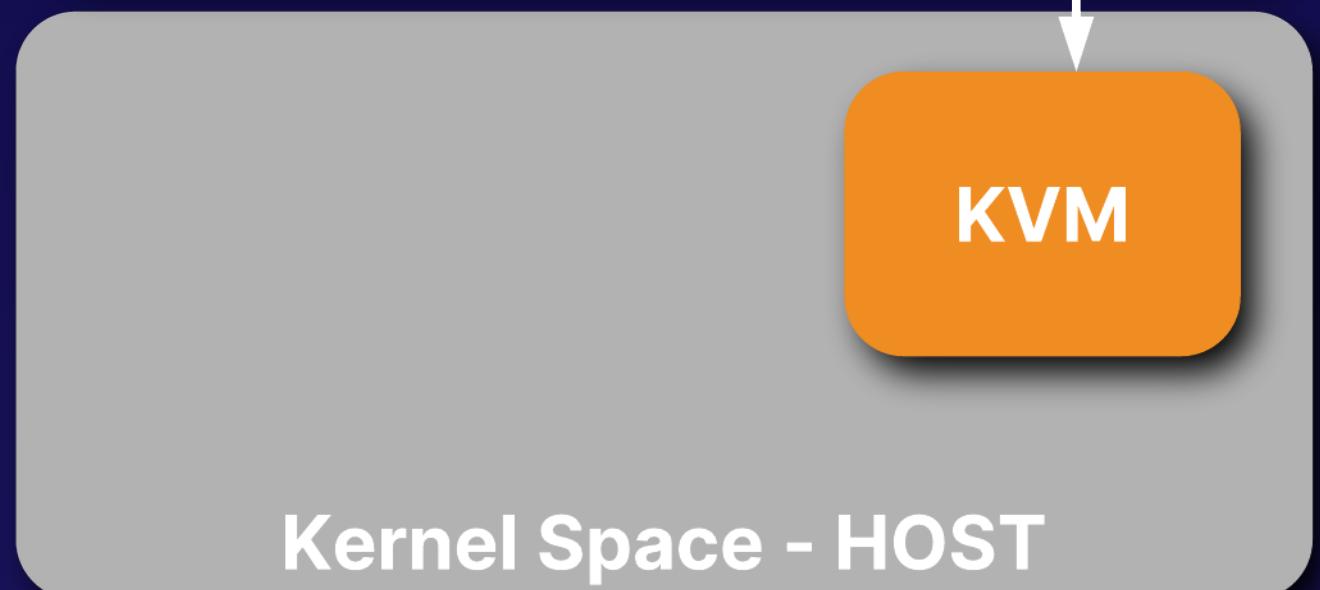
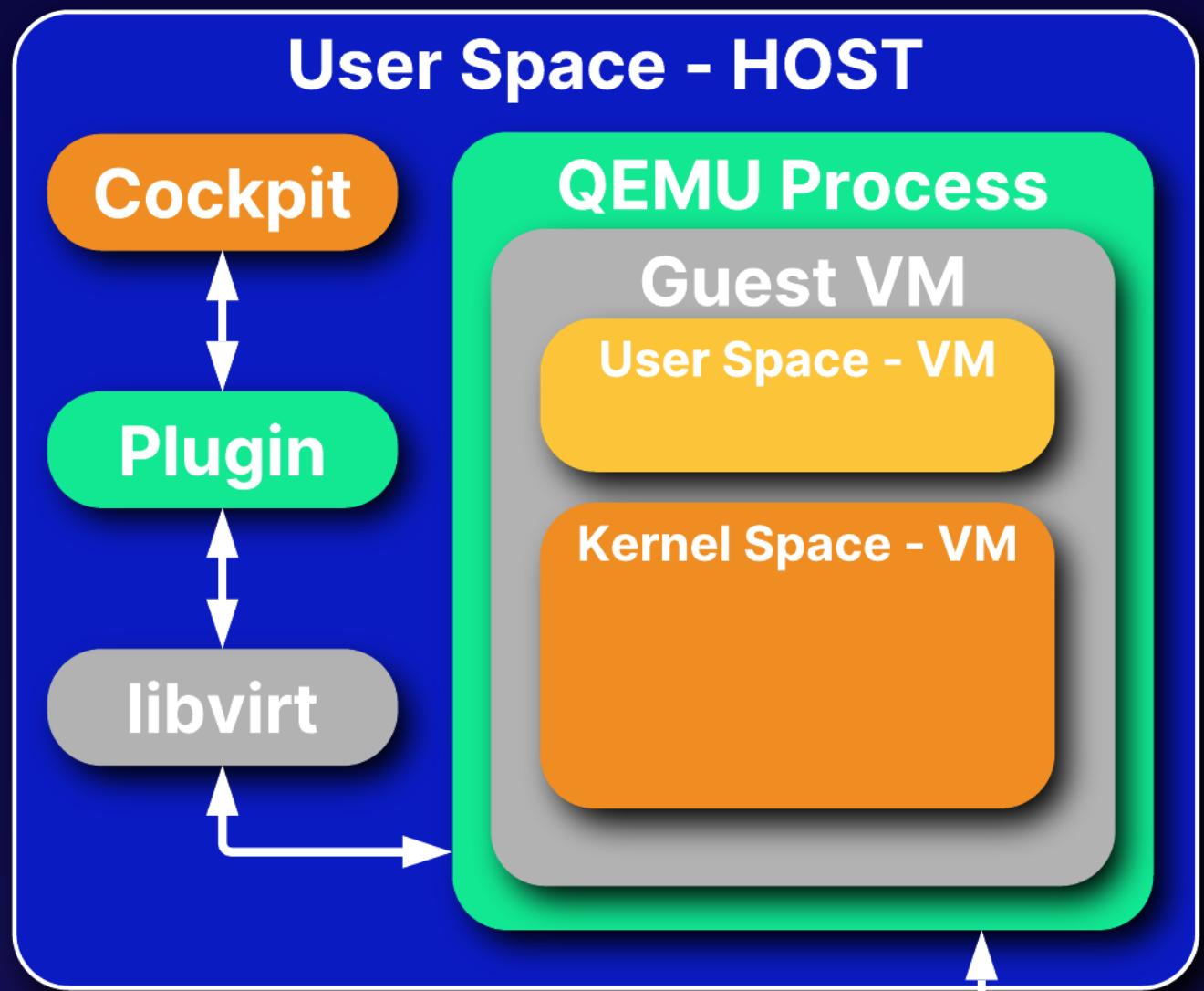
From the **Cockpit Project** website:

