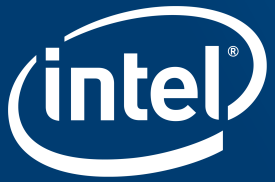


Kata Containers

Samuel Ortiz samuel.ortiz@intel.com - March 5th, 2019

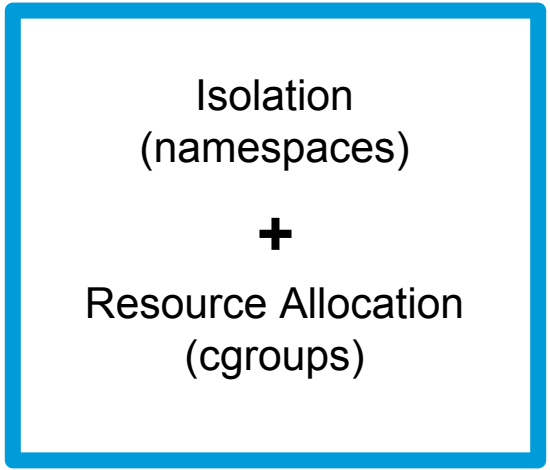
Agenda

- What are Kata Containers?
- What is it for?
- Who is running the project?

What is it?

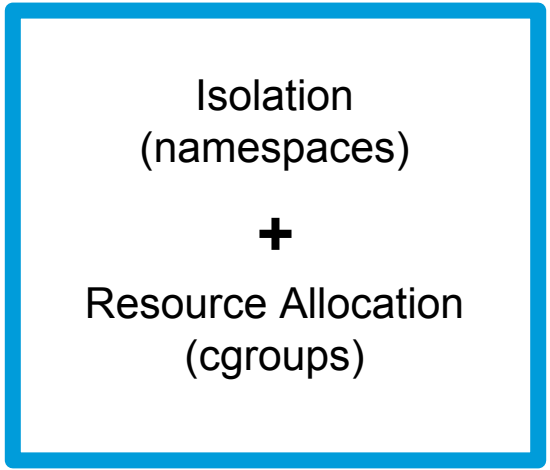
Container Workload =

Container Workload =



Container Workload =

RUNTIME



Container Workload =

RUNTIME

Isolation
(namespaces)

+

Resource Allocation
(cgroups)

Image Format
Encryption
Verification
Distribution

Container Workload =

RUNTIME

Isolation
(namespaces)

+

Resource Allocation
(cgroups)

PACKAGING

Image Format
Encryption
Verification
Distribution

Container Workload =

RUNTIME

Isolation
(namespaces)

+

Resource Allocation
(cgroups)

PACKAGING

Image Format
Encryption
Verification
Distribution

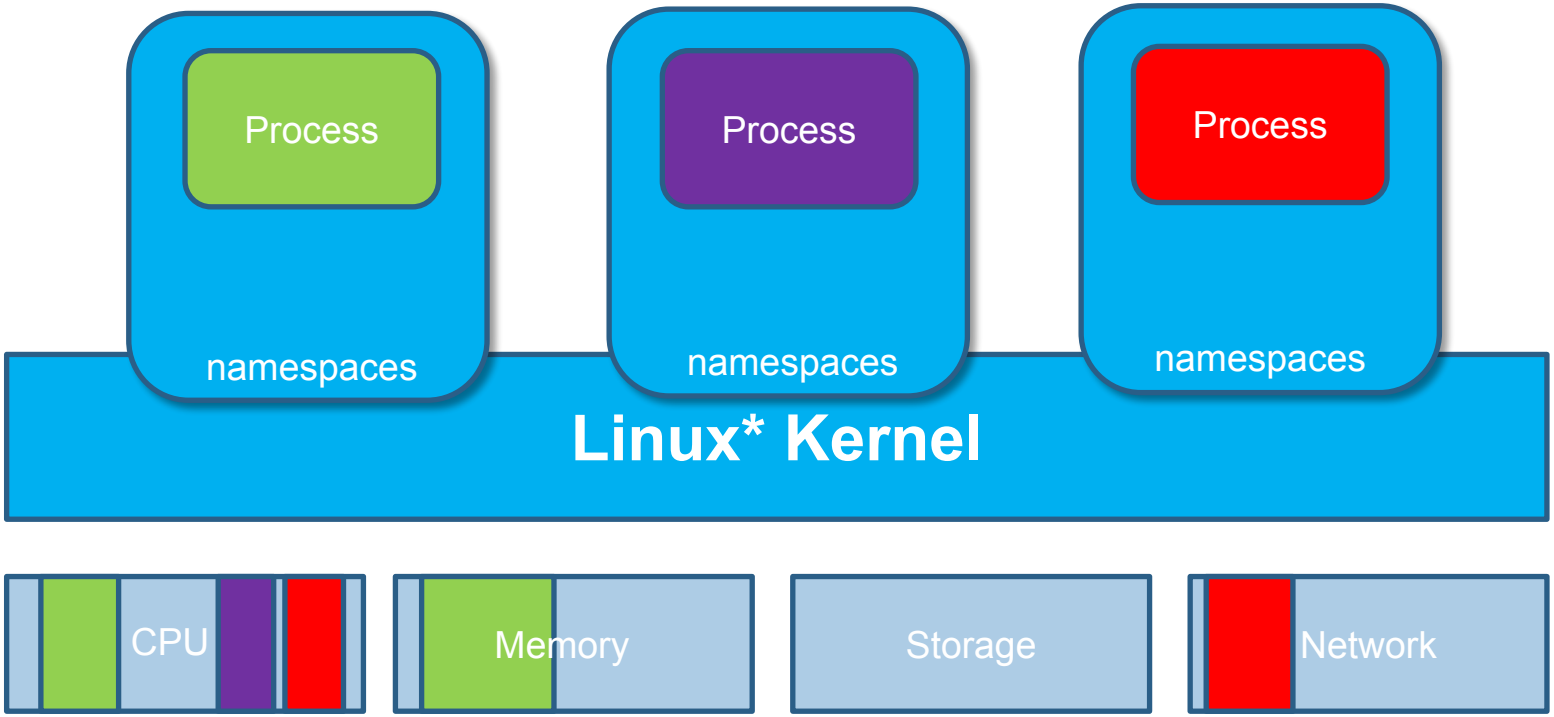
Kata Containers =

RUNTIME

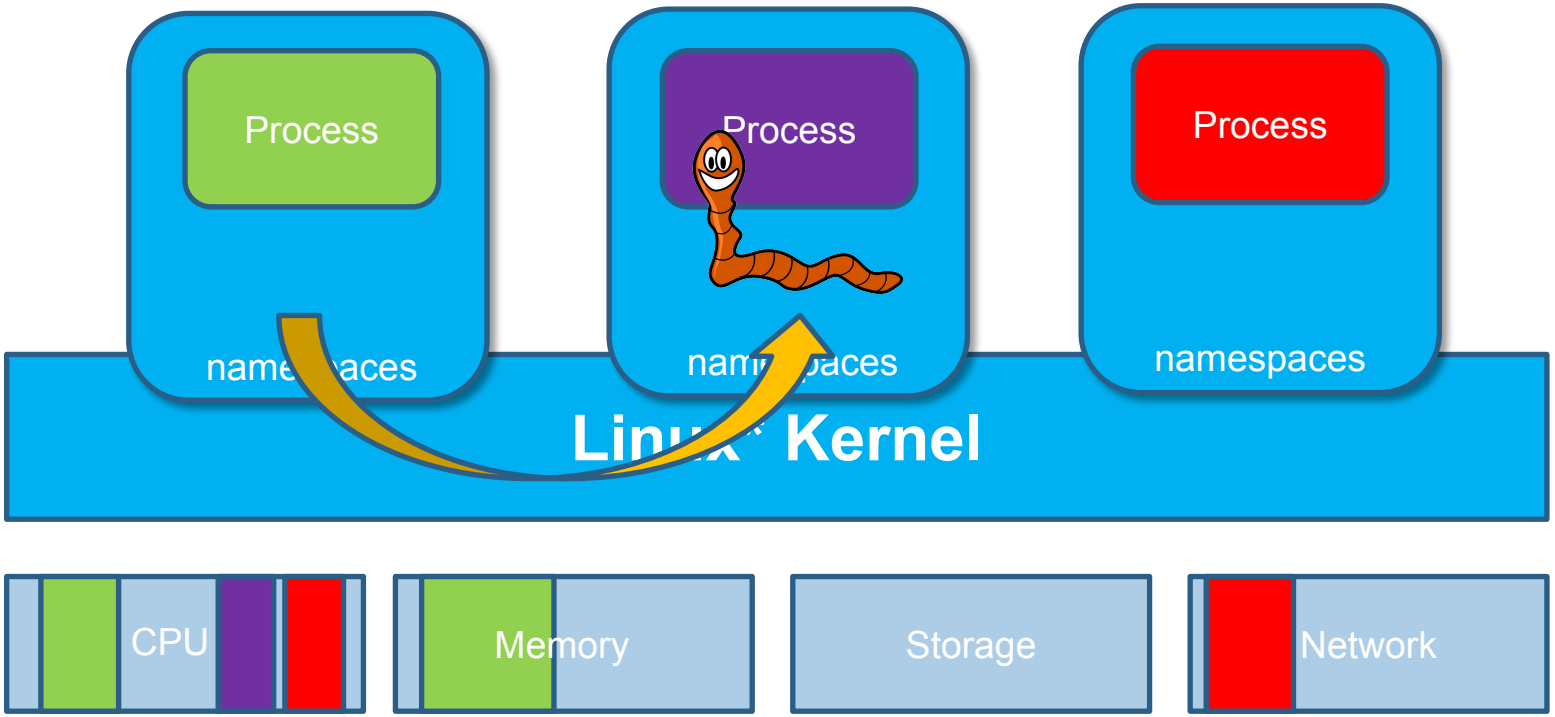
Isolation
(namespaces)

+

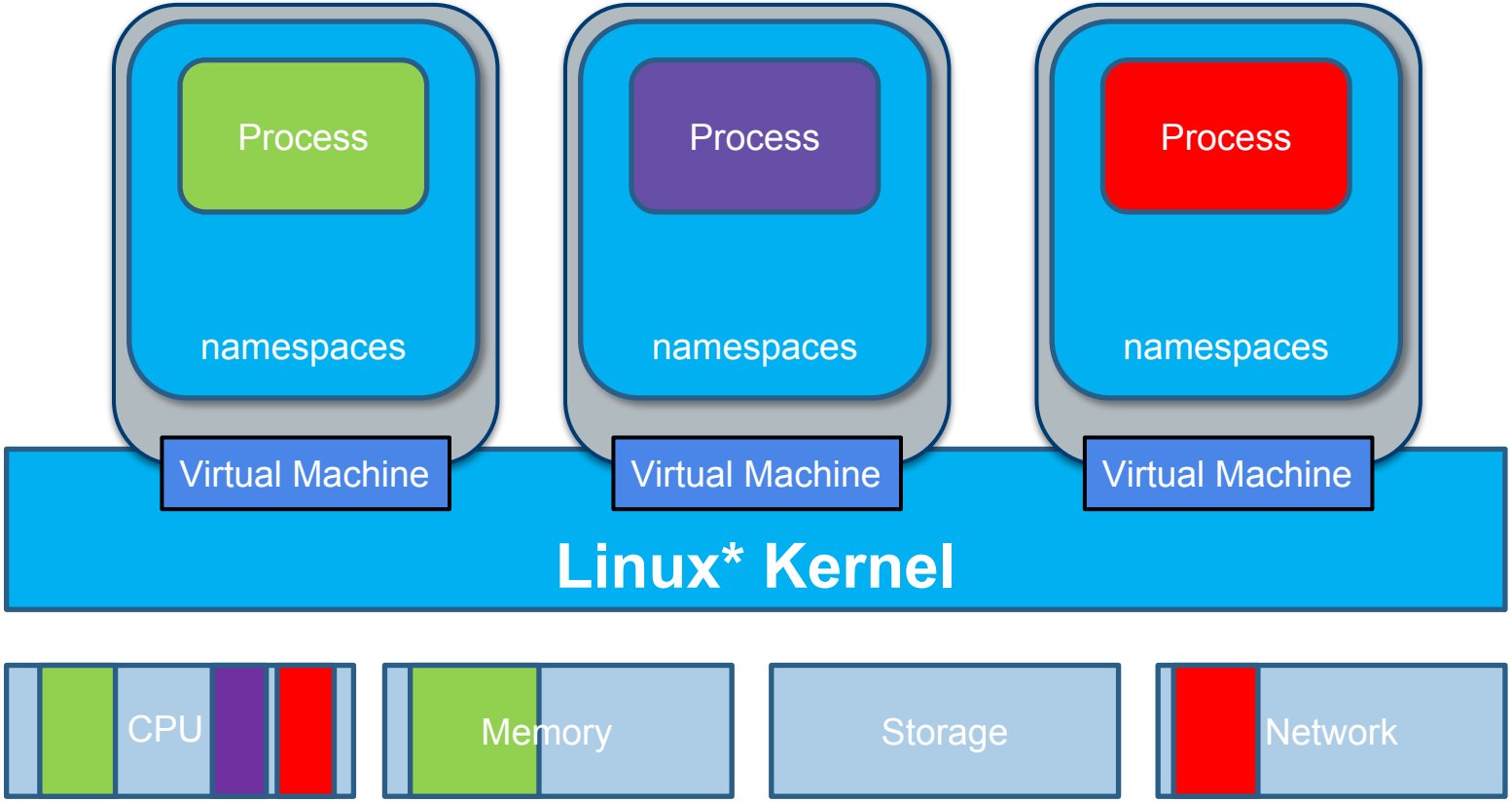
Resource Allocation
(cgroups)



*Other names and brands may be claimed as the property of others.


Process
Process
Process
namespaces
namespaces
Linux* Kernel
CPU
Memory
Storage
Network


ESCAPE
DEAD ISLAND
#TheCrateEscape



*Other names and brands may be claimed as the property of others.

