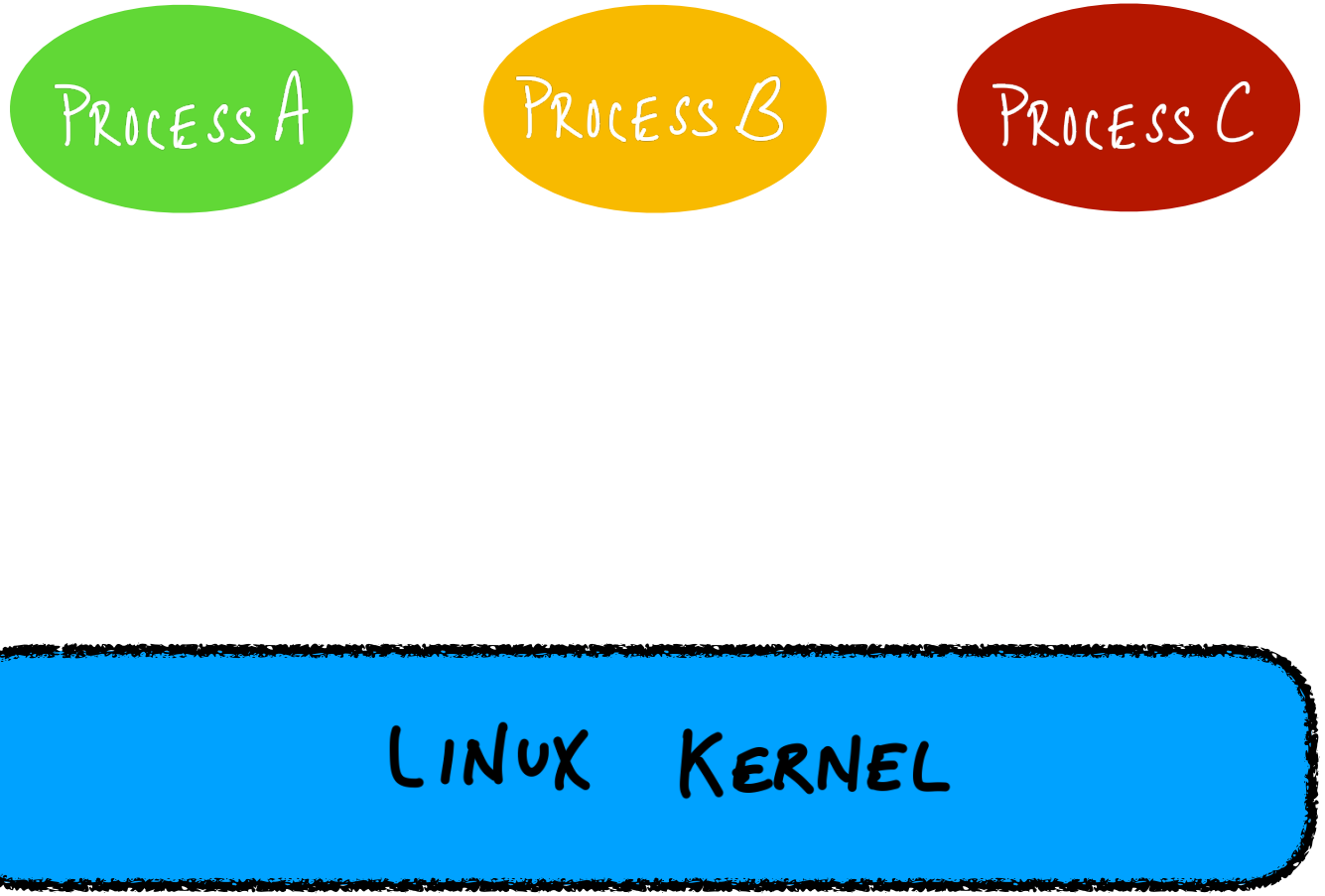
Kata Containers: the speed of containers, security of VMs - even in a nested environment!

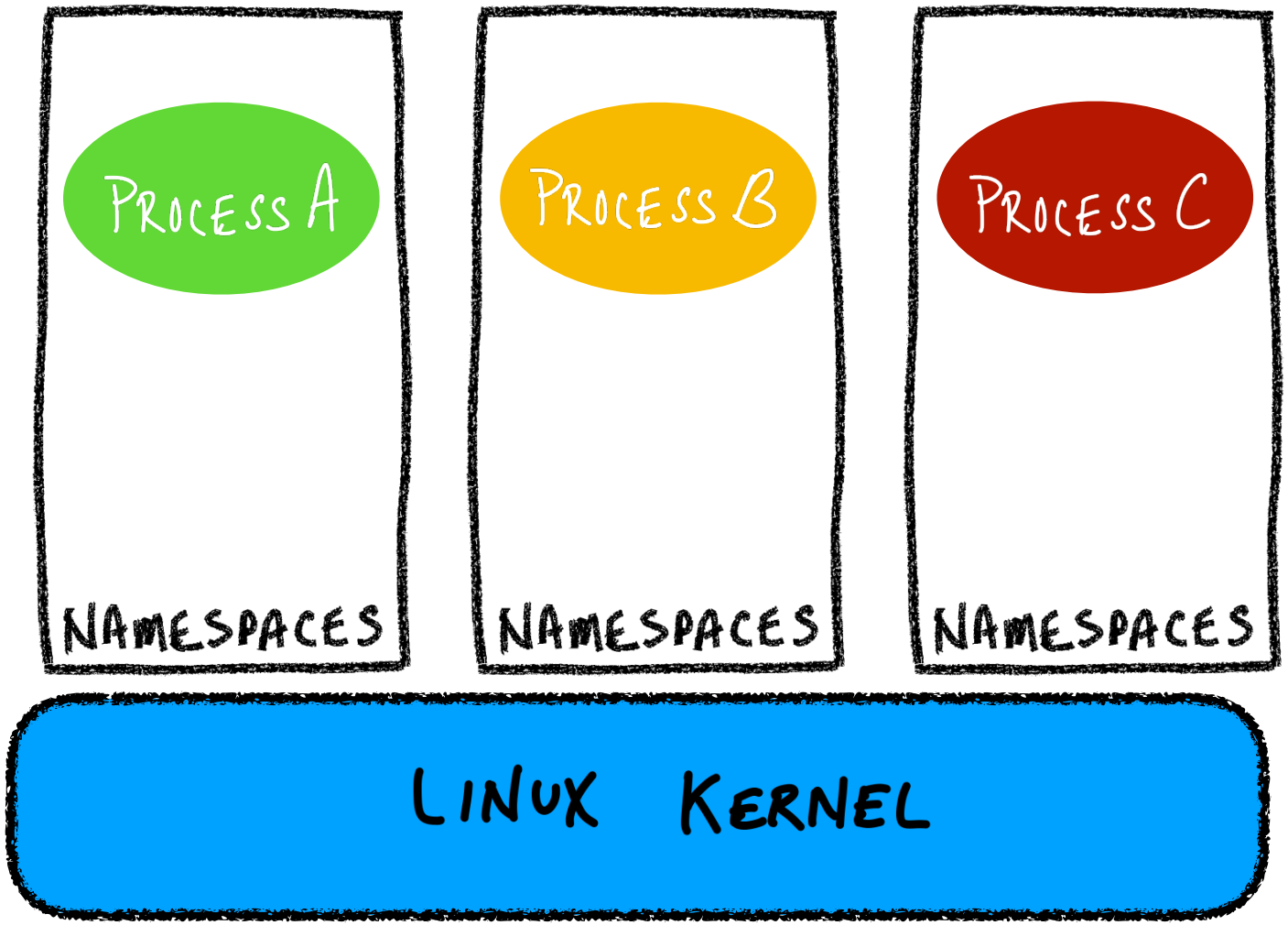
Eric Ernst, Intel
K. Y Srinivasan, Microsoft
Shiny Sebastian, Intel