“The easy-to-use, integrated, glanceable,

and open **web-based** interface for your

servers.”

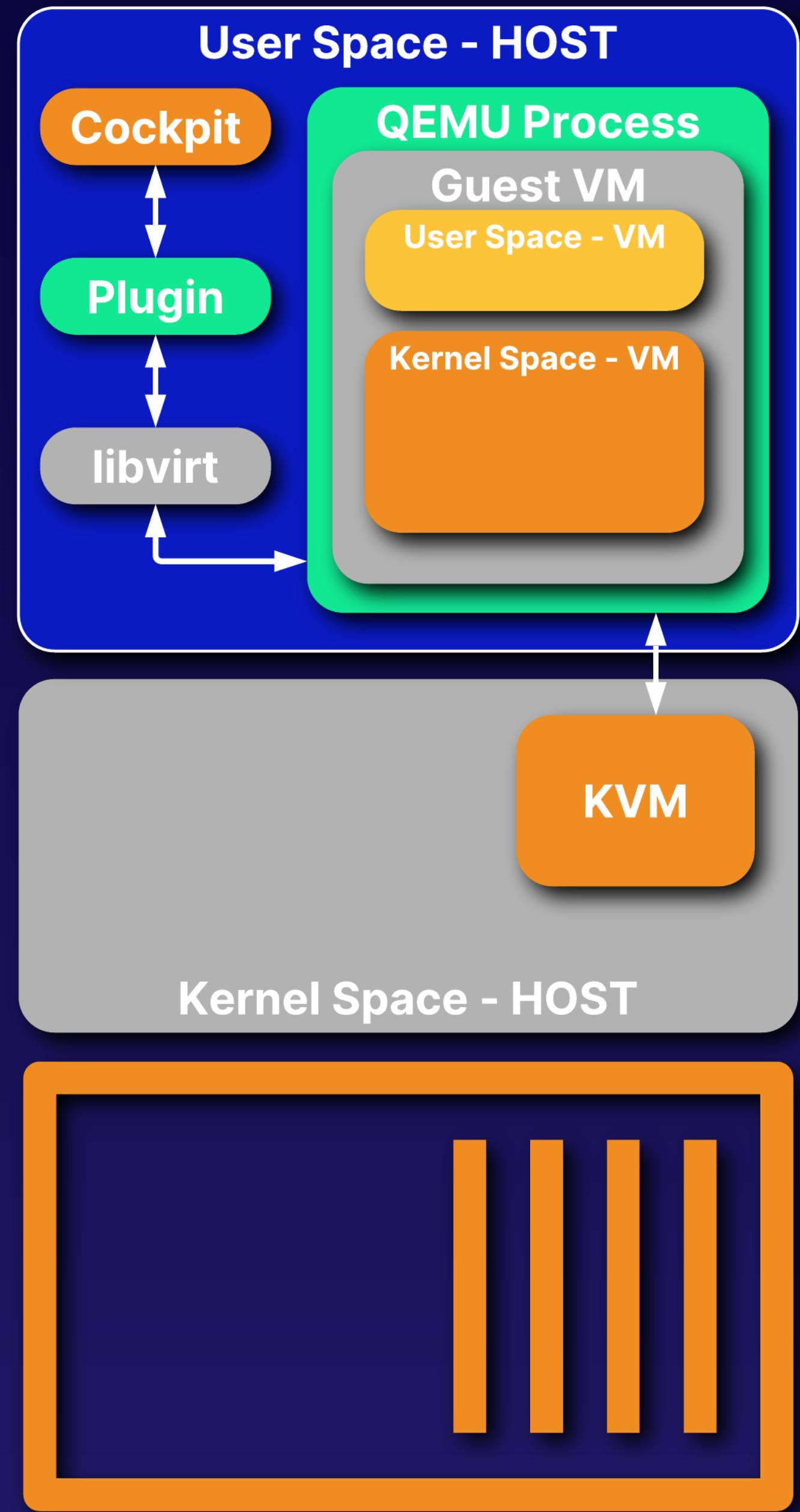
**Cockpit and Web Console are the same thing. Cockpit is the upstream project that feeds Web Console in the RHEL offering.**



# Cockpit/Web Console

How does **Cockpit** work?

**Cockpit** runs in user space and leverages a **plugin** to use **libvirt** to access the underlying **hypervisor**. Because of this, **Cockpit** doesn't have to know how to “**talk to**” the **hypervisor** — it only has to communicate with **libvirt** via the **libvirtd** service.