As secure as a virtual machine

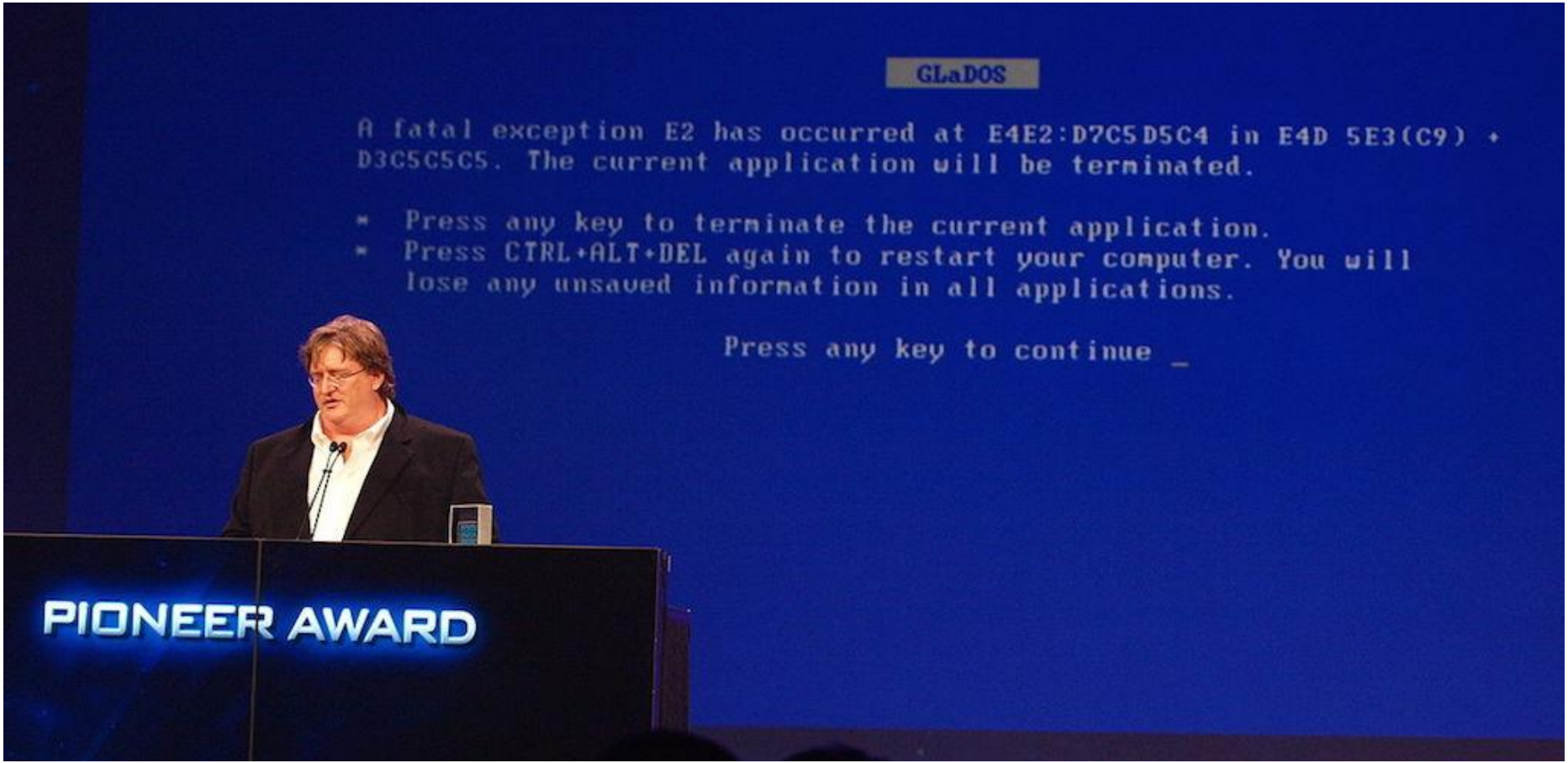
As secure as a virtual machine

Small and fast like a container

As secure as a virtual machine

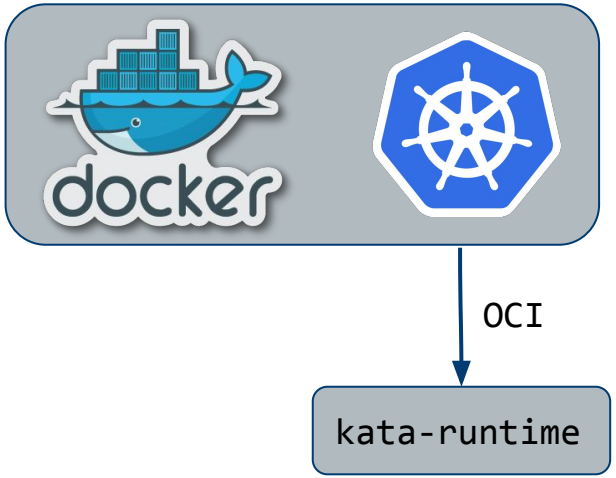
Small and fast like a container

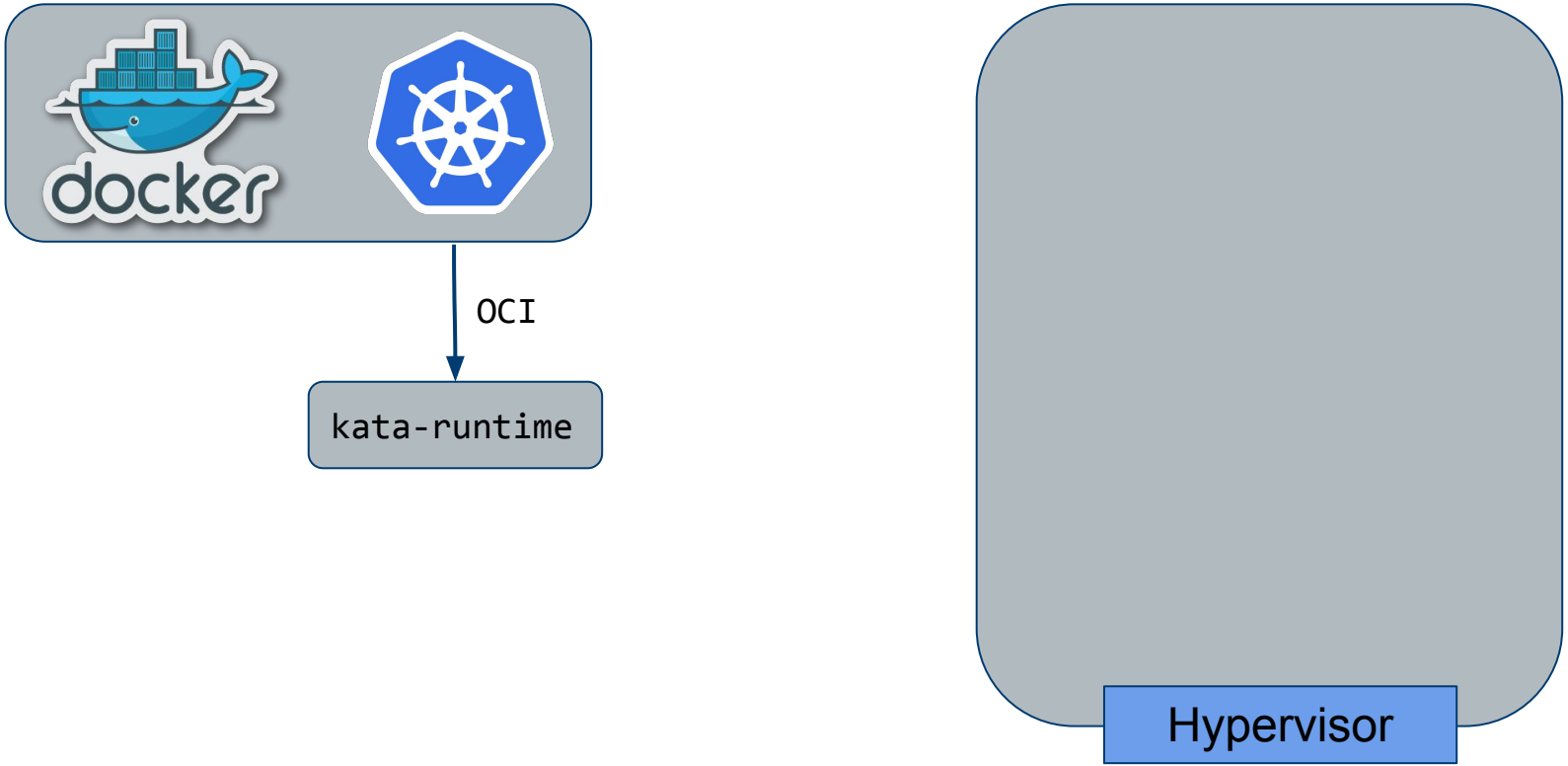
Looks and behaves like a container

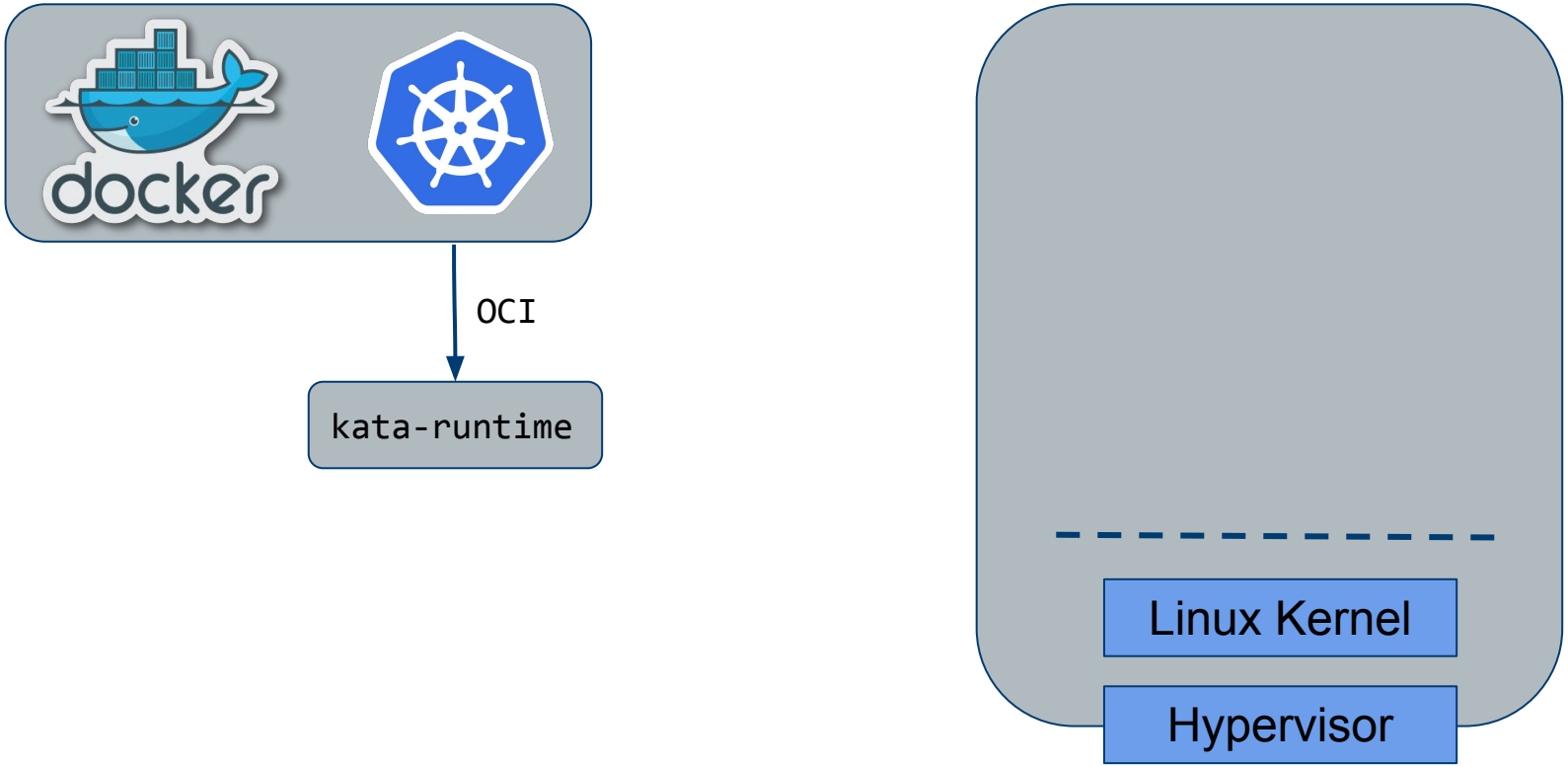

GLaDOS
A fatal exception E2 has occurred at E4E2:D7C5D5C4 in E4D 5E3(C9) +
D3C5C5C5. The current application will be terminated.
* Press any key to terminate the current application.
* Press CTRL+ALT+DEL again to restart your computer. You will
lose any unsaved information in all applications.
Press any key to continue _
PIONEER AWARD

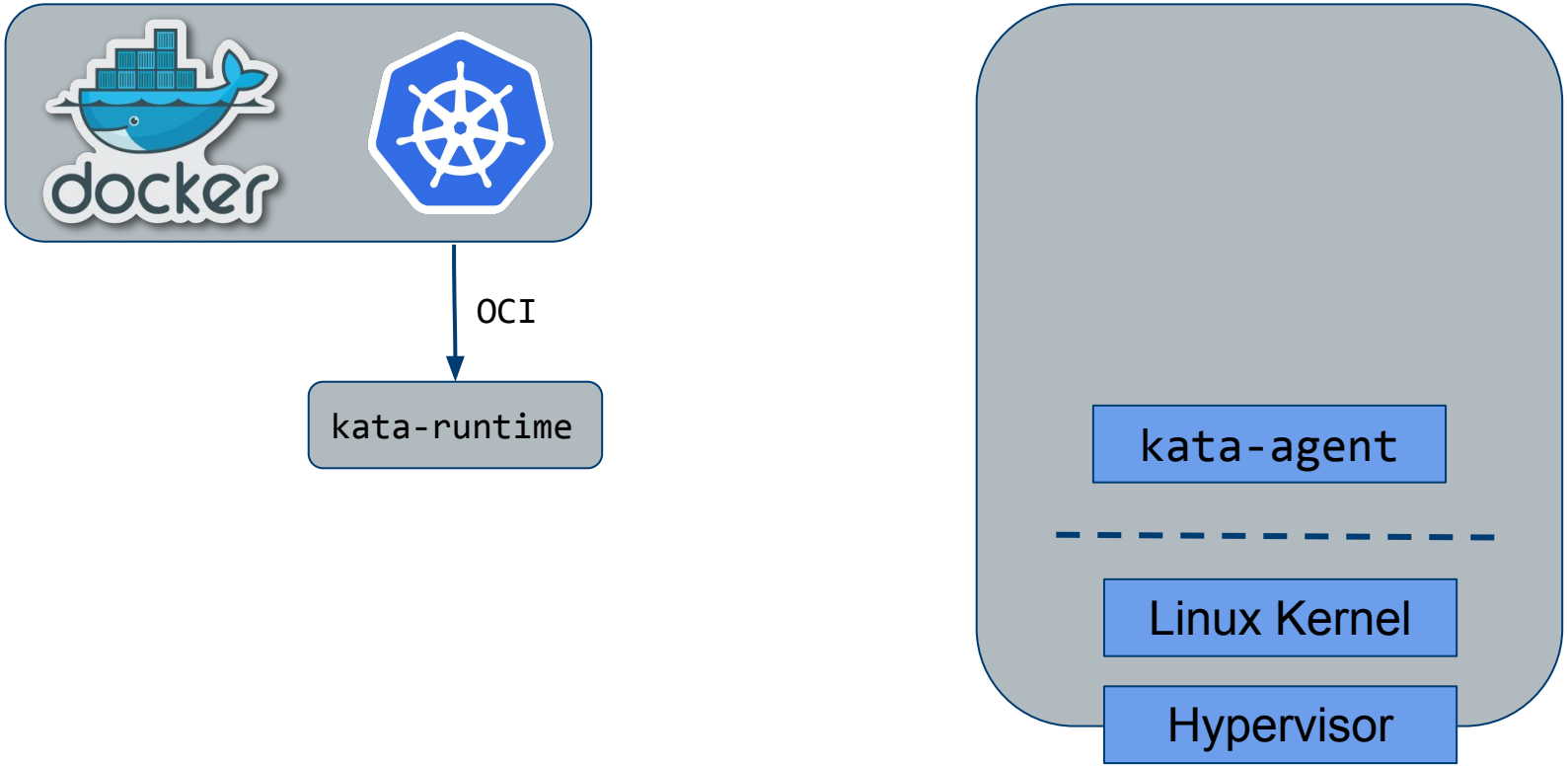

GLaDOS
A fatal exception E2 has occurred at E4E2:D7C5D5C4 in E4D 5E3(C9) +
D3C5C5C5. The current application will be terminated.
* Press any key to terminate the current application.
DEMO
PIONEER AWARD