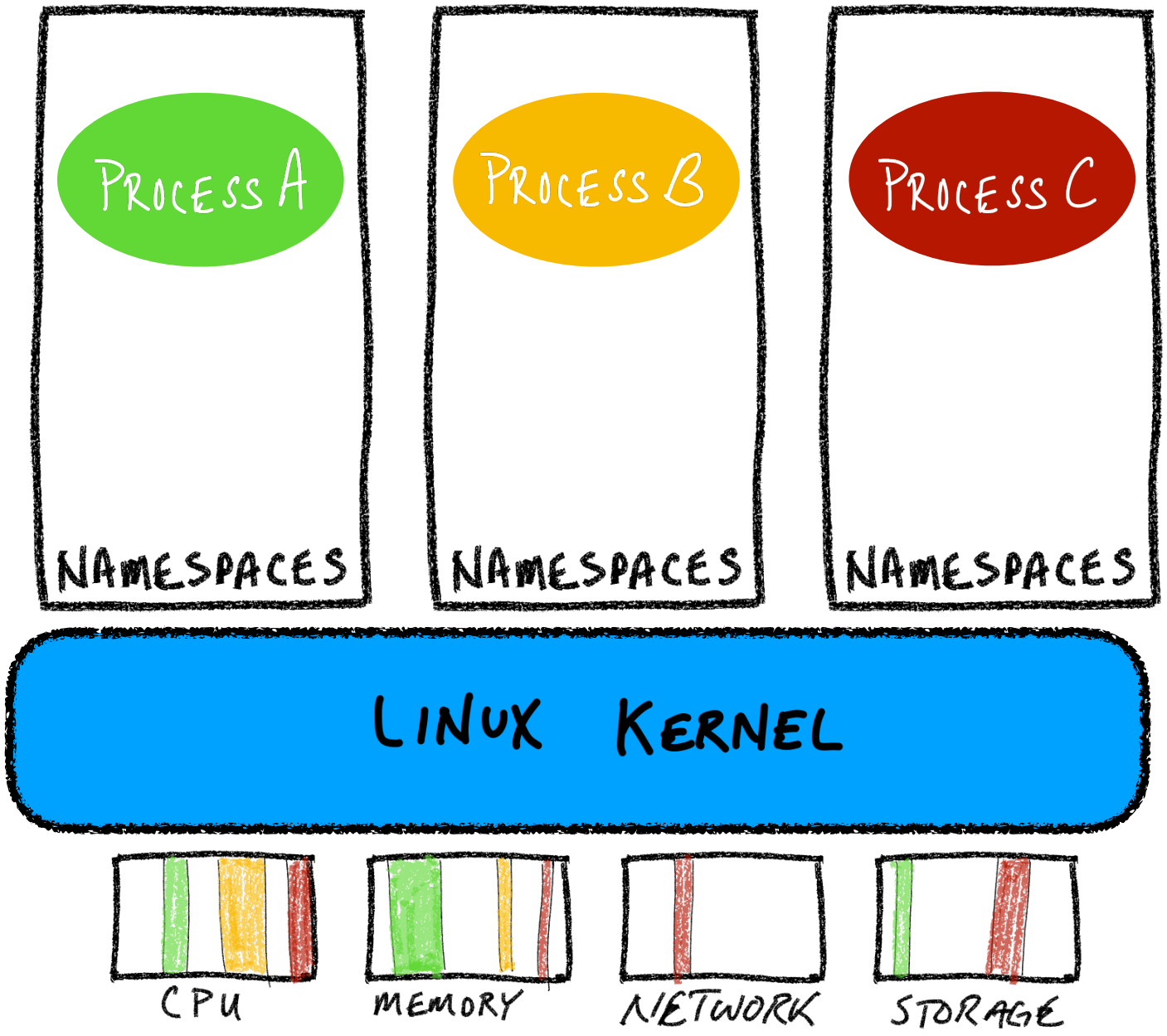
THE LINUX FOUNDATION

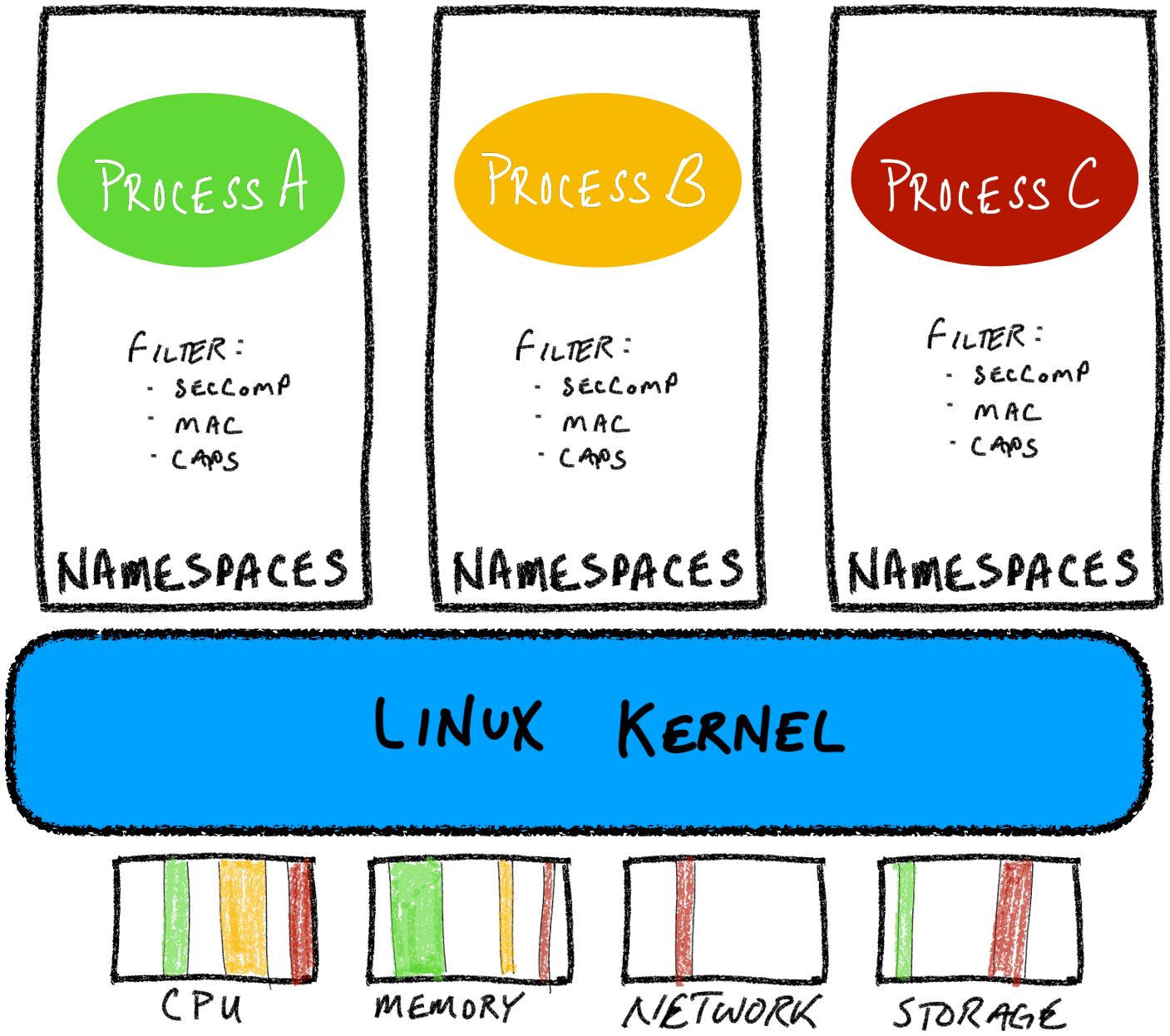
Agenda