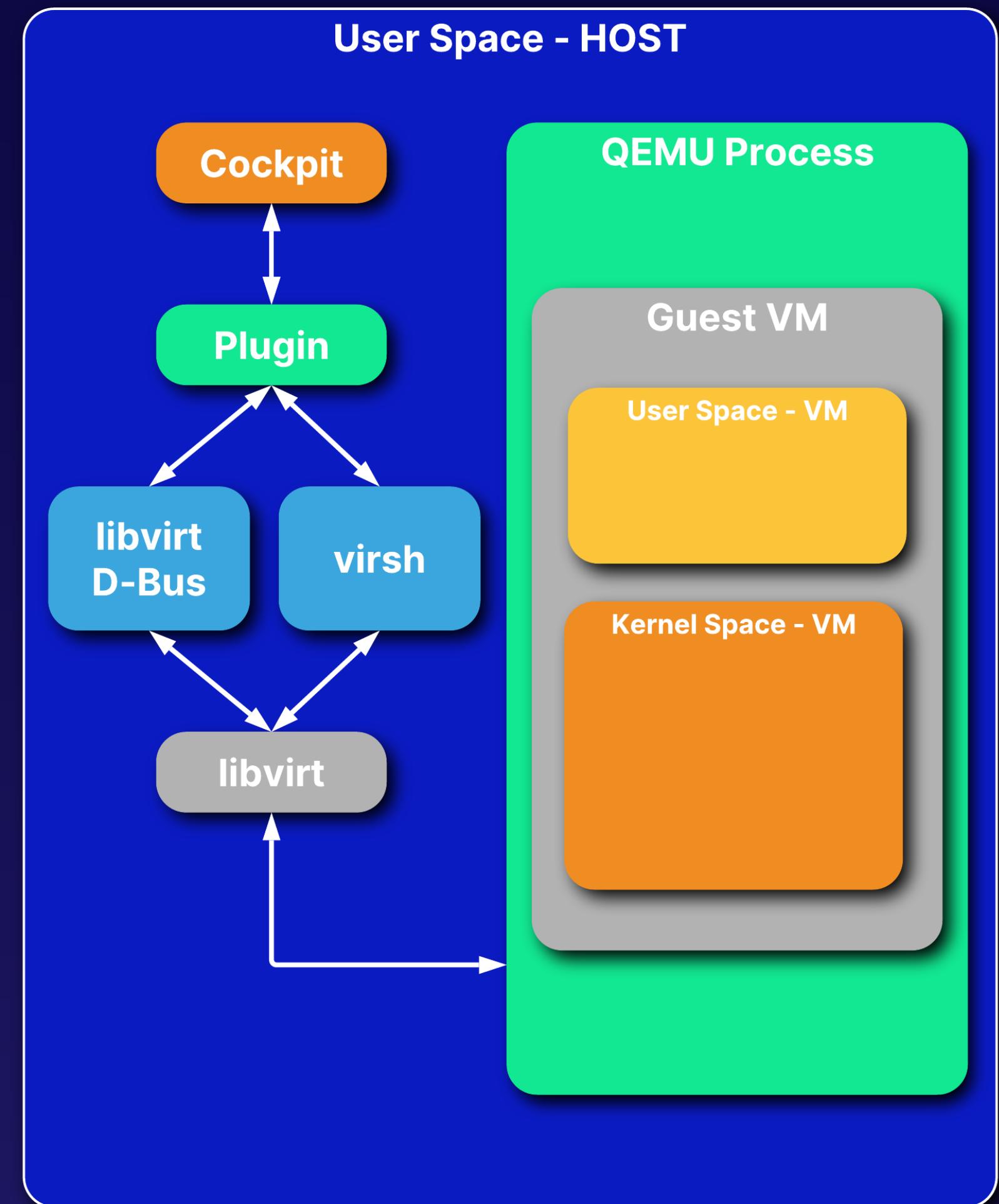


# Cockpit/Web Console

**How does Cockpit plugin work?**

“Access to **Libvirt** is wrapped either by the **virsh** tool or **libvirt D-Bus API** bindings, depending if the latter is installed on the system.”

— Cockpit Project



If Linux  
virtualization  
were a  
Corvette...



**Cockpit** would be  
the handy  
smartphone app  
that lets you unlock  
your doors when  
your keys are  
locked inside.



In this course, we've explored **KVM Virtualization on Linux** from **four** perspectives:

**Workstation virtualization using CentOS 8**

- Virtual Machine Manager  
(`virt-manager`)

**Web-based administration using CentOS 8**

- Cockpit/Web Console

**CLI administration using CentOS 8**

**Explored oVirt on CentOS 7**

# Functionality

## Why use **CLI** utilities?

- More control than other methods
- Enable scripting/automation
- No **GUI/web browser** required:
  - Less overhead
  - Easier to access the CLI
- Many management utilities leverage **virsh**:
  - Easier to **troubleshoot** issues