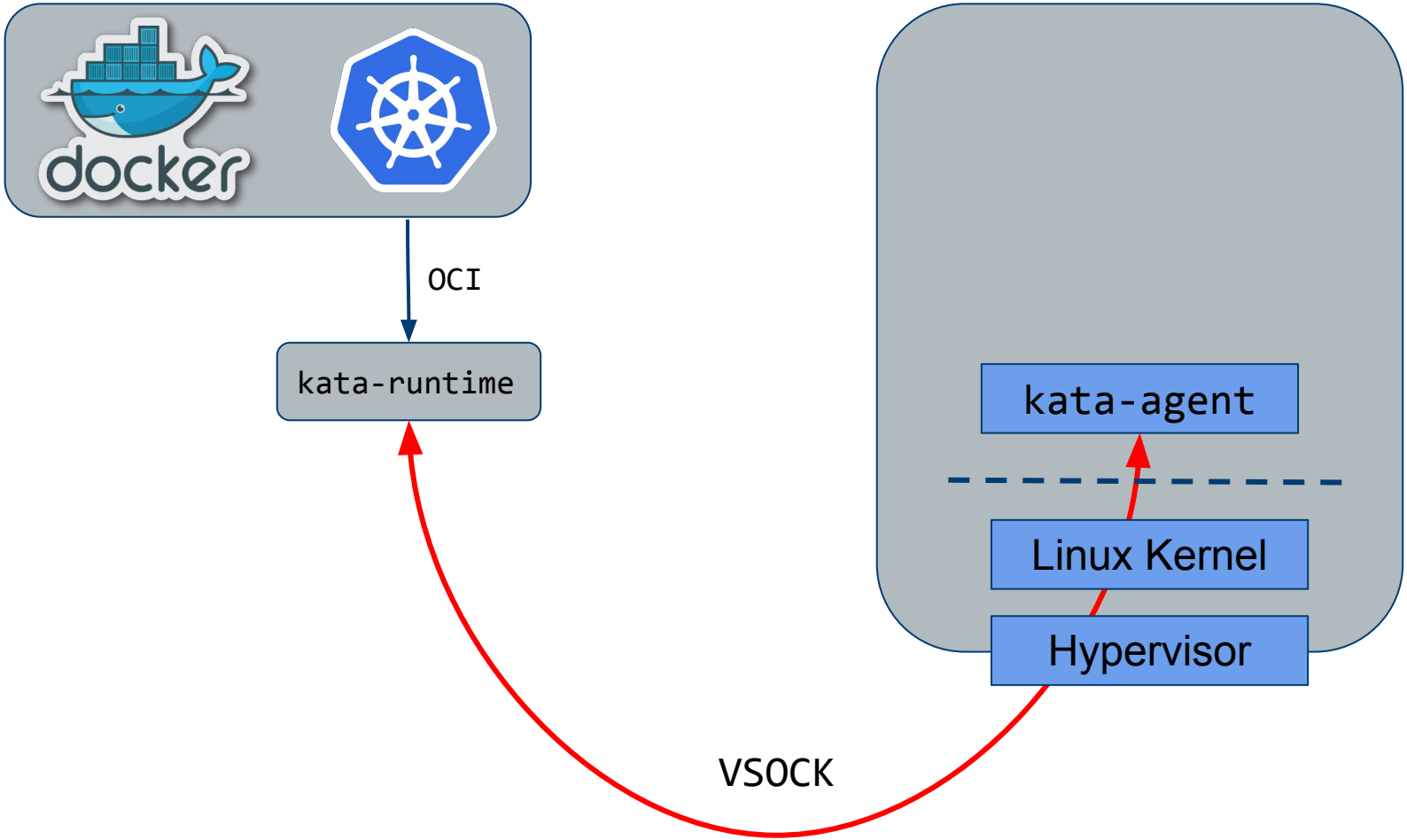
As secure as a virtual machine

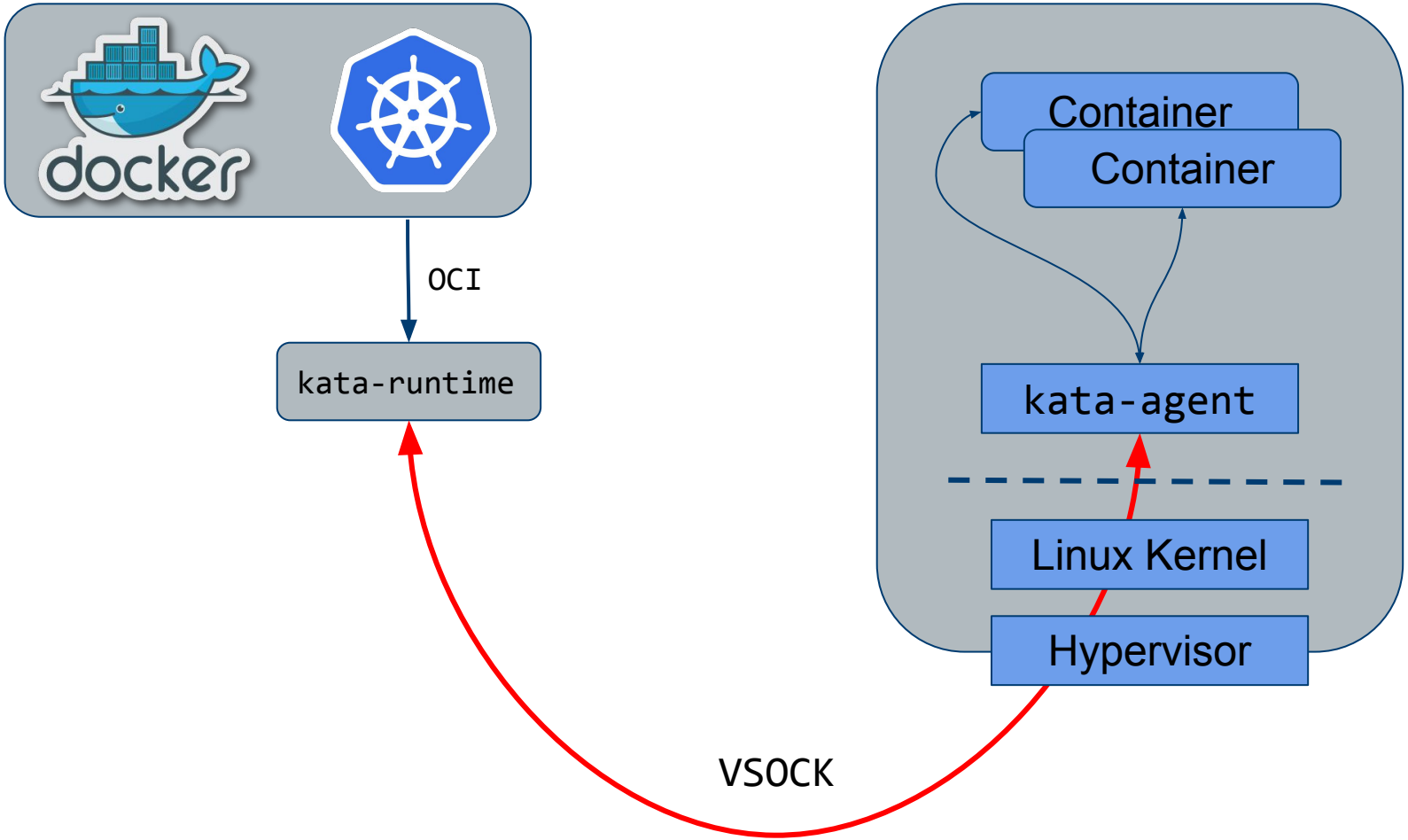

docker
OCI
kata-runtime

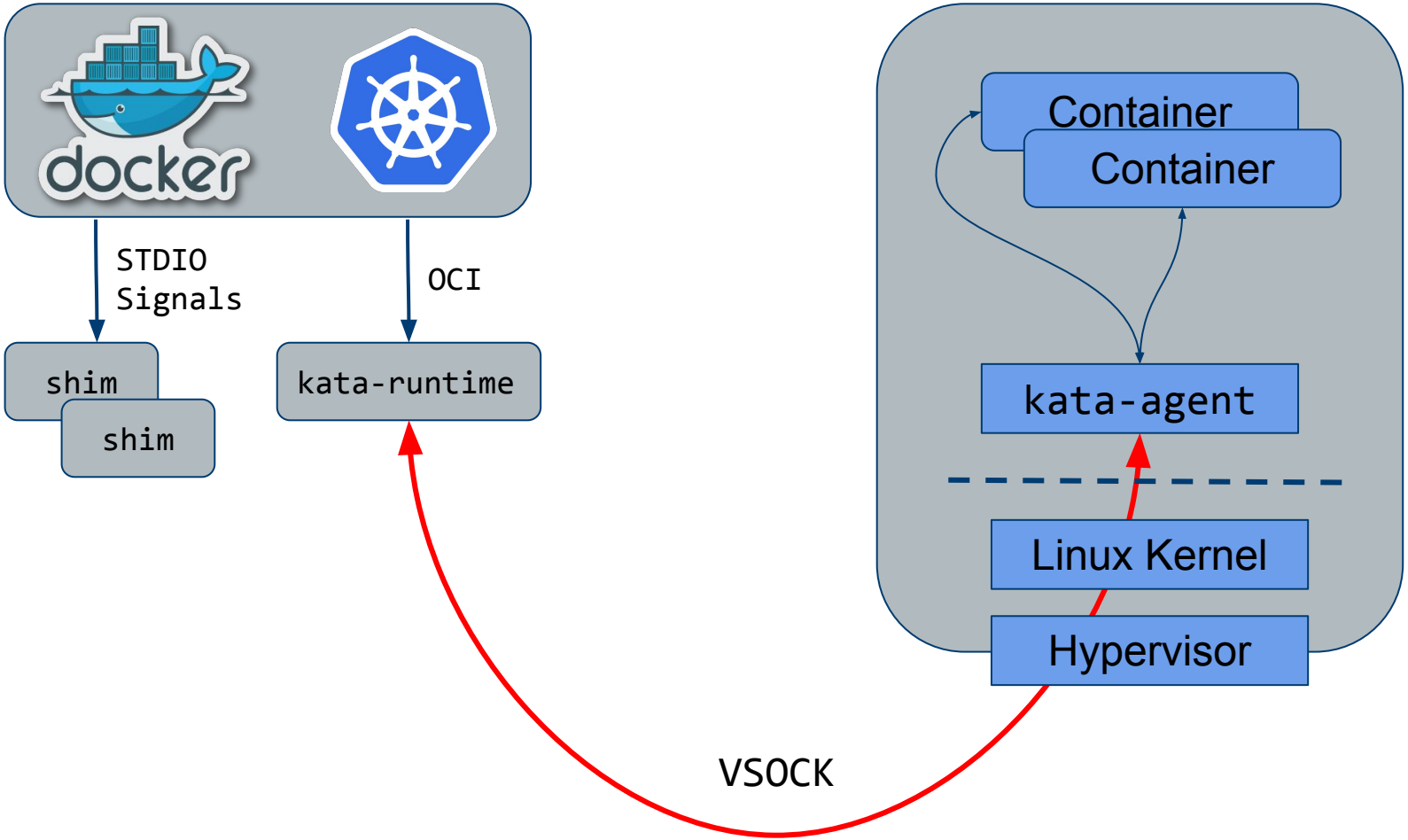

docker
OCI
kata-runtime
Hypervisor

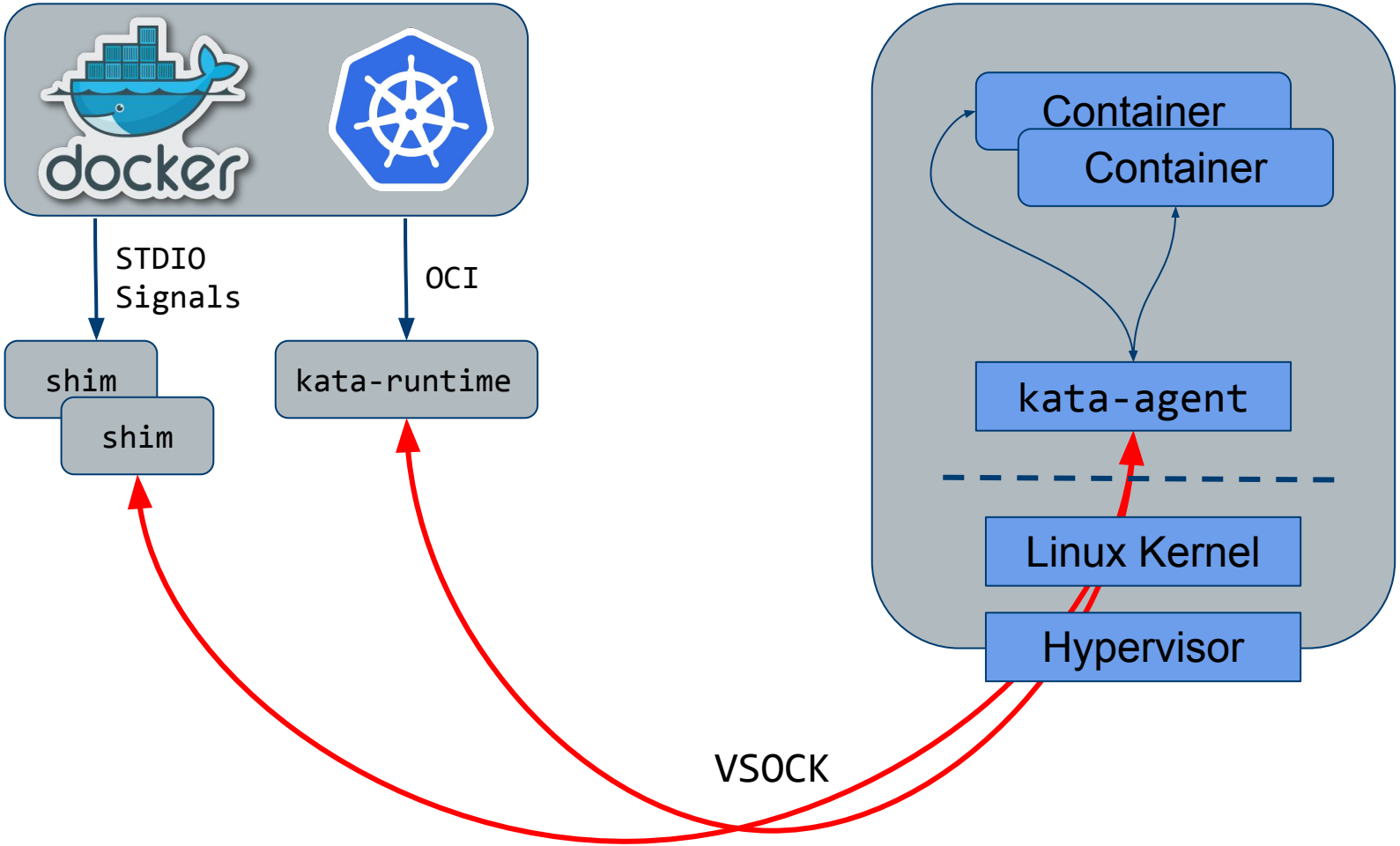

docker
OCI
kata-runtime
Linux Kernel
Hypervisor


docker
OCI
kata-runtime
kata-agent
Linux Kernel
Hypervisor


docker
OCI
kata-runtime
kata-agent
Linux Kernel
Hypervisor
VSOCK

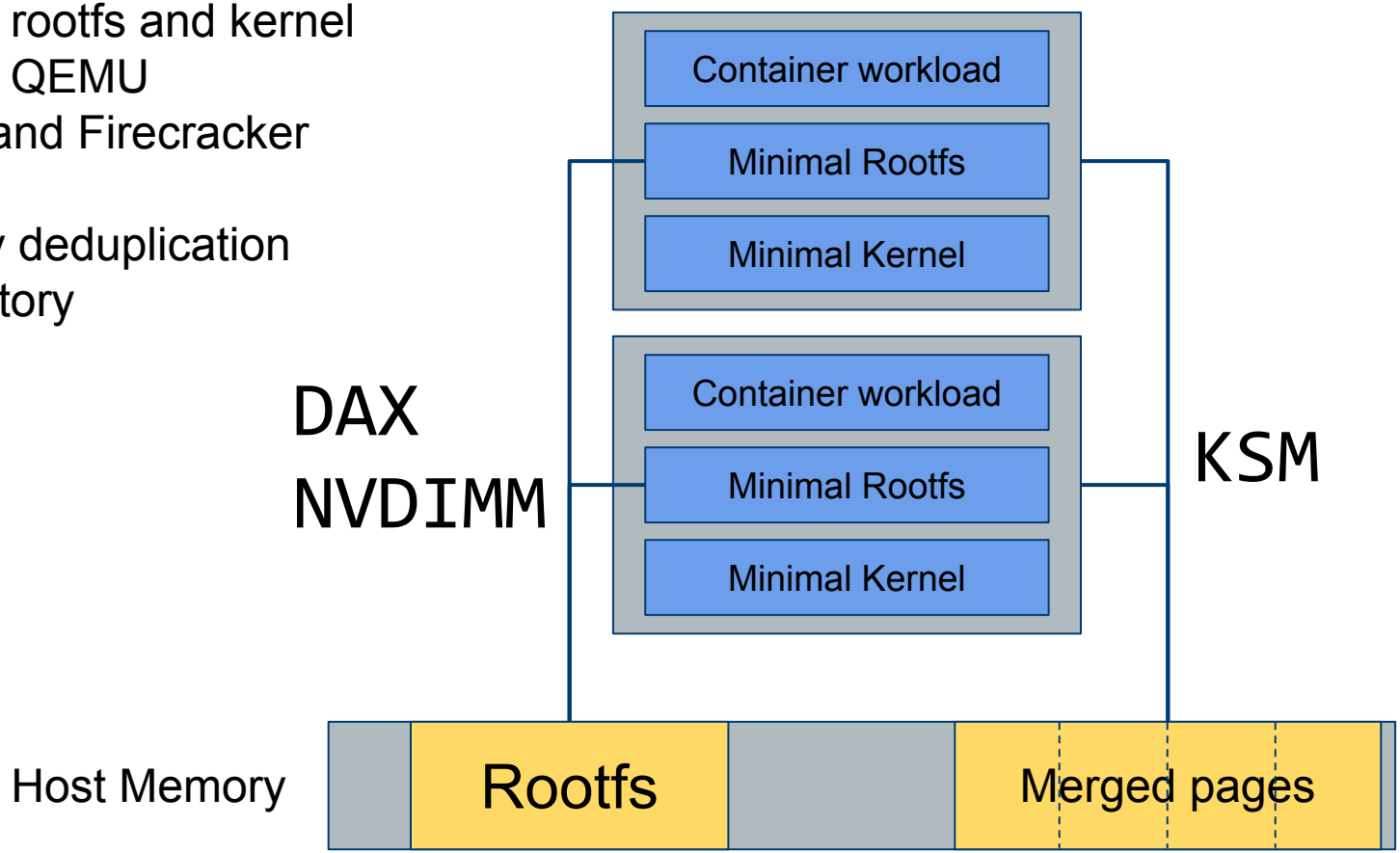

docker
OCI
kata-runtime
Container
Container
kata-agent
Linux Kernel
Hypervisor
VSOCK


docker
STDIO
Signals
OCI
shim
shim
kata-runtime
Container
Container
kata-agent
Linux Kernel
Hypervisor
VSOCK


docker
STDIO
Signals
OCI
shim
shim
kata-runtime
Container
Container
kata-agent
Linux Kernel
Hypervisor
VSOCK

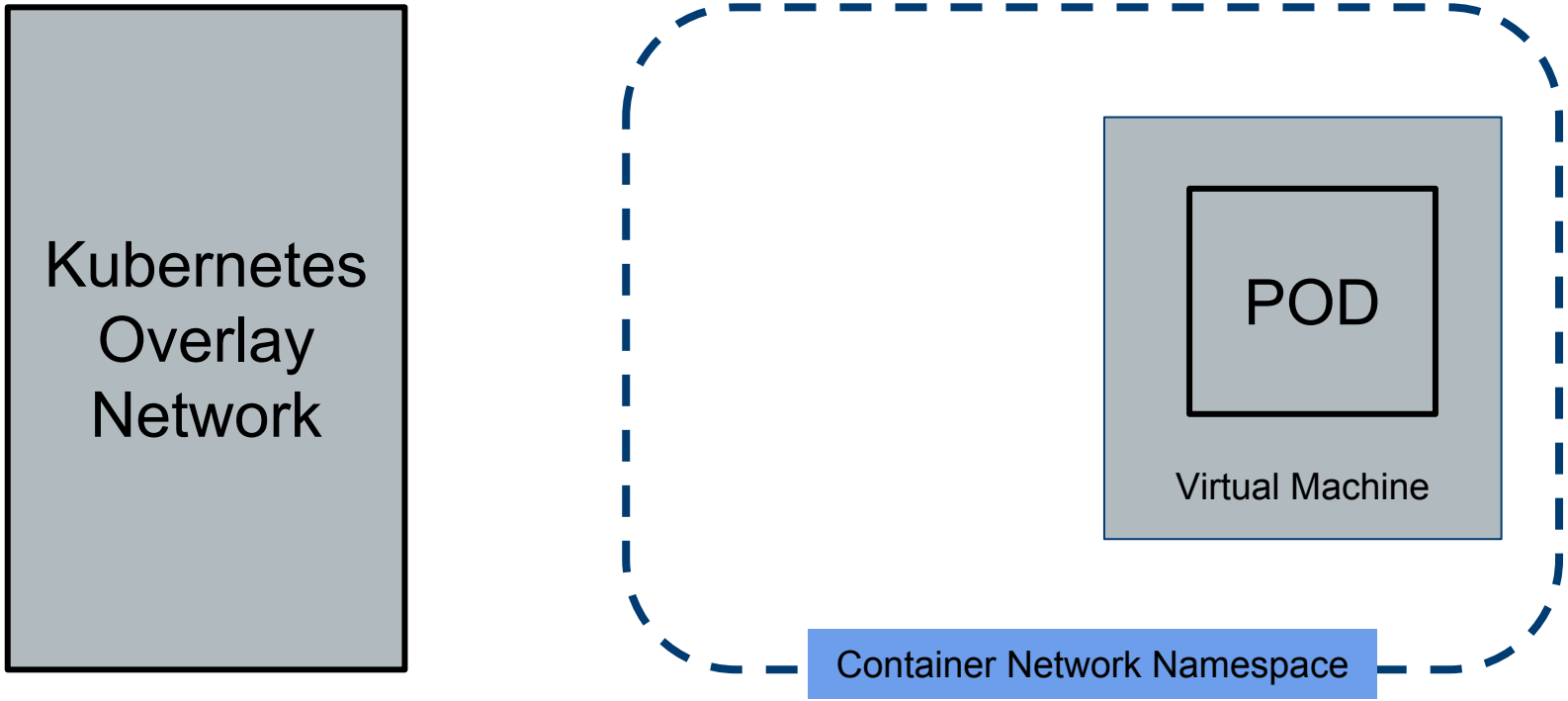
Small and fast like a container

- Minimal rootfs and kernel
- Minimal QEMU
- NEMU and Firecracker
- DAX
- Memory deduplication
- VM Factory