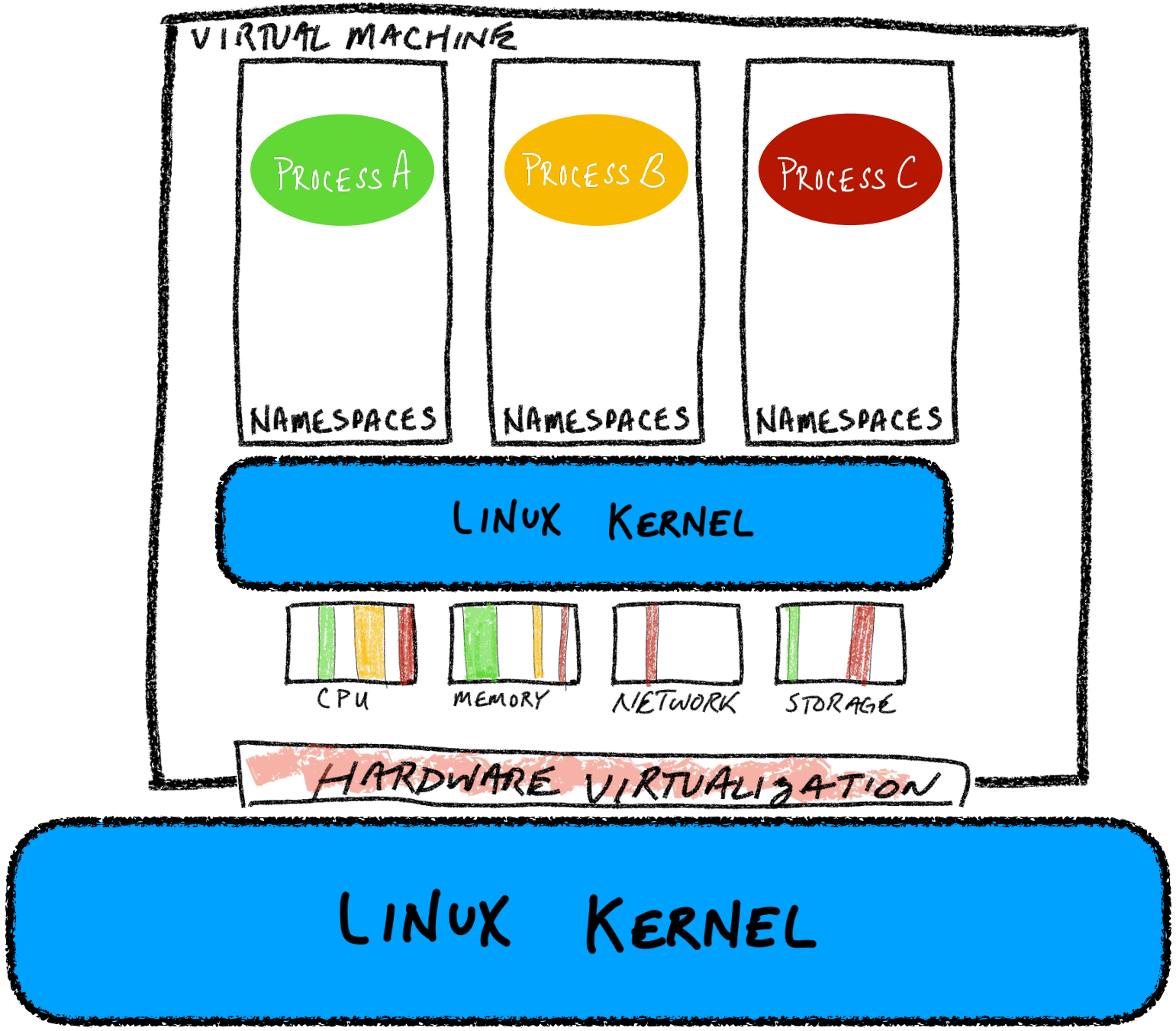
- Overview of Kata
- Nested use case
- KVM on Hyper-V
- A look at Kata nested