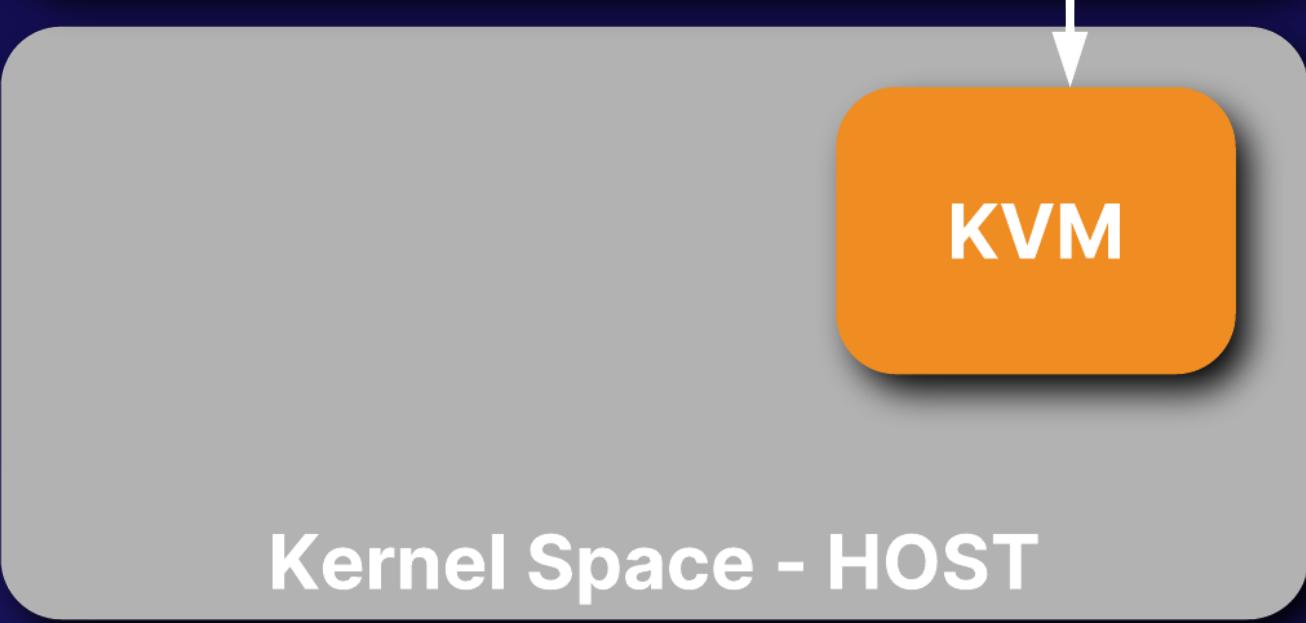
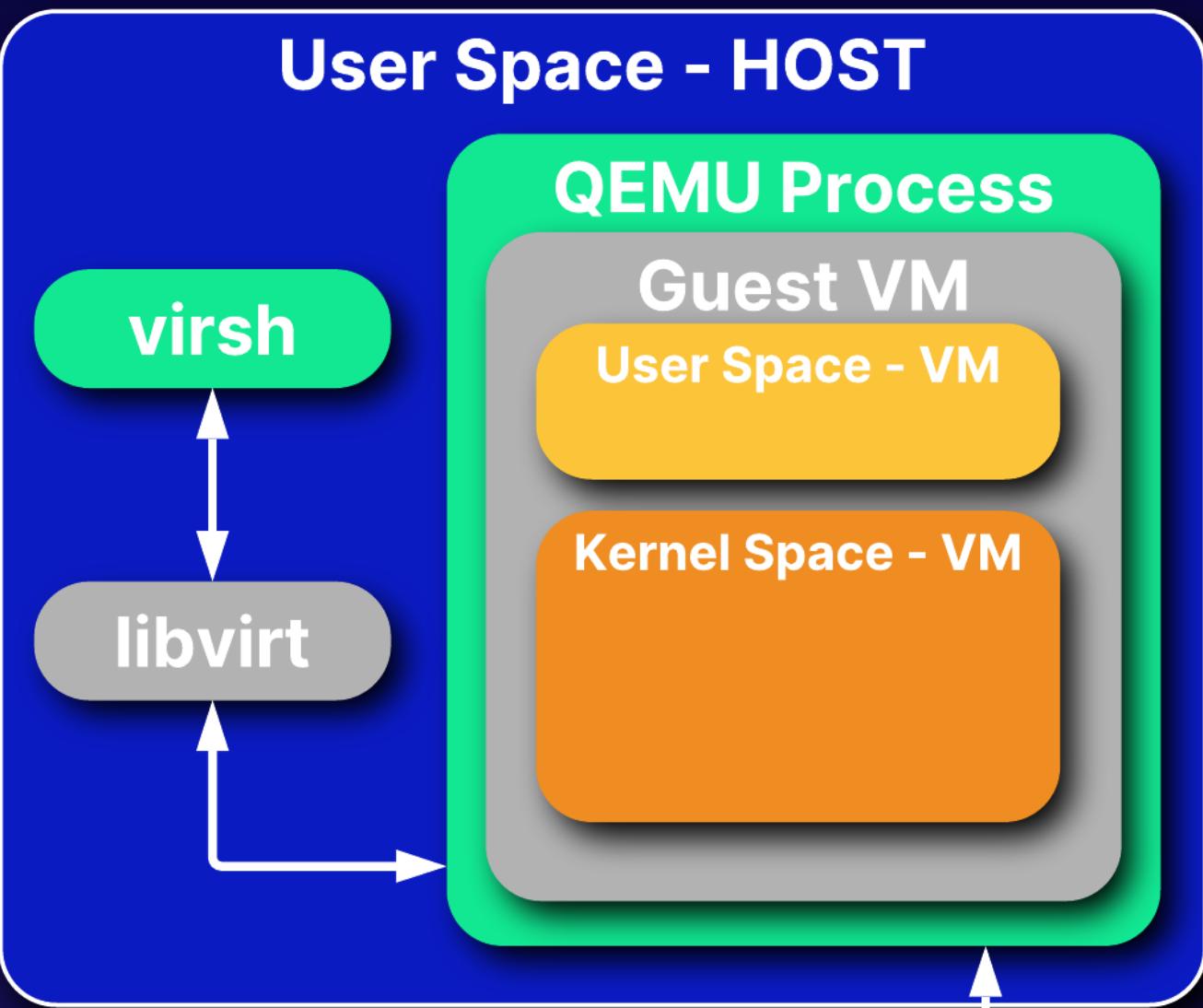
# virsh

From the **libvirt** website:

“The **virsh** program is the **main interface** for managing **virsh** guest

*domains.*”

virsh is installed when you install the libvirt package



# Functionality

**virsh can:**

- Create, edit, start, and stop guest VMs
- Provide console access for guest VMs
- Provide basic operating metrics for guest VMs
- Display the current status of guest VMs
- Manage virtual networks
- Manage virtual storage
- And more

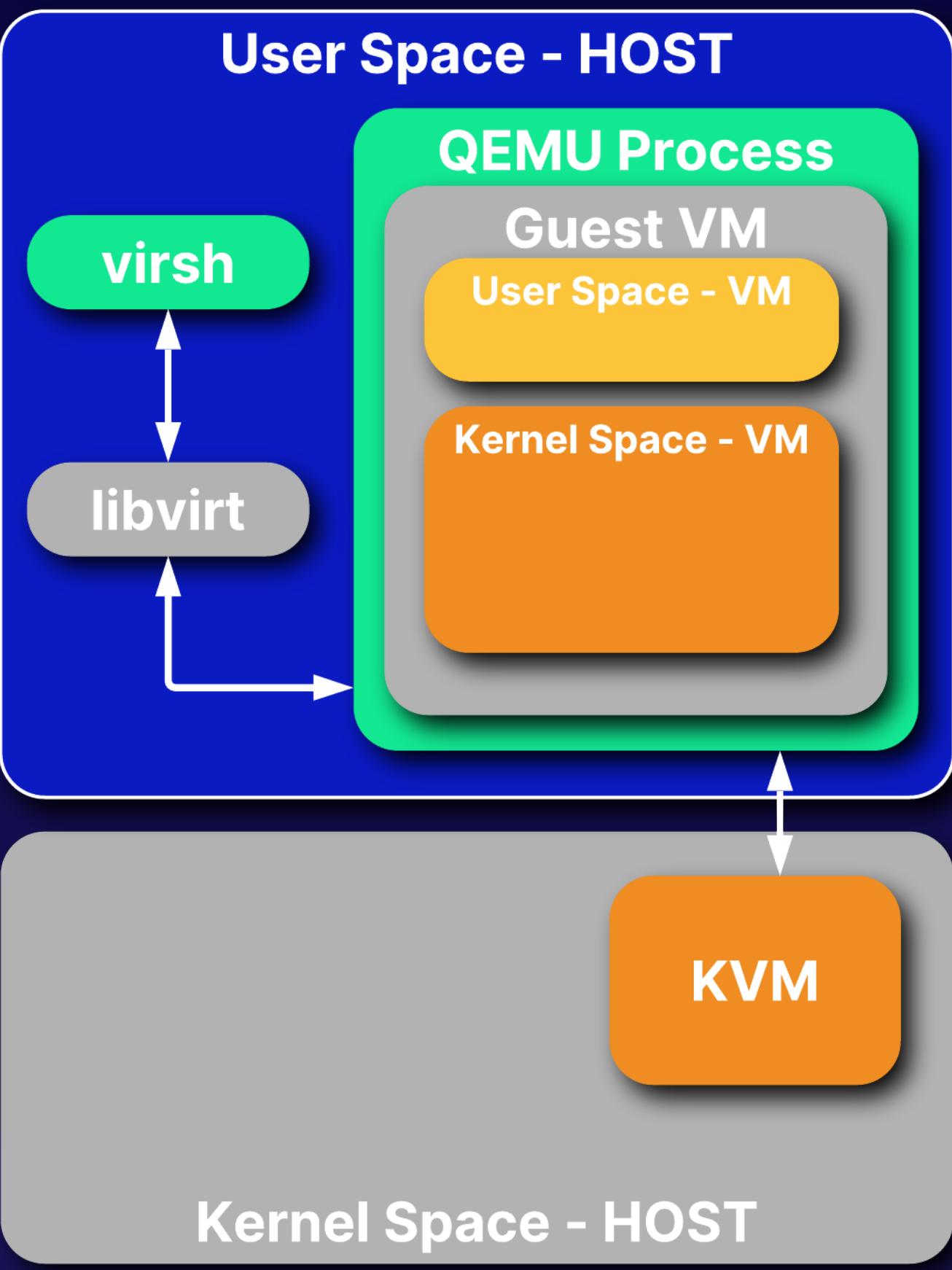


# virsh

## Interacting with virsh

You can use **virsh** in one of *two ways*:

- From the **Linux shell**
- As its own **interactive shell**:
  - The **virsh shell**



If Linux  
virtualization  
were a  
Corvette...



**virsh** would be the  
ultimate set of  
tools that can fix it  
all.



In this course, we've explored **KVM Virtualization on Linux** from **four** perspectives:

**Workstation virtualization using CentOS 8**

- Virtual Machine Manager  
(`virt-manager`)