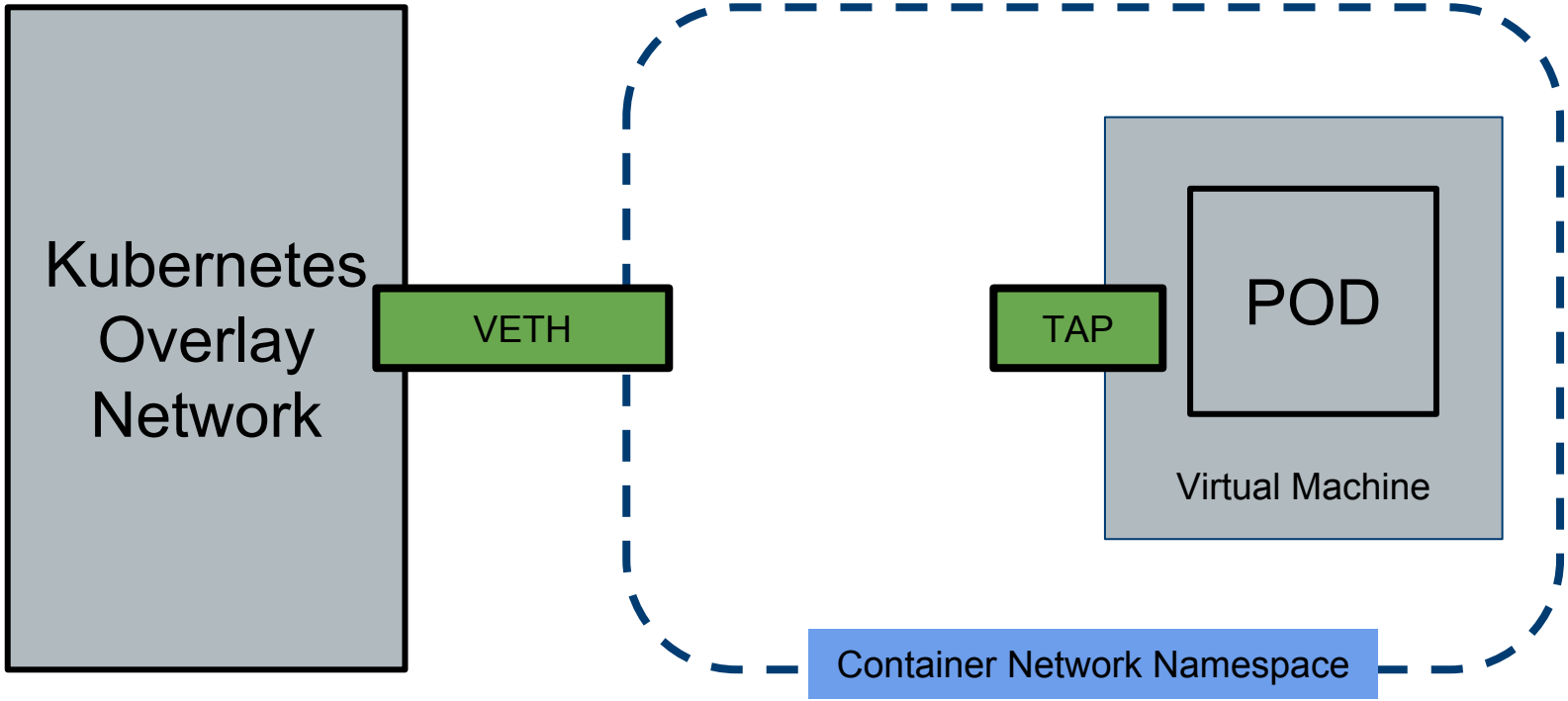


Looks and behave like a container

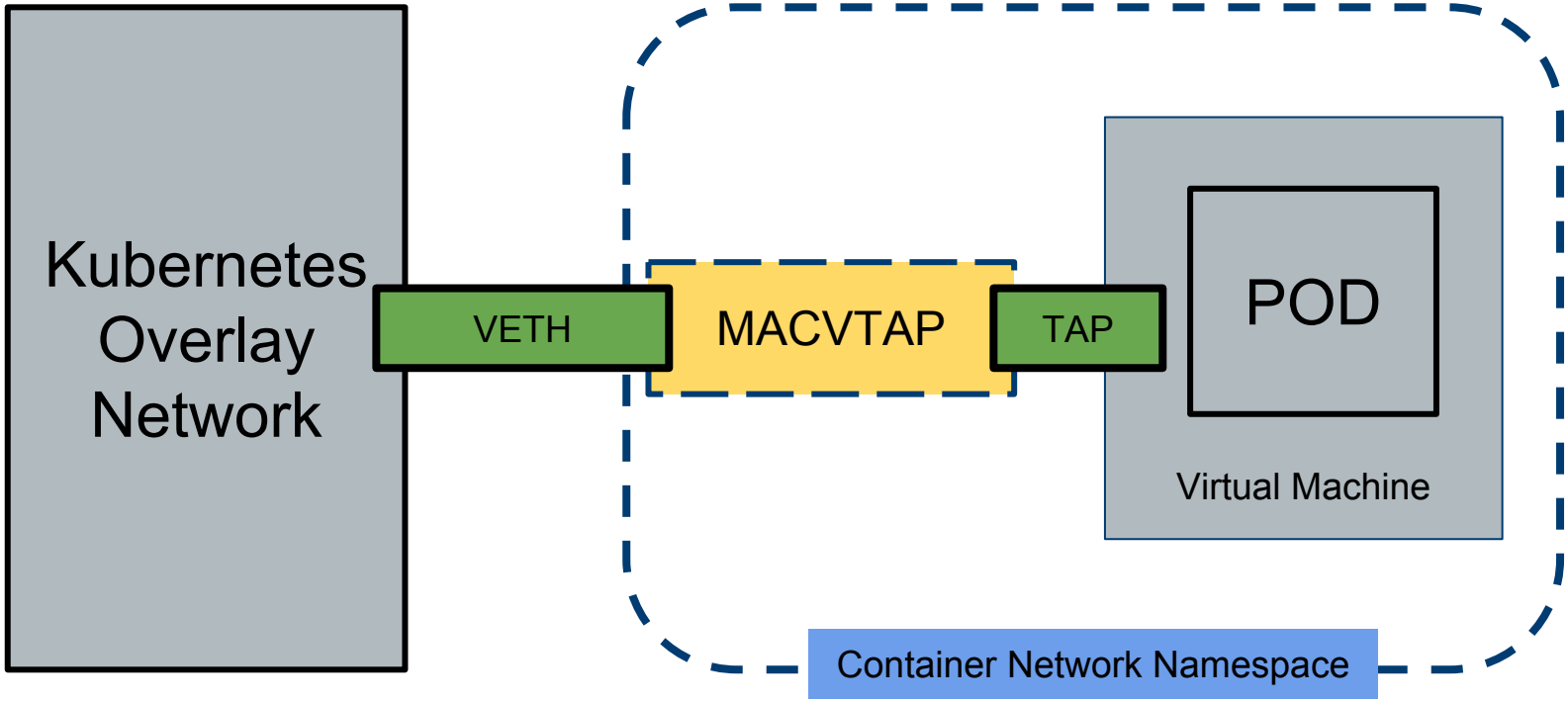
Networking


Kubernetes Overlay Network
POD
Virtual Machine
Container Network Namespace

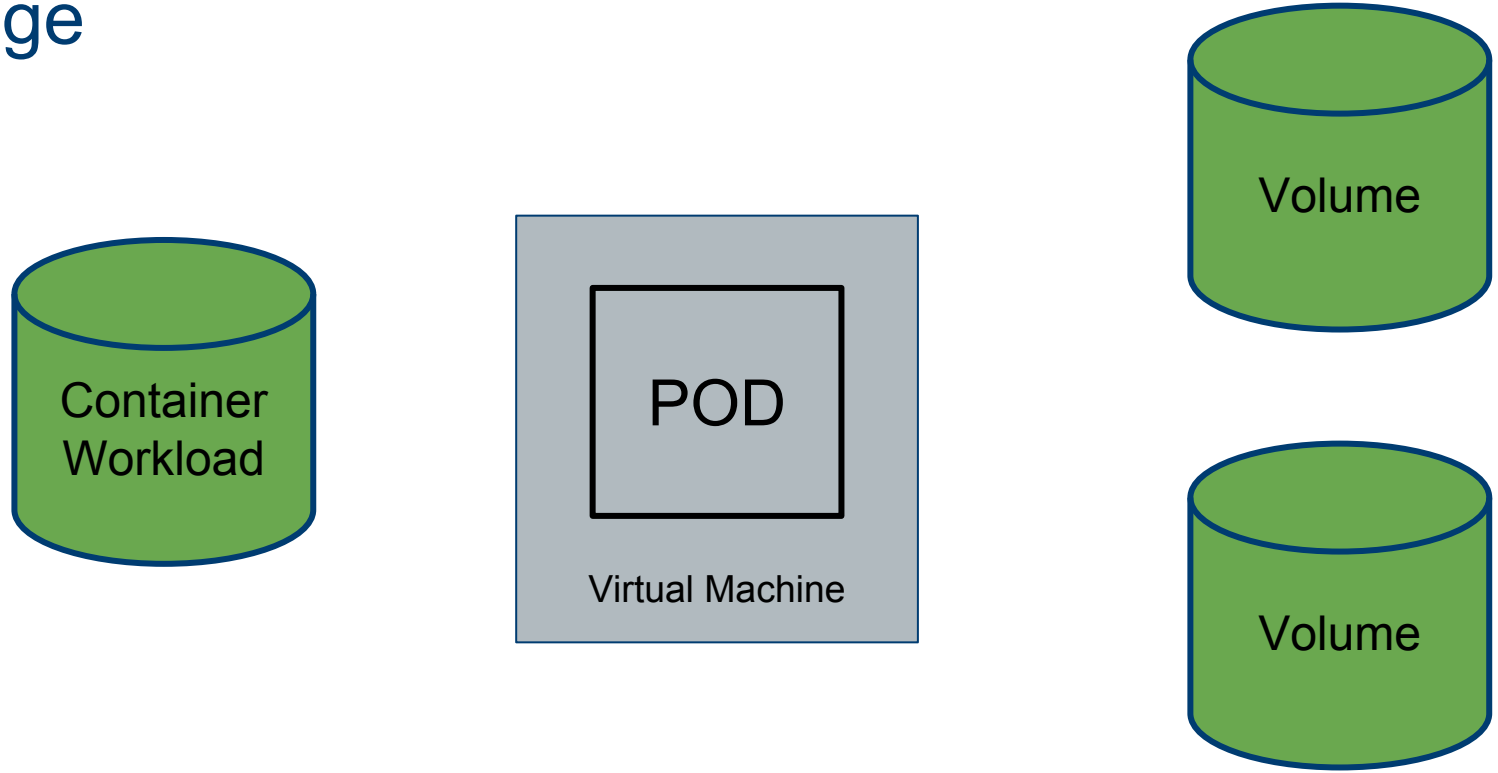
Networking


Kubernetes
Overlay
Network
VETH
TAP
POD
Virtual Machine
Container Network Namespace

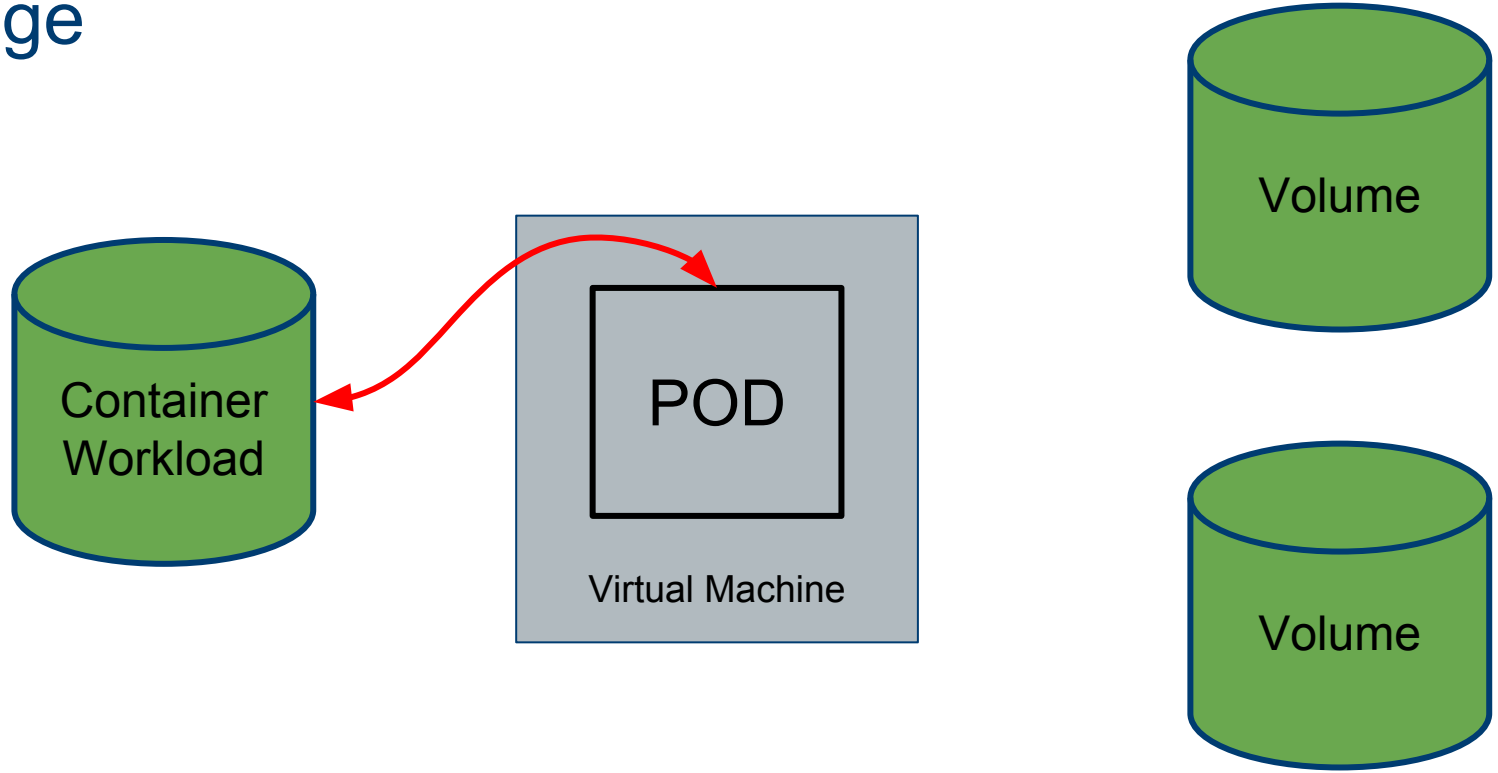
Networking


Kubernetes
Overlay
Network
VETH
MACVTAP
TAP
POD
Virtual Machine
Container Network Namespace