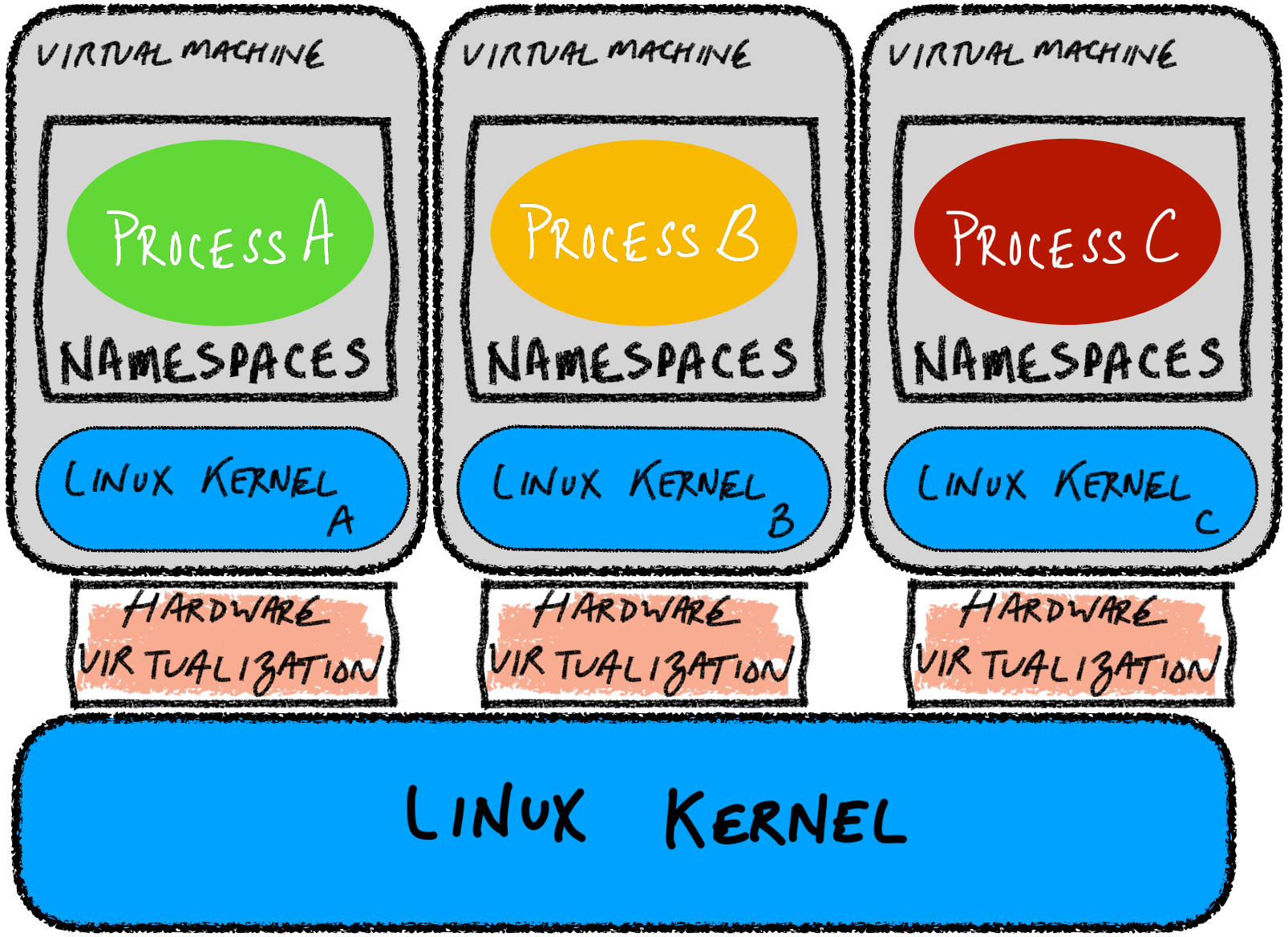

PROCESS A
PROCESS B
PROCESS C
LINUX KERNEL


PROCESS A
PROCESS B
PROCESS C
NAMESPACES
NAMESPACES
NAMESPACES
LINUX KERNEL


PROCESS A
PROCESS B
PROCESS C
NAMESPACES
NAMESPACES
NAMESPACES
LINUX KERNEL
CPU
MEMORY
NETWORK
STORAGE


PROCESS A
FILTER:
- SECCOMP
- MAC
- CAPS
NAMESPACES
PROCESS B
FILTER:
- SECCOMP
- MAC
- CAPS
NAMESPACES
PROCESS C
FILTER:
- SECCOMP
- MAC
- CAPS
NAMESPACES
LINUX KERNEL
CPU
MEMORY
NETWORK
STORAGE


VIRTUAL MACHINE
PROCESS A
PROCESS B
PROCESS C
NAMESPACES
NAMESPACES
NAMESPACES
LINUX KERNEL
CPU
MEMORY
NETWORK
STORAGE
HARDWARE VIRTUALIZATION
LINUX KERNEL


VIRTUAL MACHINE
PROCESS A
NAMESPACES
LINUX KERNEL A
VIRTUAL MACHINE
PROCESS B
NAMESPACES
LINUX KERNEL B
VIRTUAL MACHINE
PROCESS C
NAMESPACES
LINUX KERNEL C
HARDWARE VIRTUALIZATION
HARDWARE VIRTUALIZATION
HARDWARE VIRTUALIZATION
LINUX KERNEL