**Web-based administration using CentOS 8**

- Cockpit/Web Console

**CLI administration using CentOS 8**

**Explored oVirt on CentOS 7**

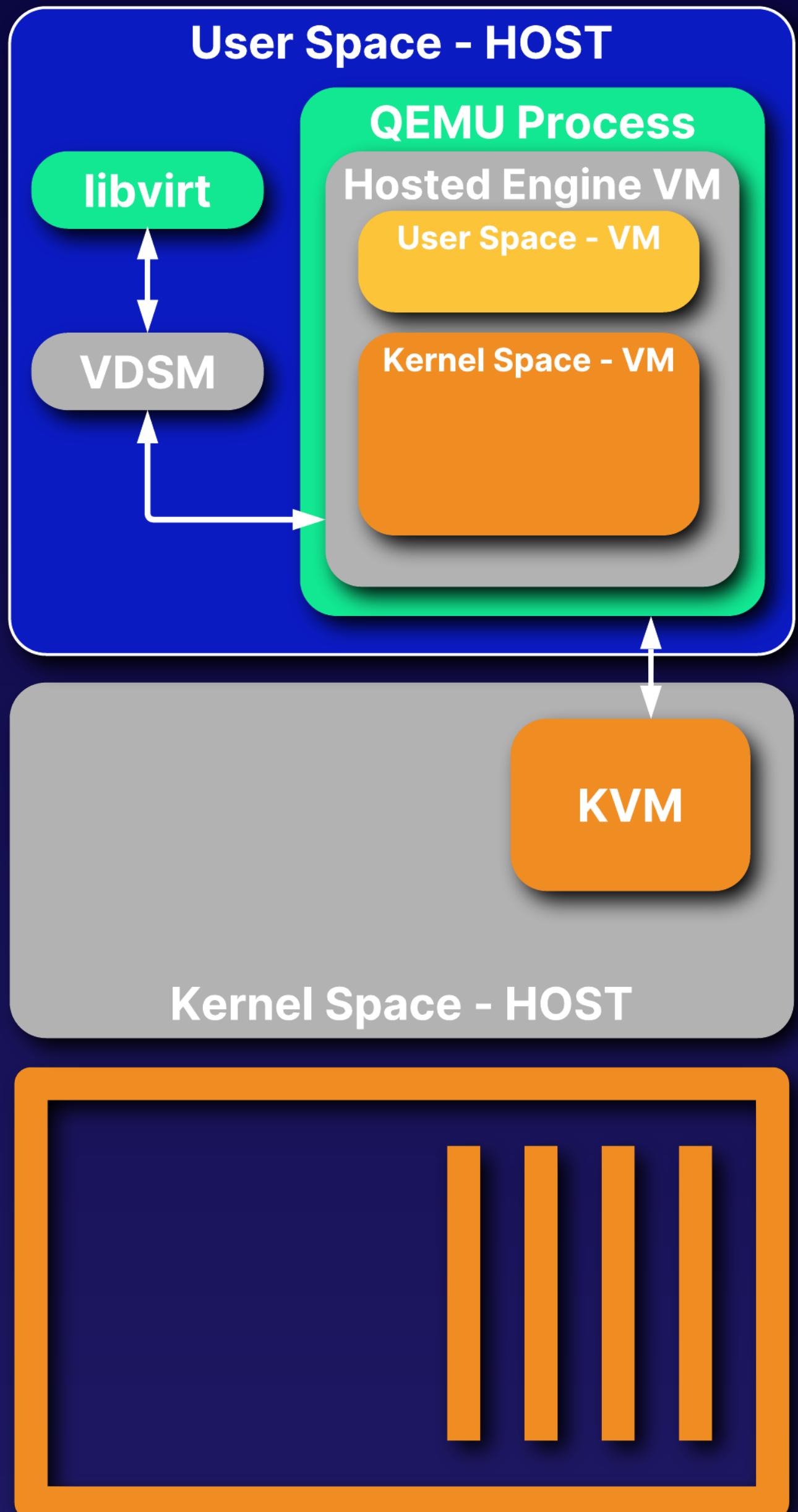
# oVirt

From the oVirt website:

“**oVirt** is an *open-source* distributed virtualization solution, designed to manage your *entire enterprise infrastructure*. **oVirt** uses the trusted **KVM** hypervisor and is built upon several other community projects, including **libvirt**, **Gluster**, **PatternFly**, and **Ansible**.”