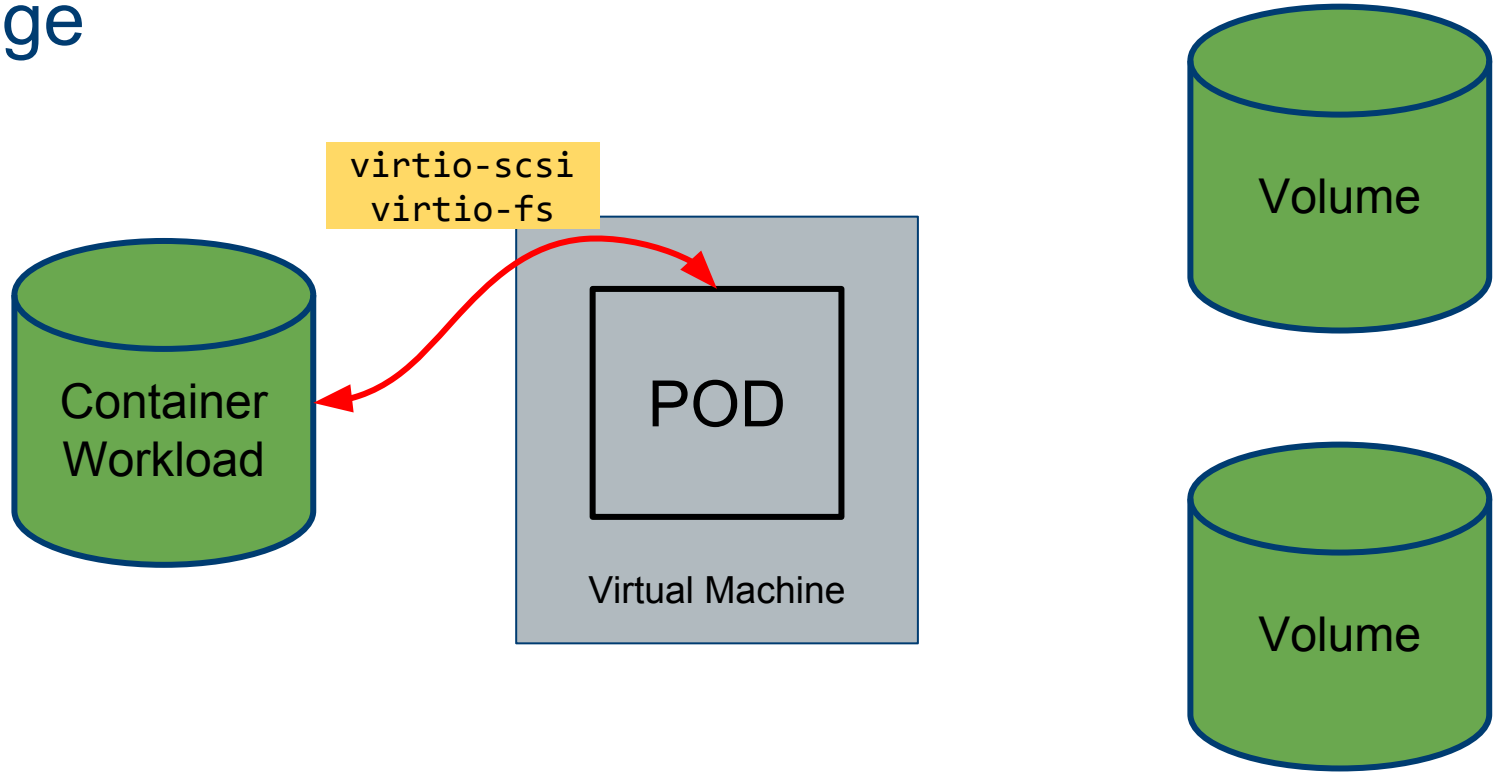
Storage



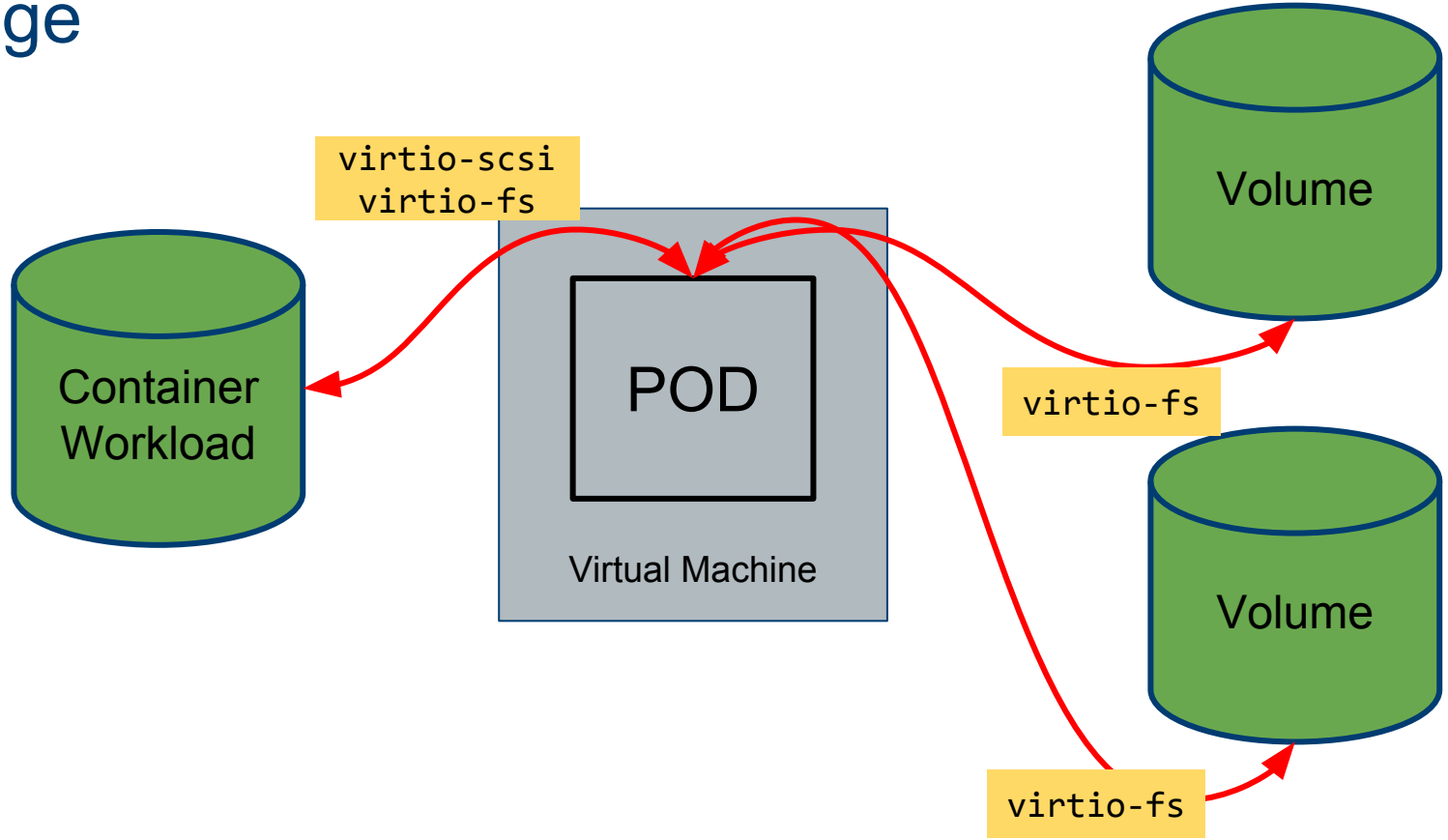
Storage



Storage


virtio-scsi
virtio-fs
Container
Workload
POD
Virtual Machine
Volume
Volume

Storage

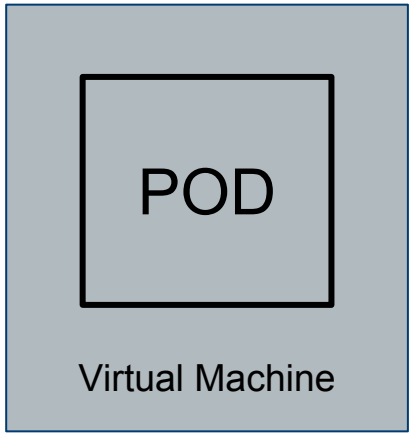

virtio-scsi
virtio-fs
Container Workload
POD
Virtual Machine
Volume
virtio-fs
Volume
virtio-fs