* VIRTUAL MACHINE CREATED WITH KVM/QEMU

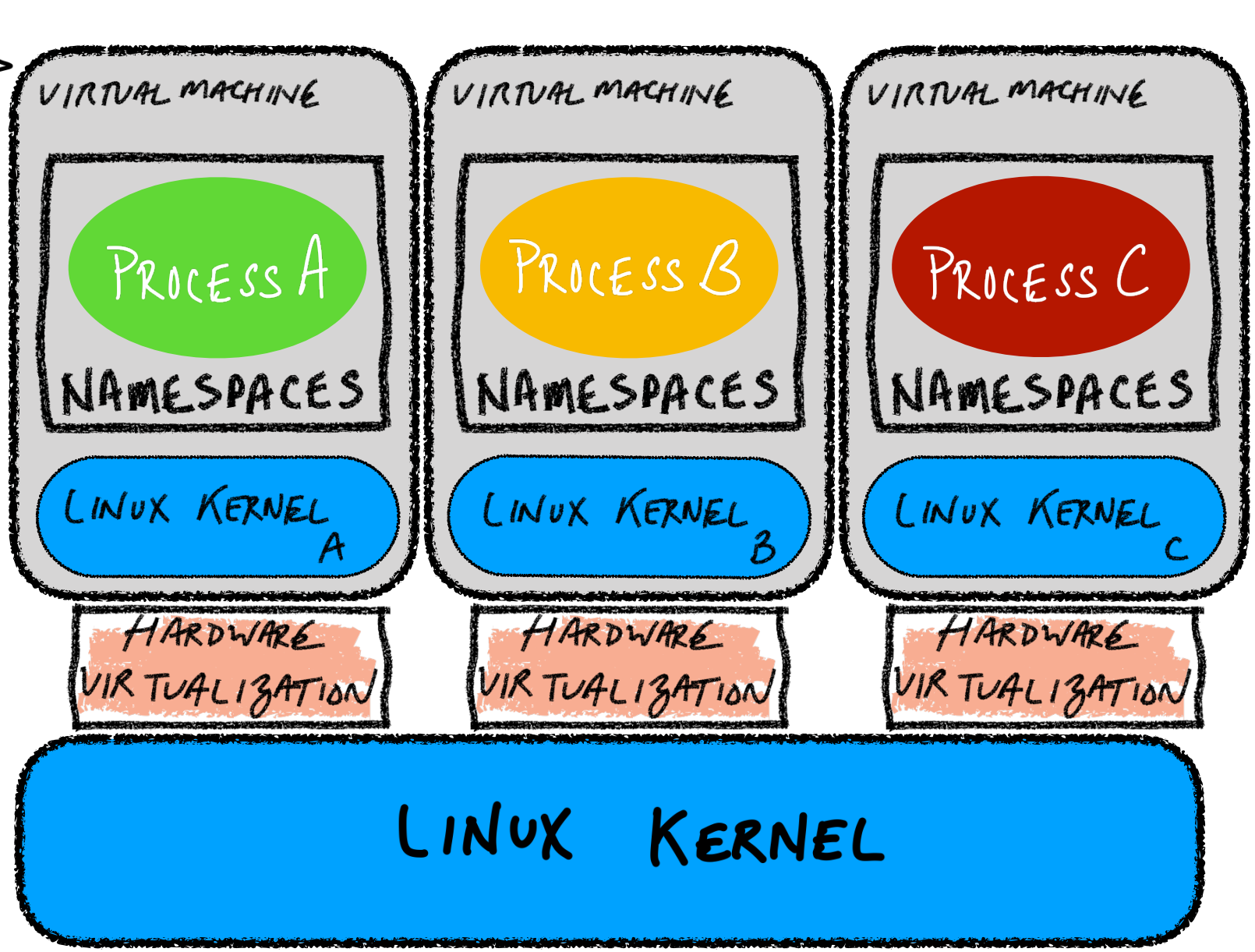
* ACTIVELY WORKING ON NEMU

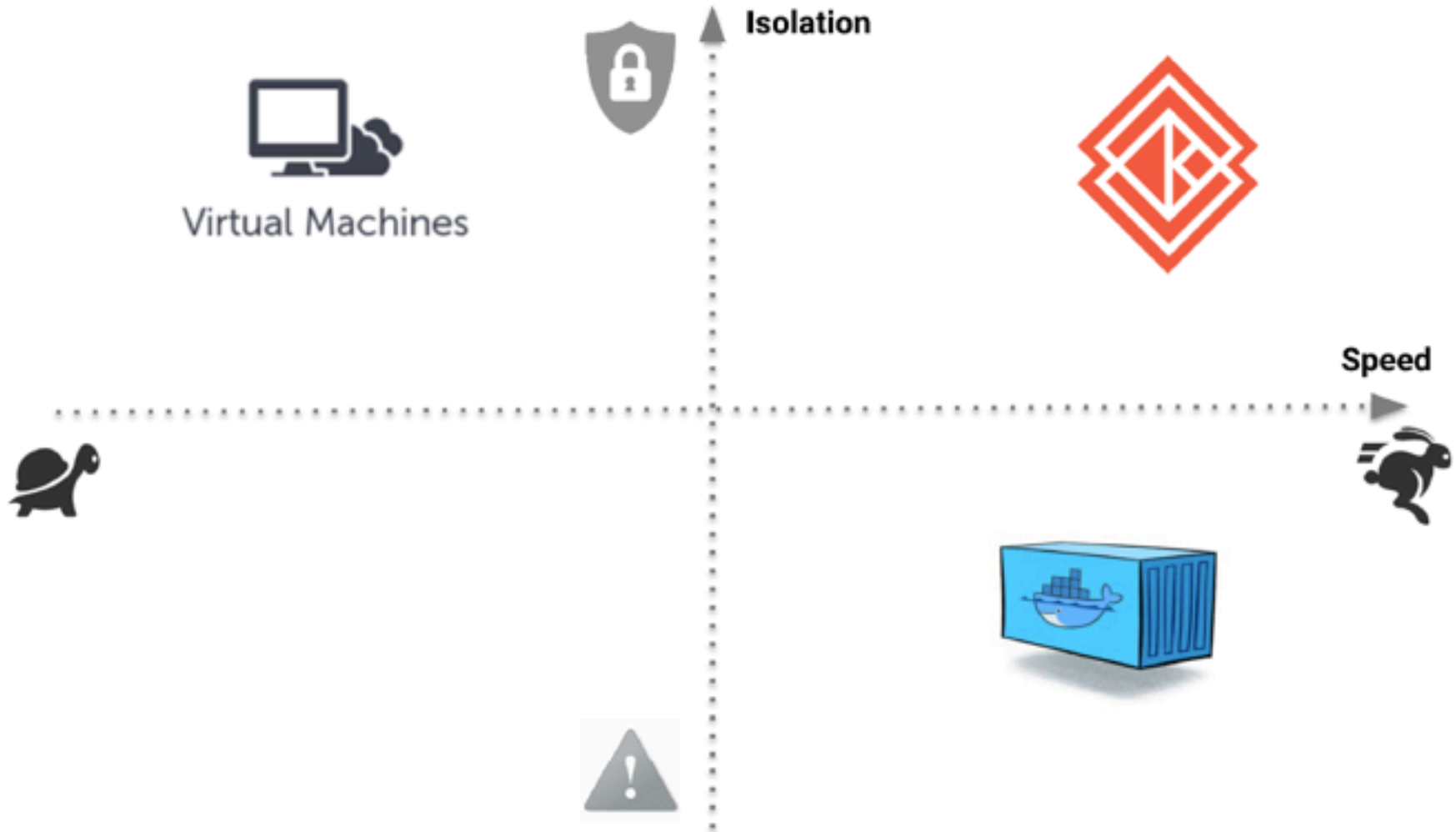