oVirt is built on libvirt/QEMU/KVM

oVirt is the upstream of RHEV

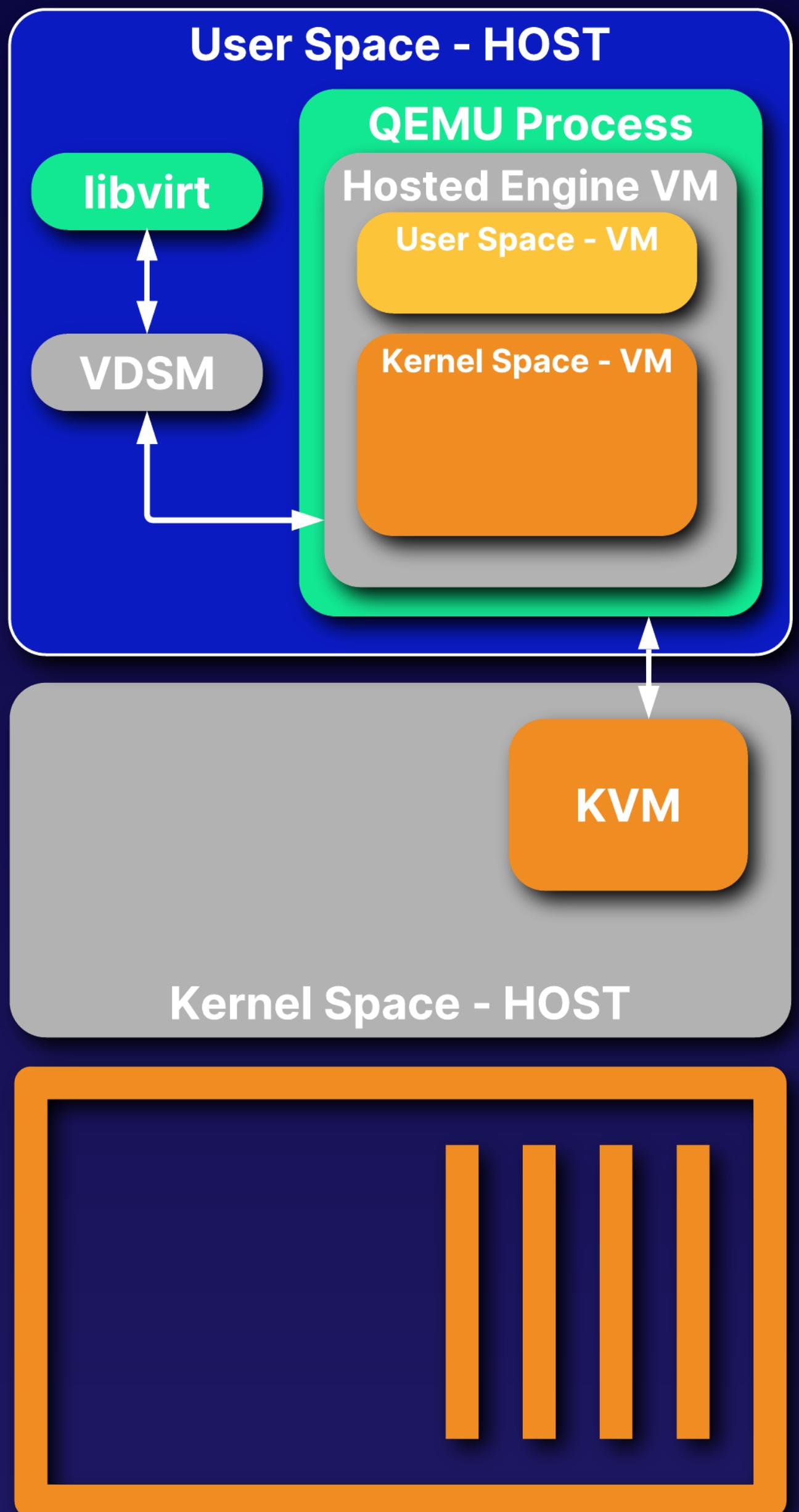


# oVirt

From the oVirt website:

## What's included?

- Rich **web-based** user interfaces for both *admin* and *non-admin* users
- Integrated **management** of *hosts*, *storage*, and *network configuration*
- **Live migration** of *virtual machines* and *disks* between *hosts* and *storage*
- **High availability** of *virtual machines* in the event of *host failure*



# Functionality

## ovirt can:

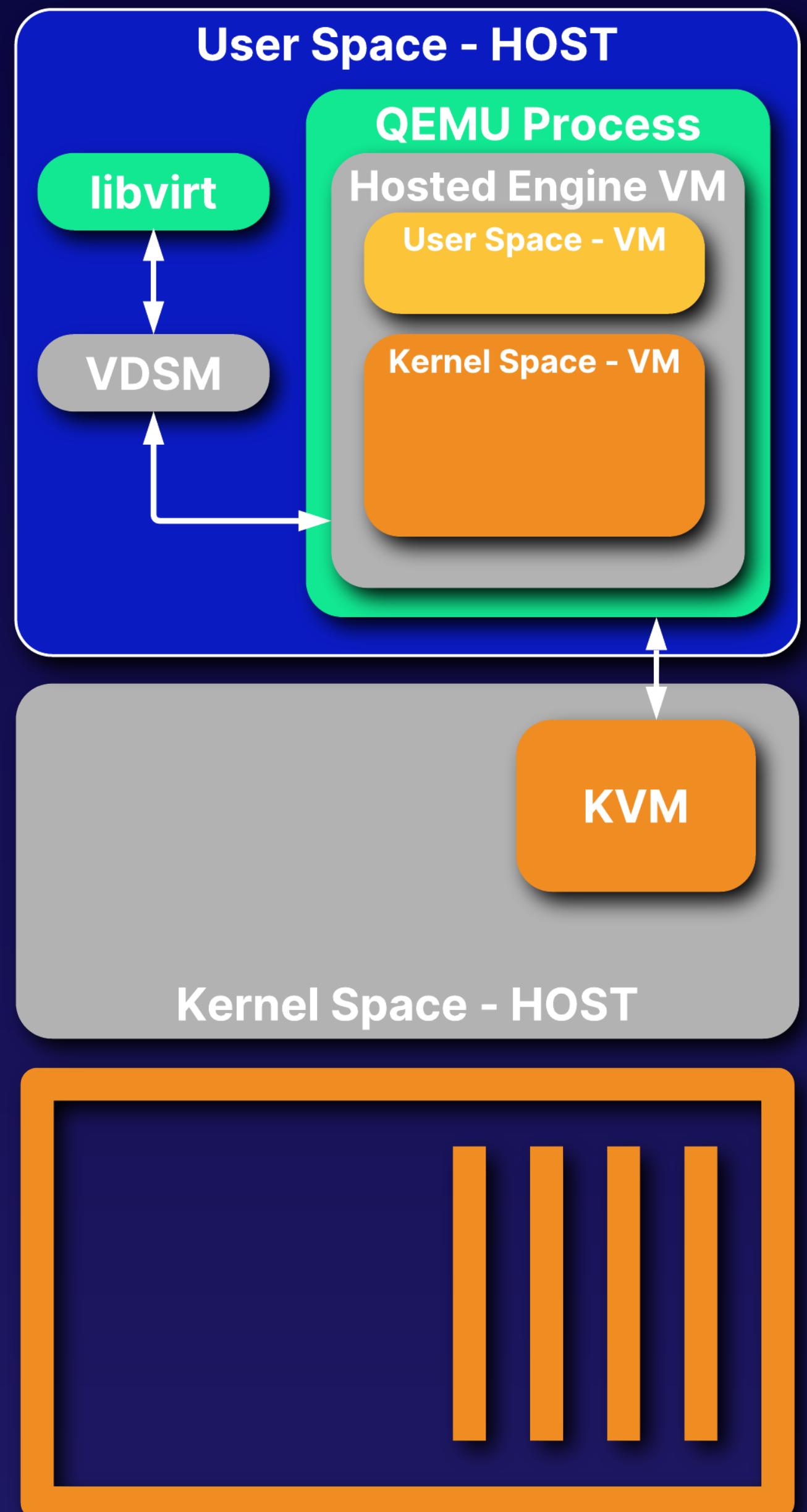
- Create, edit, start, and stop guest VMs
- Move guest VMs between hosts
- Provide console access for guest VMs
- Provide operating metrics for guest VMs
- Display the current status of guest VMs
- Manage virtual networks
- Manage virtual storage
- Provide disaster recovery
- More