I/O - Devices

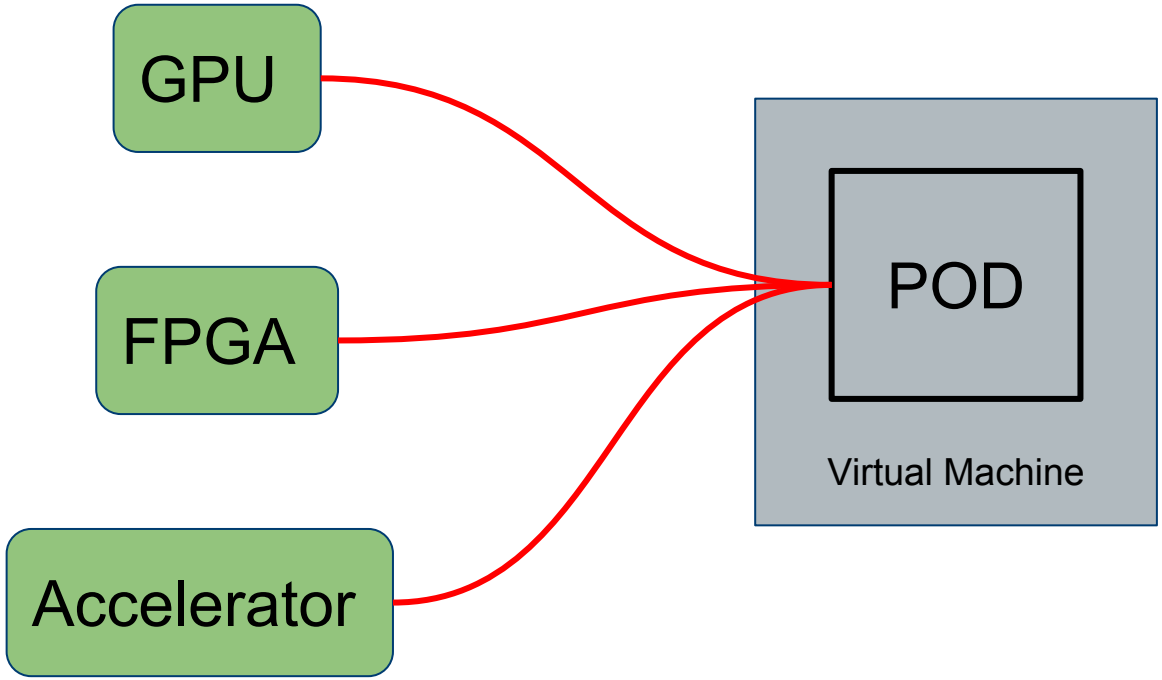
GPU

FPGA

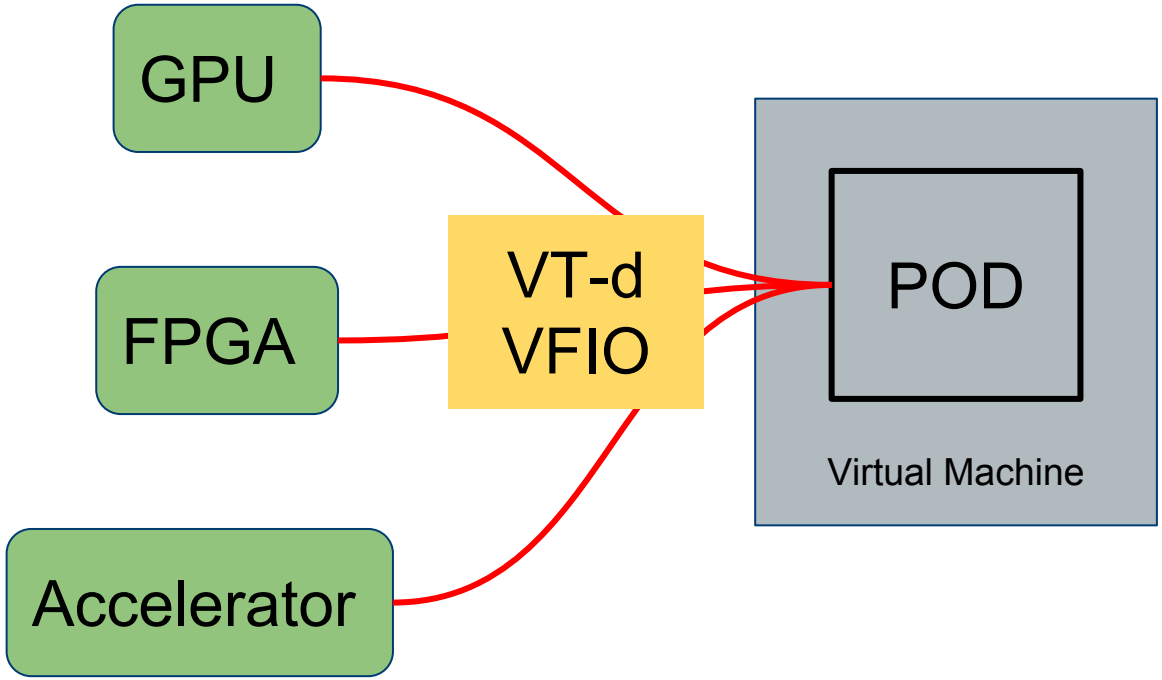
Accelerator



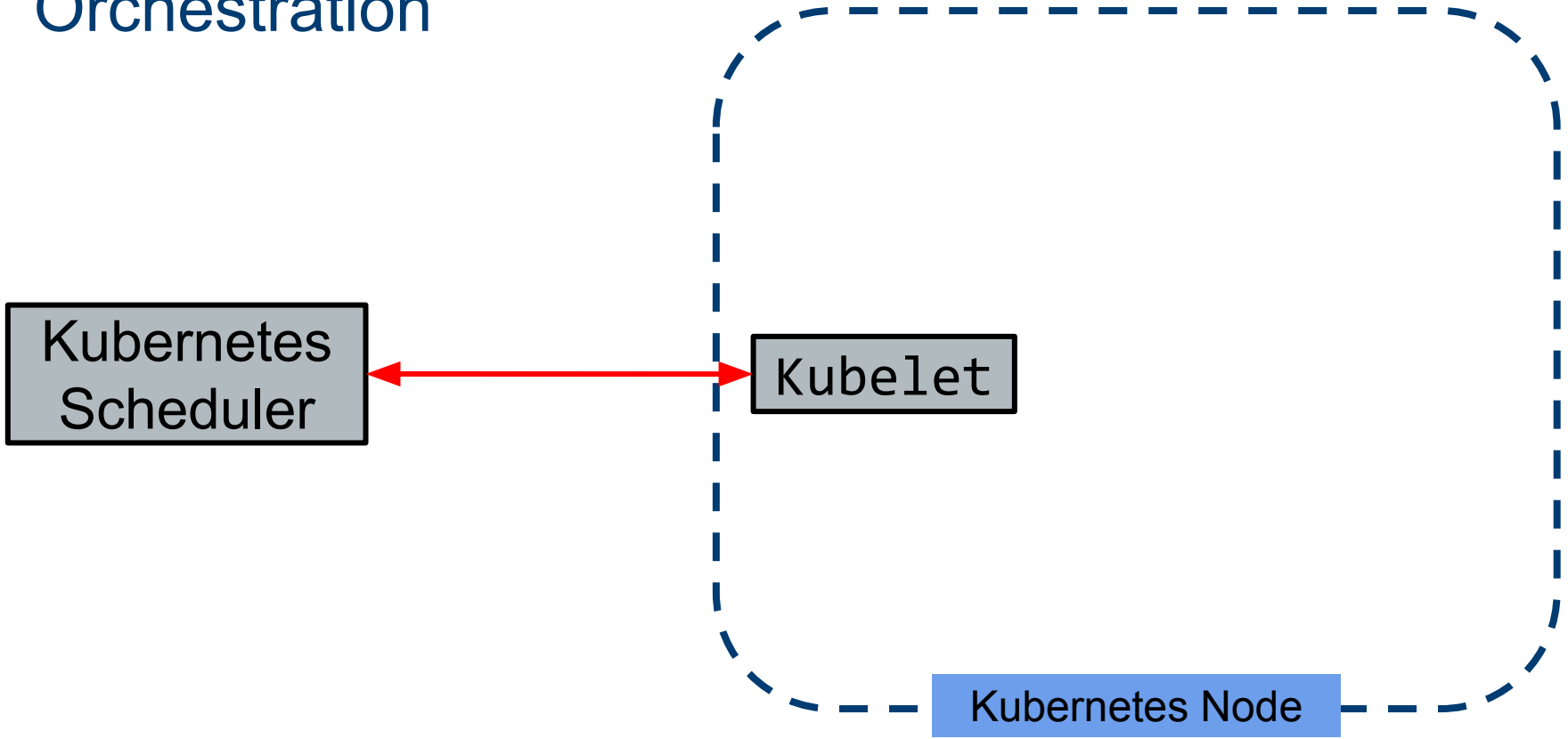
I/O - Devices



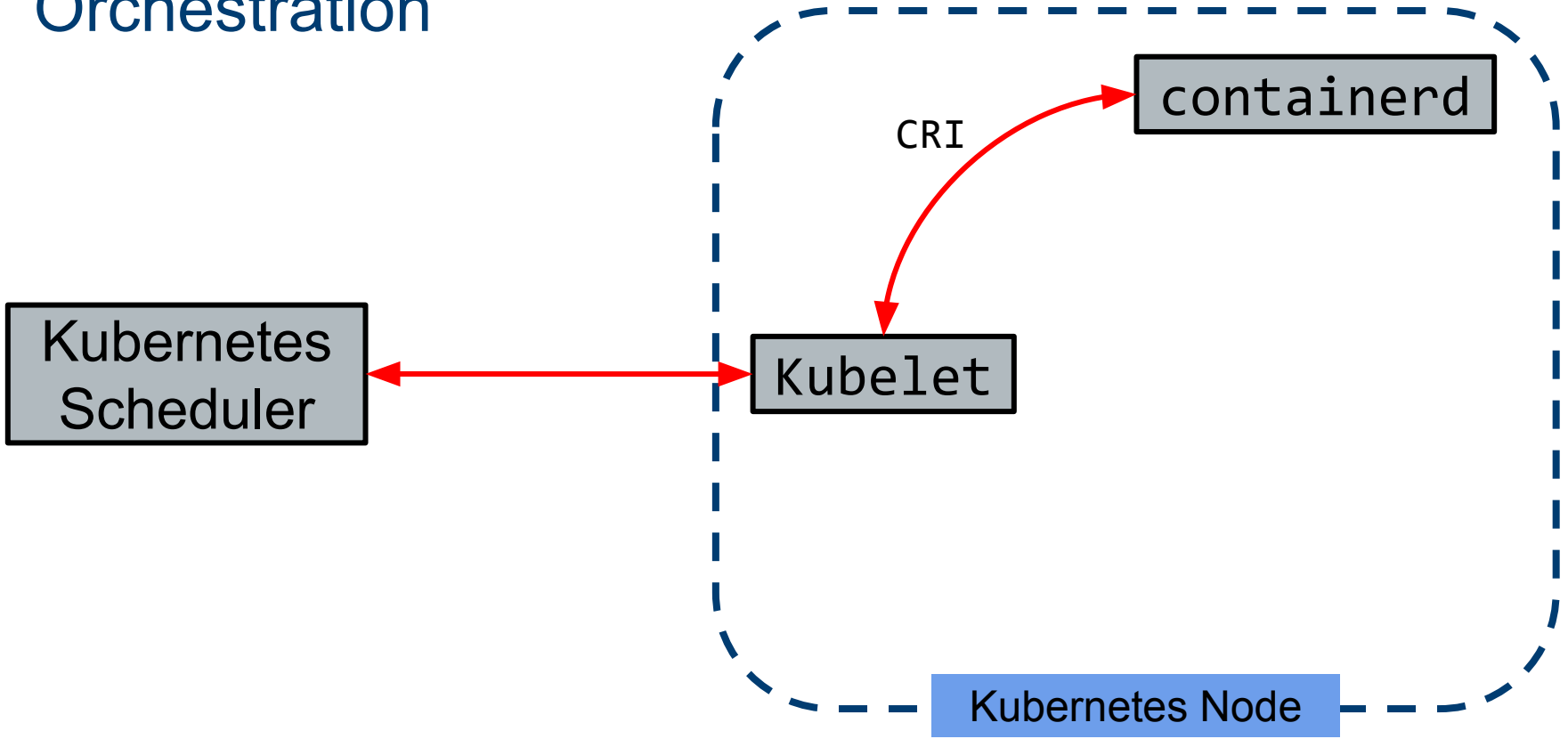
I/O - Devices