NO EMULATION

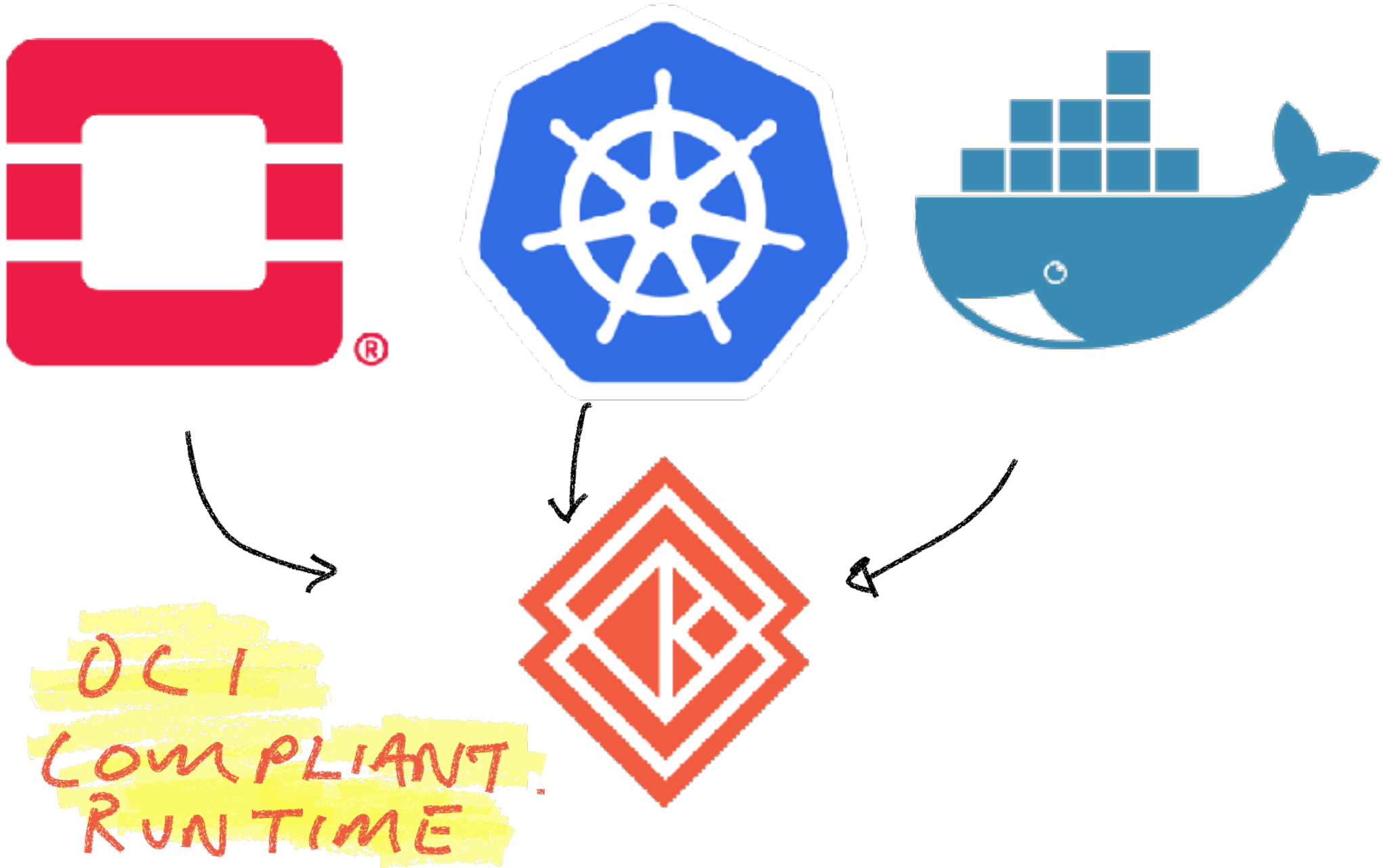
✓ New enlightened machine type

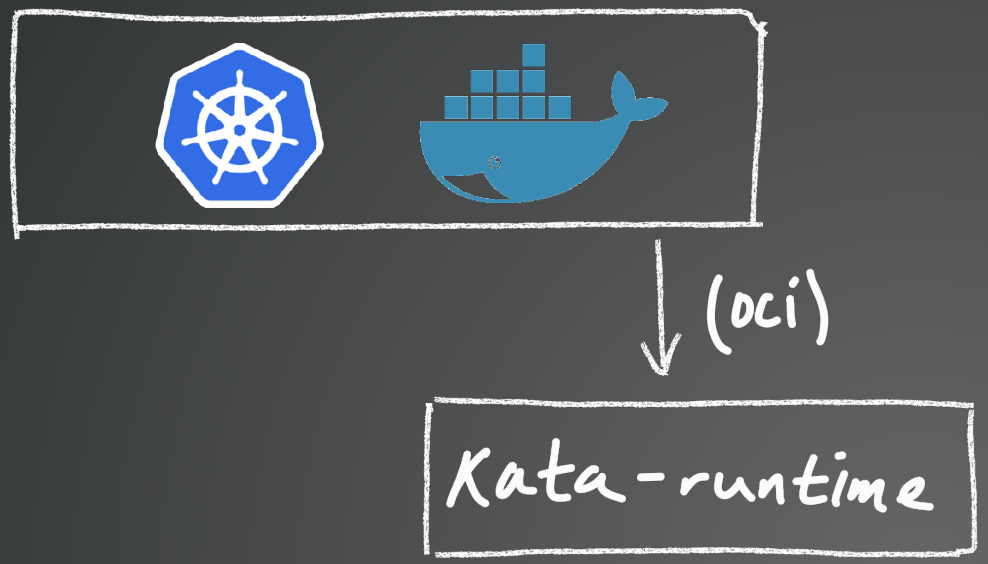