# oVirt

oVirt is made up of two components:

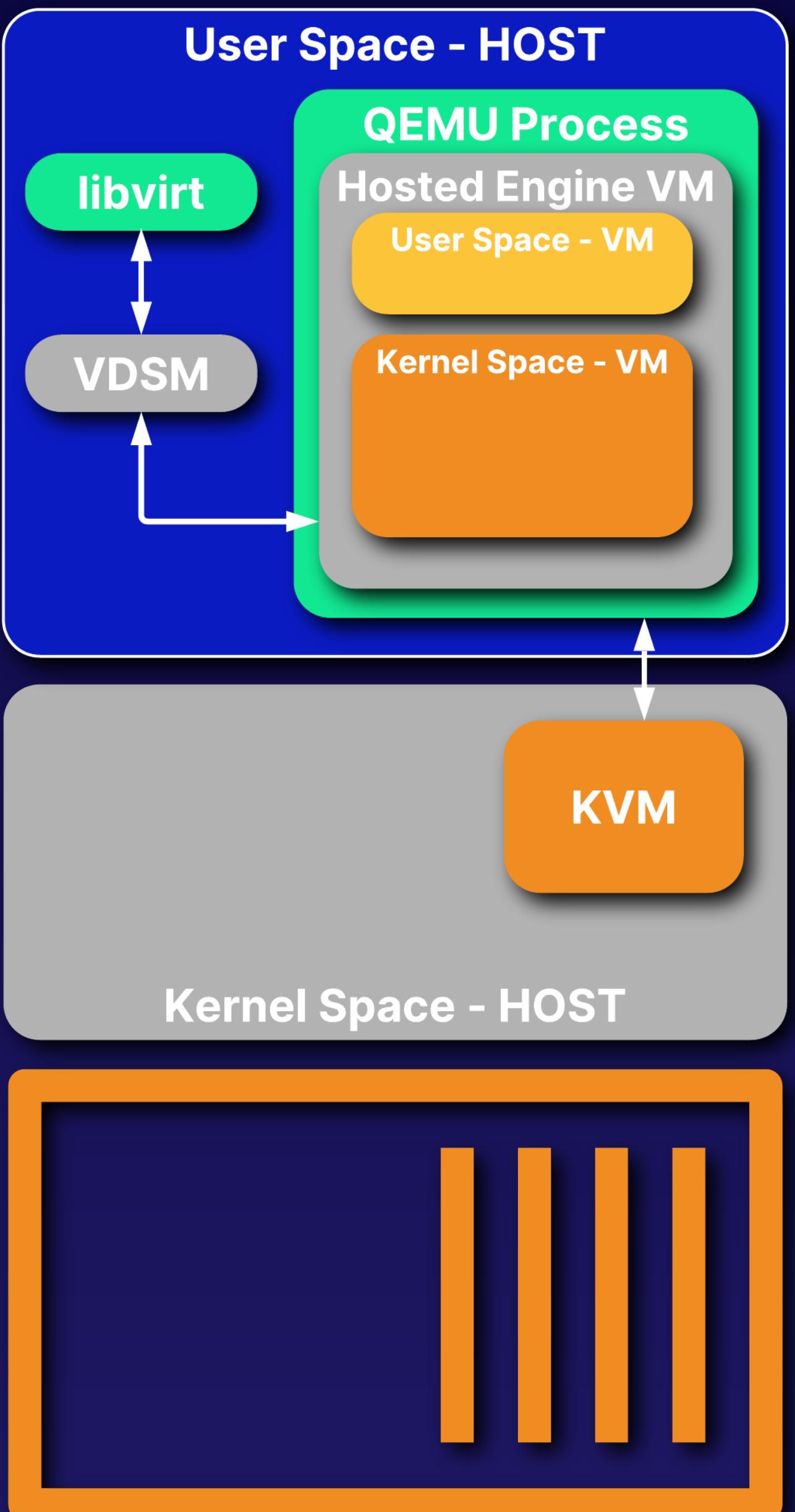
- oVirt engine:
  - Standalone or self-hosted
  - Java backend
  - GWT web toolkit frontend
  - WildFly (JBoss) application server
  - PostgreSQL database



# oVirt

oVirt is made up of **two** components:

- **oVirt node:**
  - **RHEL/CentOS** (predominantly)
  - **QEMU/KVM/libvirt**
  - **VDSM**
    - Node management **API**
    - **Gluster/PatternFly/Ansible**
  - Nodes can be **clustered**



If Linux  
virtualization  
were a  
Corvette...



**oVirt would be the  
pit crew that helps  
you *win the race!***





We hope  
you  
enjoyed the  
course!



In this course, we've aspired to provide an interesting journey through four scenarios that use **KVM Virtualization on Linux**. We hope you enjoyed it!



**Tom Dean**  
Linux Training Architect