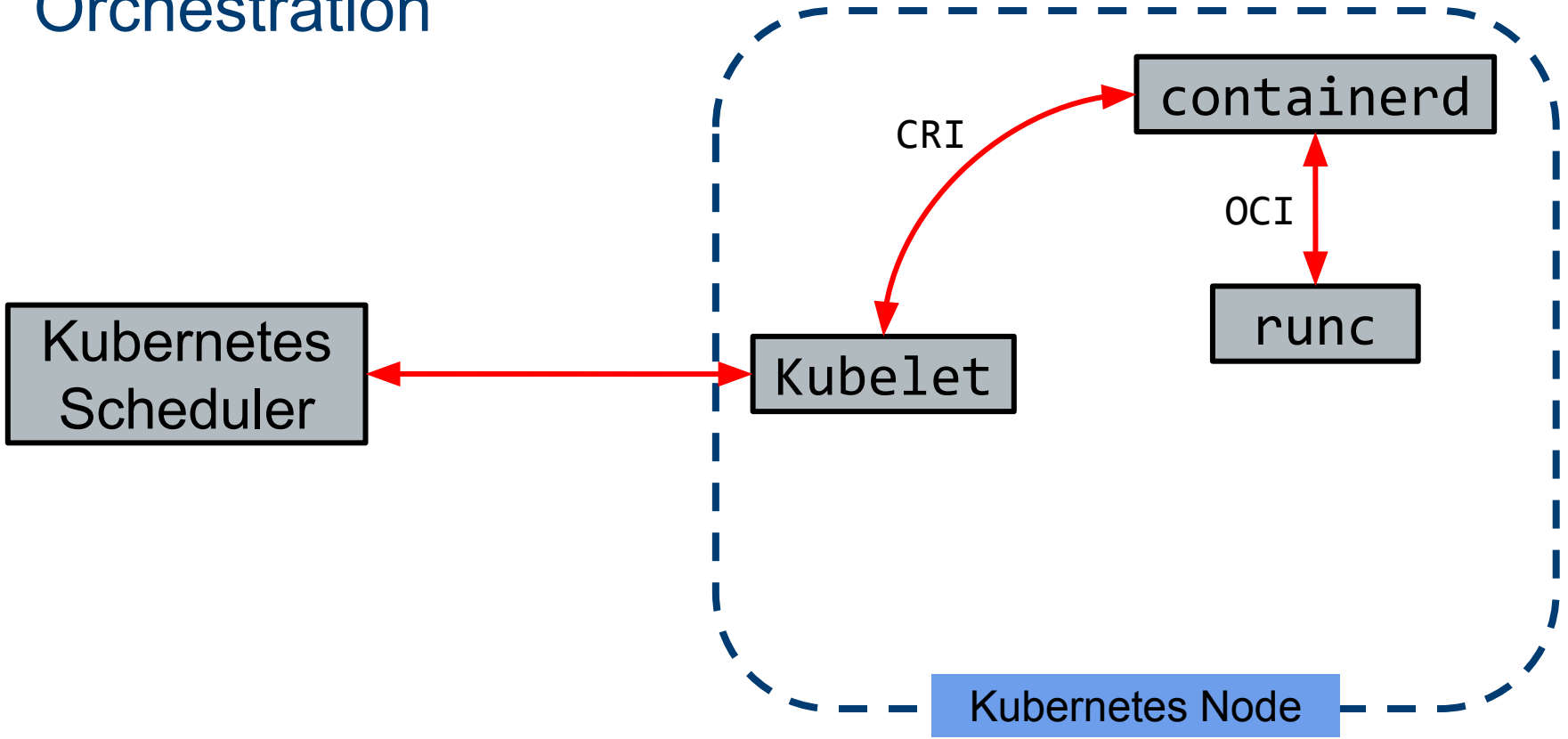
Orchestration



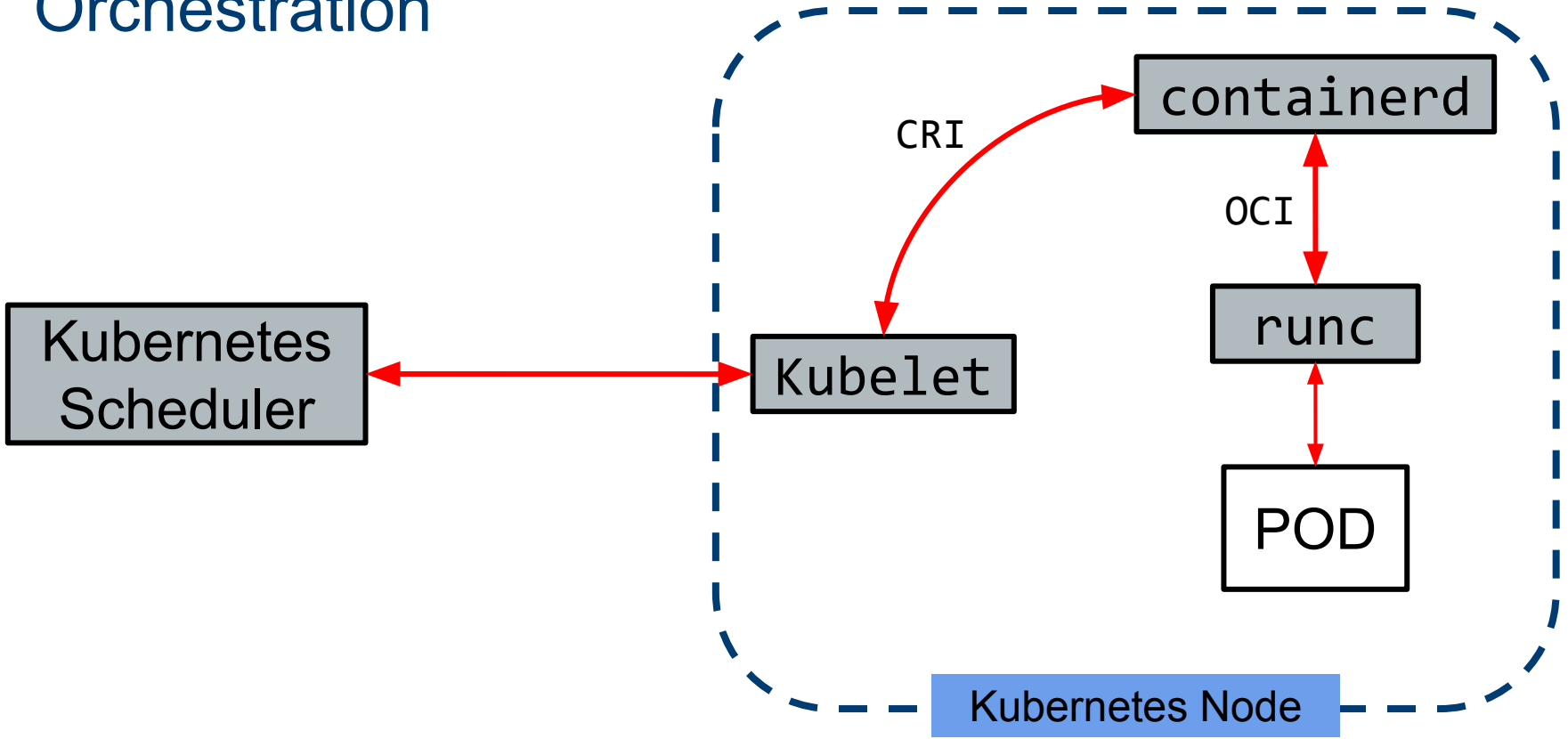
Orchestration



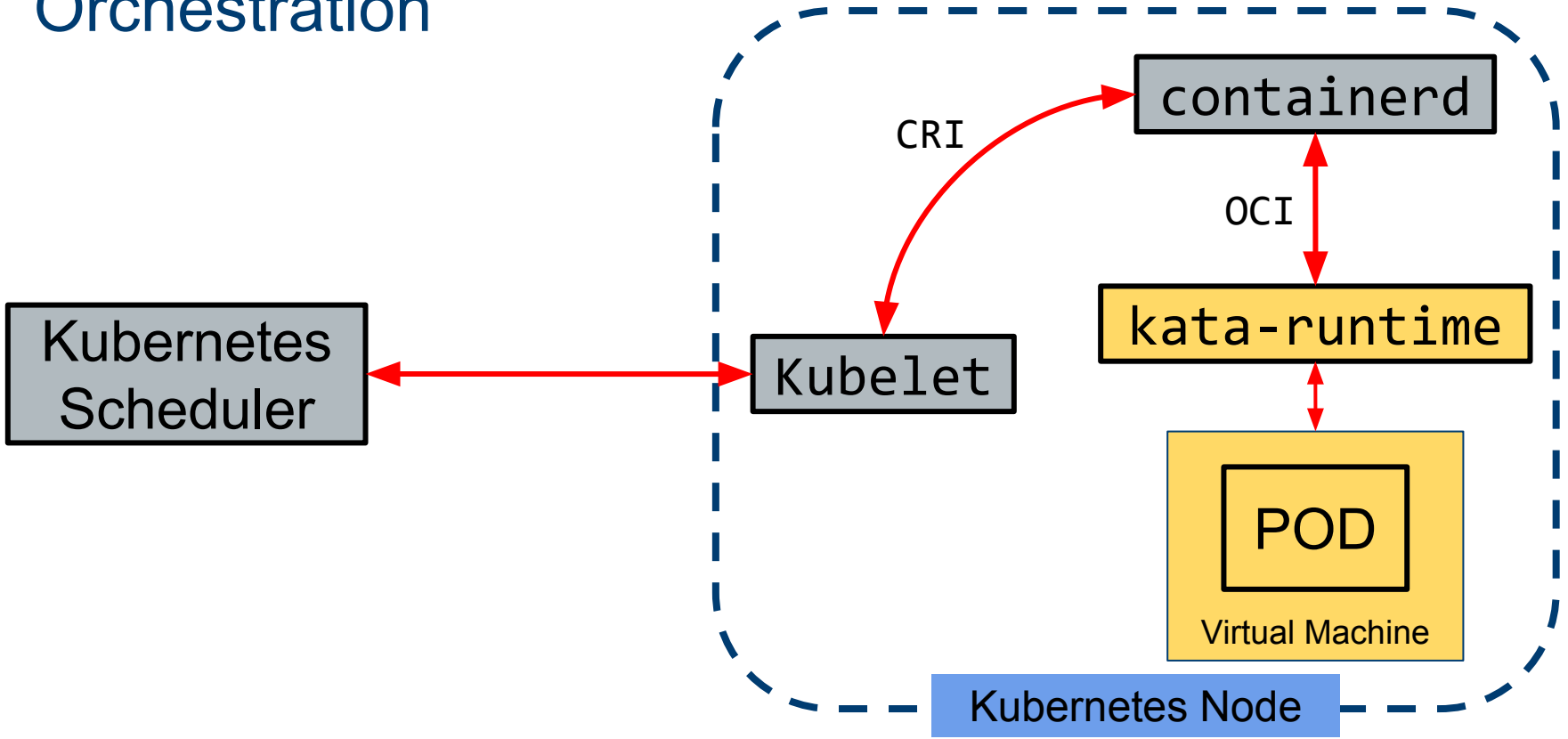
Orchestration



Orchestration



Orchestration



What is it for?

Where does Kata make sense?