✓ Based on QEMU

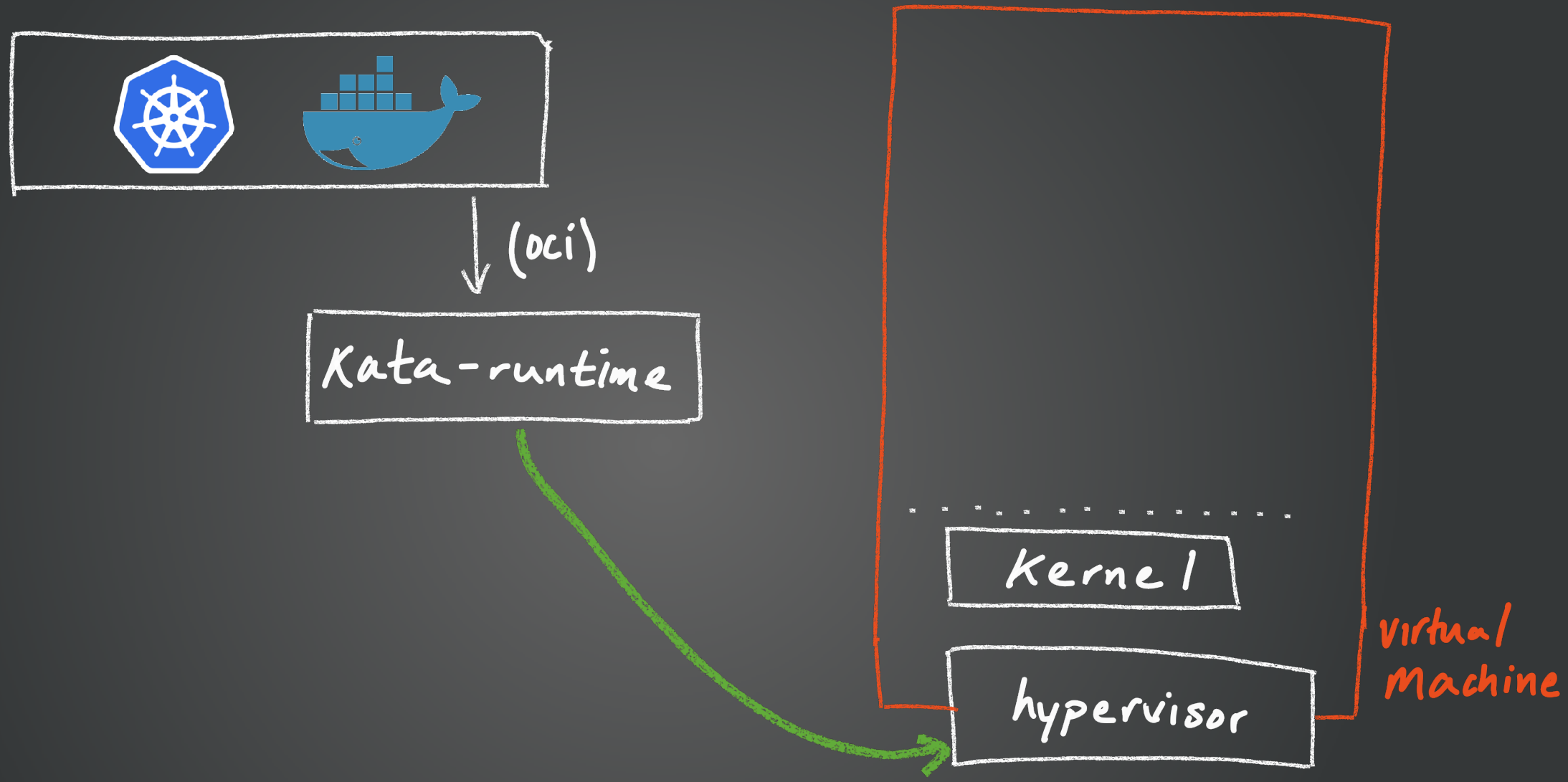
✓ See talk @ kvm-forum on Thursday

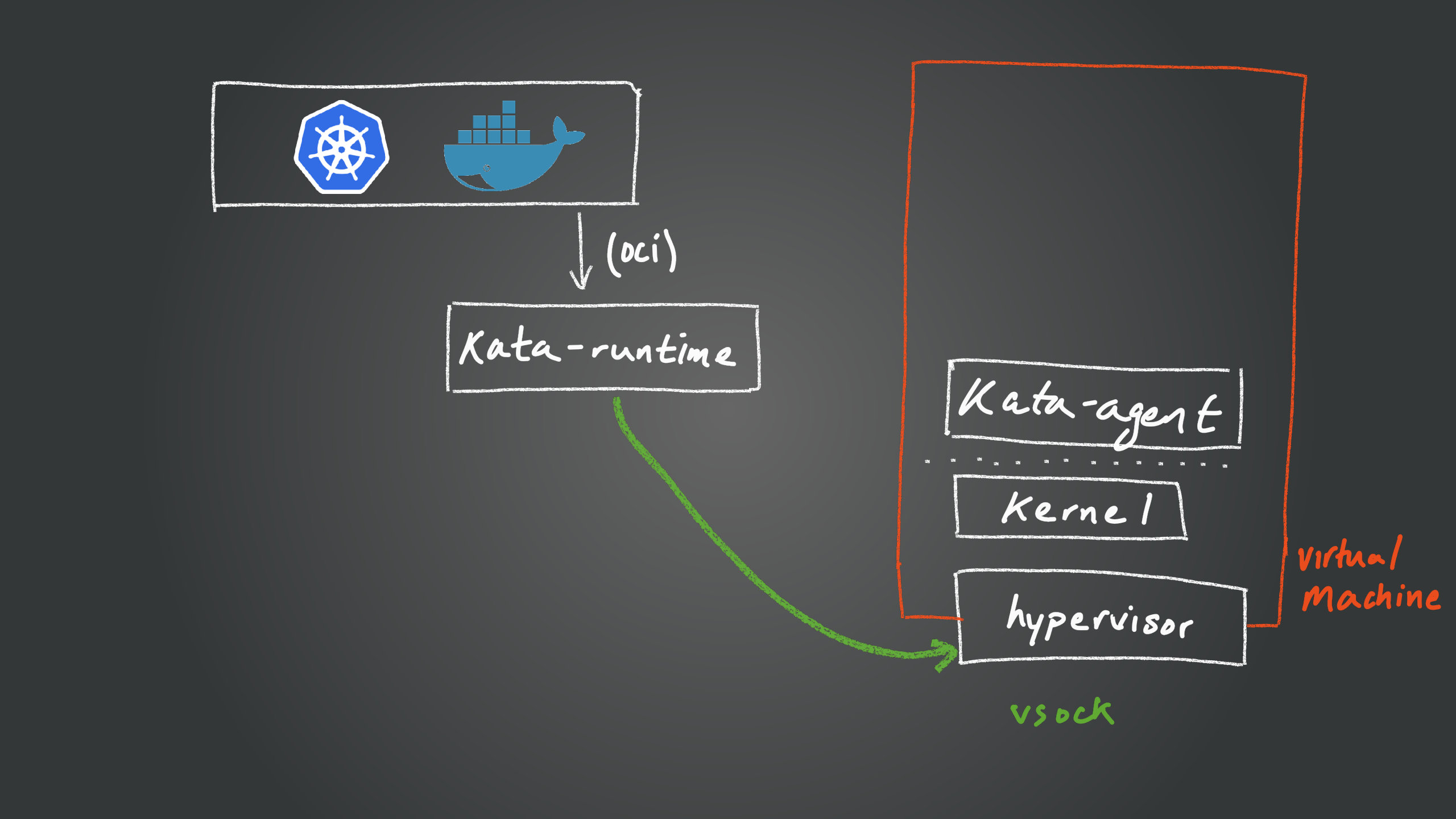