Regulated and sensitive production environments

Bare metal infrastructure

Mixed level of trusts

Specific kernel/OS dependencies

OS dependencies

Always run on top of a specified kernel

Always run on top of a specified rootfs

Complete decoupling from host specifications

Container as a Service (CaaS)

Bare metal infrastructure

Mixed levels of trust

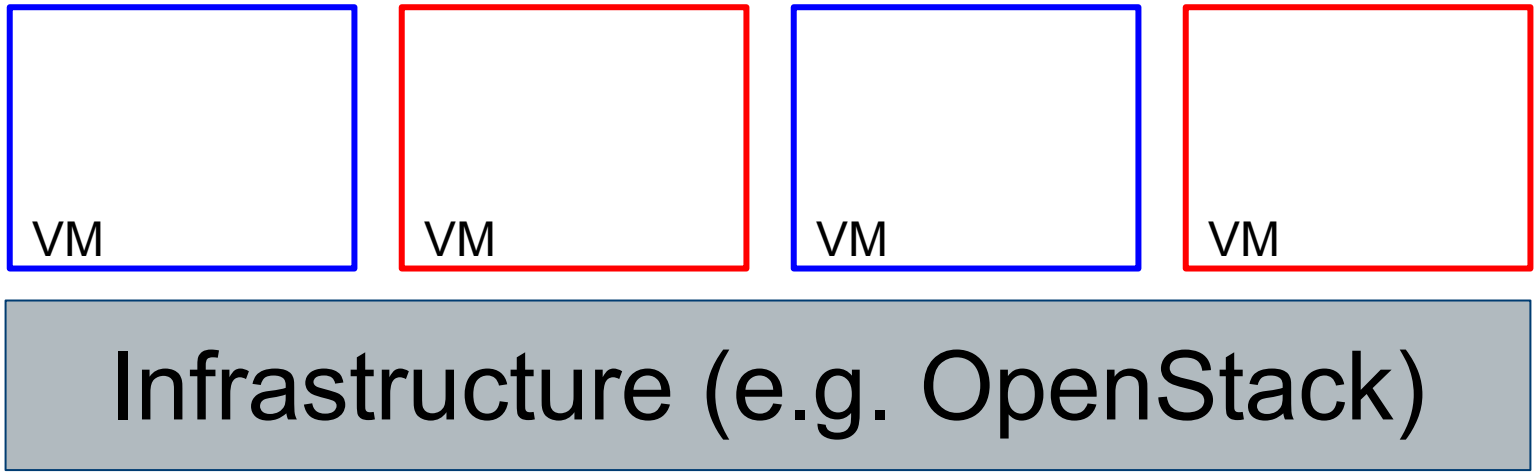
Container as a Service (CaaS)

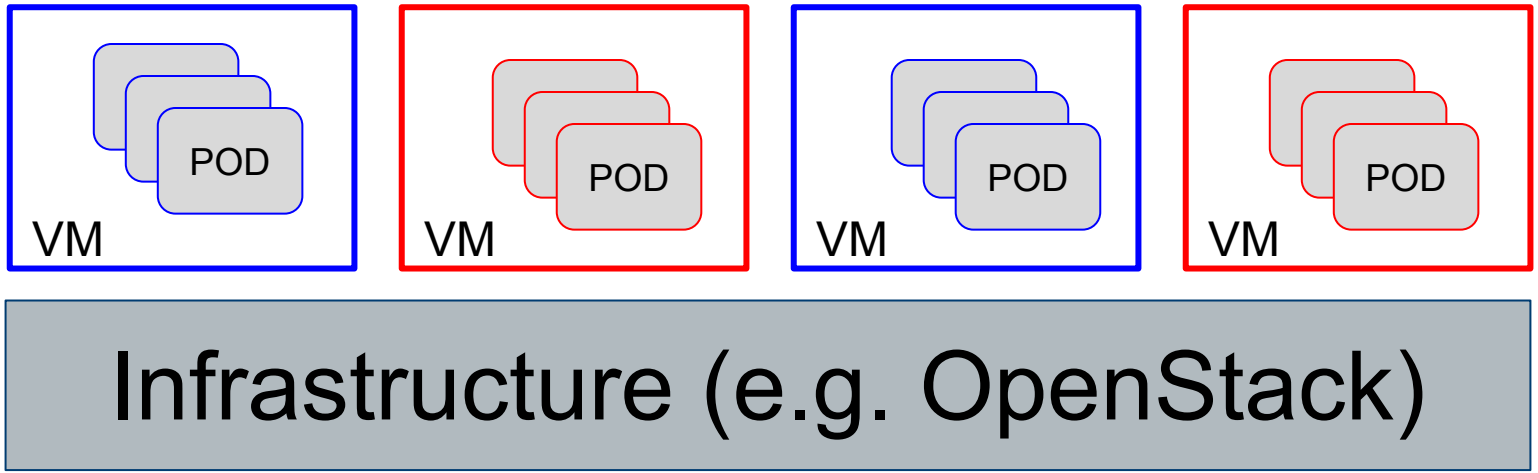
Bare metal infrastructure

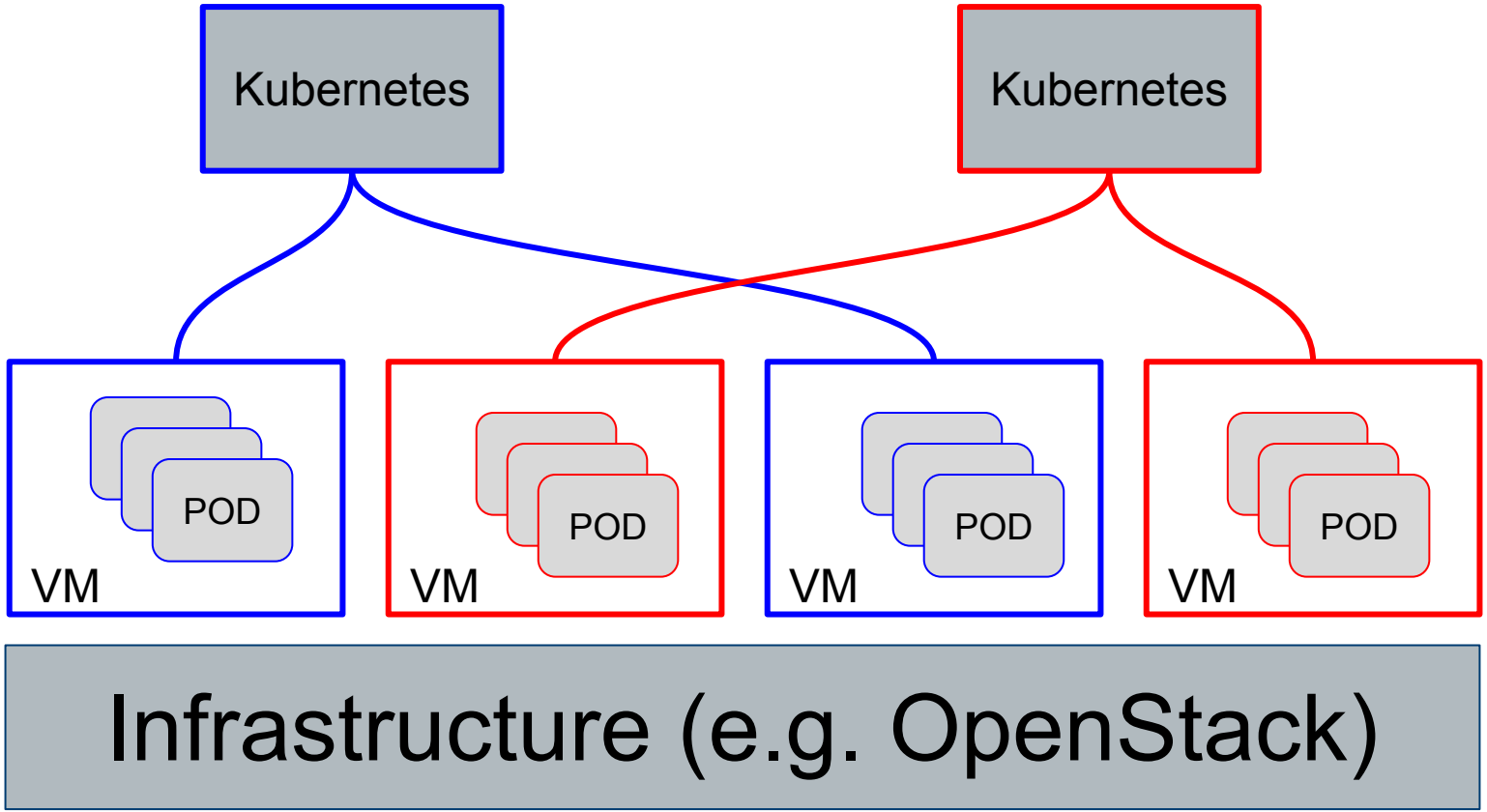
Mixed levels of trust

- Multi-tenant
- Untrusted workloads

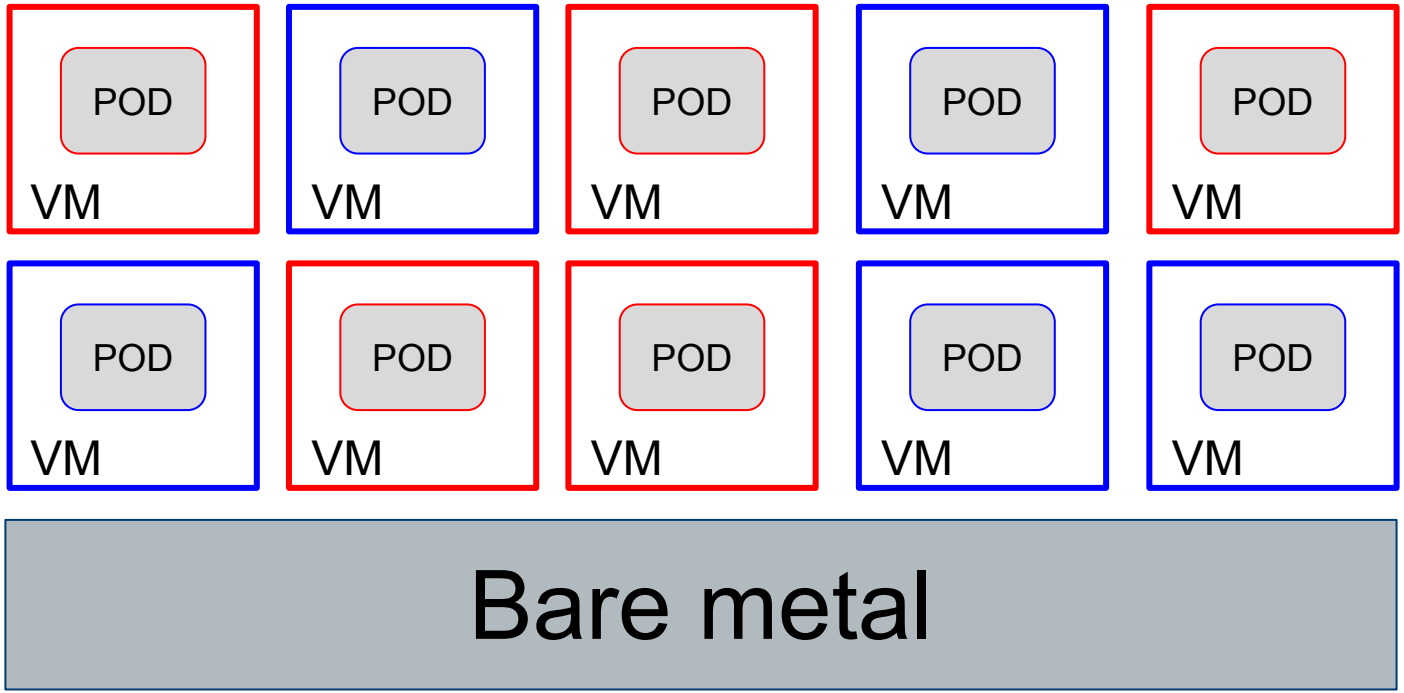
Infrastructure (e.g. OpenStack)

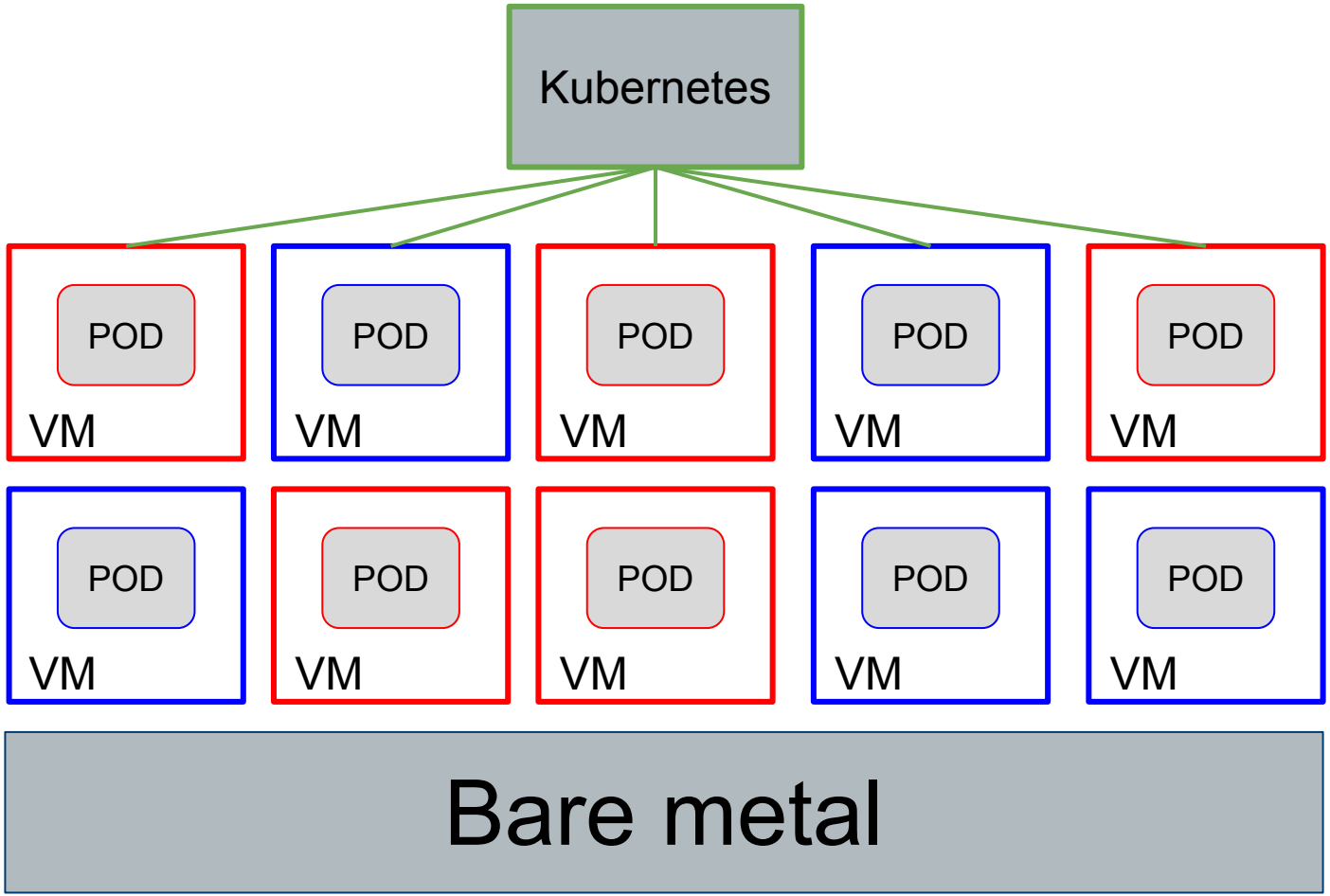

VM
VM
VM
VM
Infrastructure (e.g. OpenStack)


POD
VM
POD
VM
POD
VM
POD
VM
Infrastructure (e.g. OpenStack)


Kubernetes
Kubernetes
POD
POD
POD
POD
VM
VM
VM
VM
Infrastructure (e.g. OpenStack)

Bare metal


POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
Bare metal


Kubernetes
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
POD
VM
Bare metal

Who is running it?

Open Source

Open Governance

OpenStack Foundation Umbrella

Committee: Google, Huawei, Hyper, Intel, Microsoft

Contributors: AMD, ARM, IBM, Nvidia, RedHat, Suse, etc

Latest: Kata 1.5

Periodic Releases (1.6-rc1)

Kata 2.0

https://katacontainers.io

https://github.com/kata-containers

samuel.ortiz@intel.com


intel
experience
what's inside™