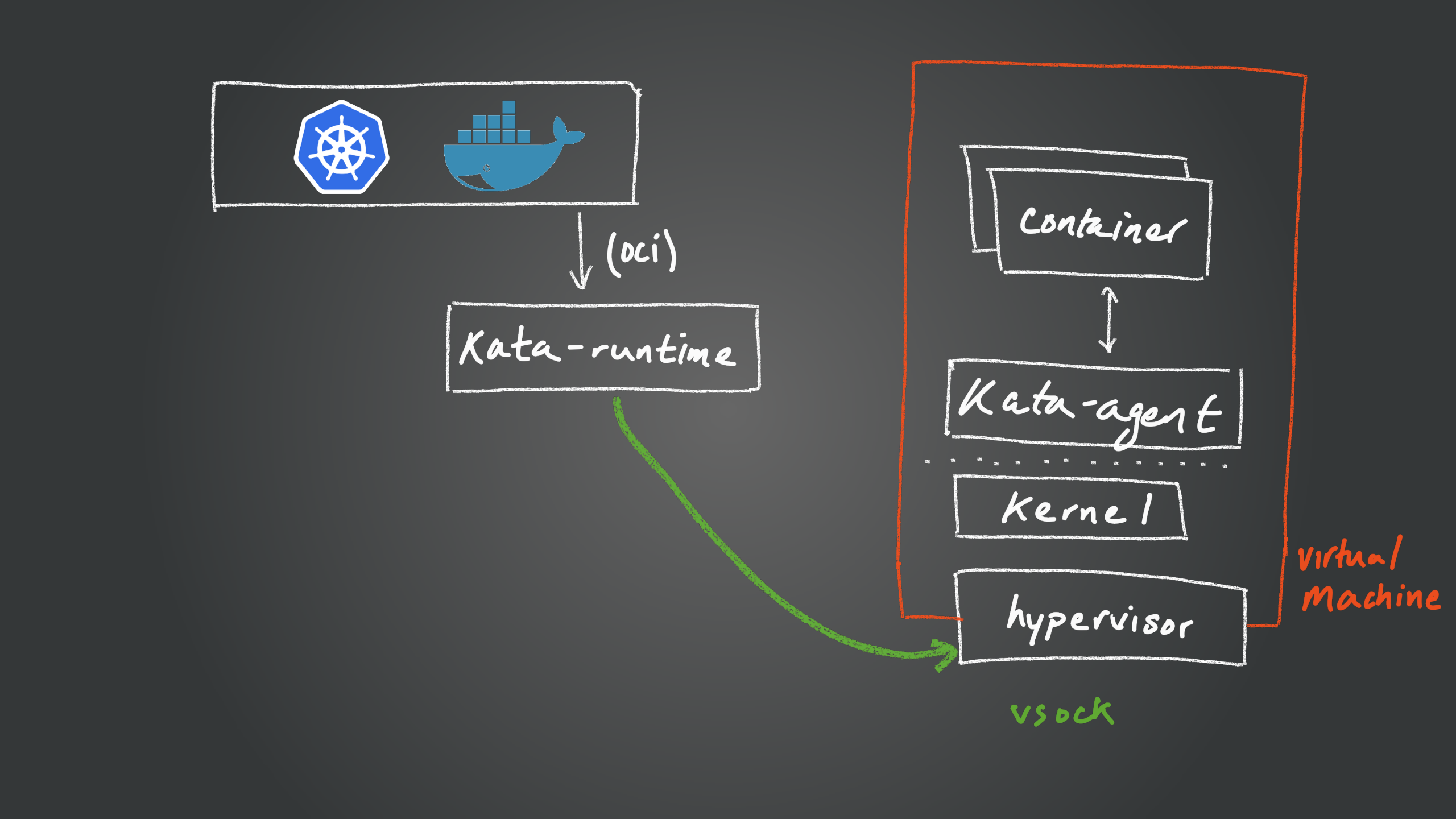

Isolation
Virtual Machines
Speed

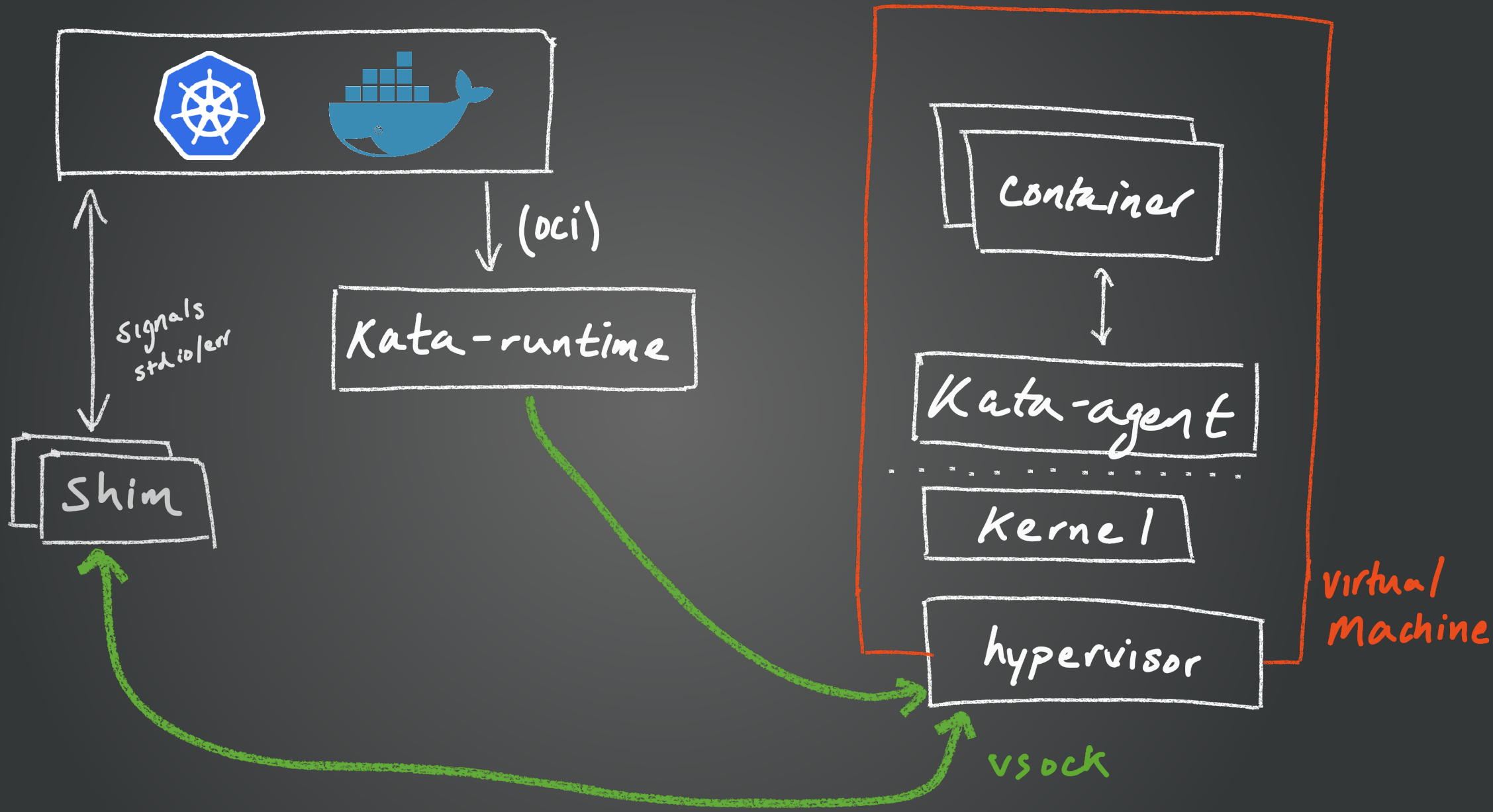

OCI
COMPLIANT
RUNTIME

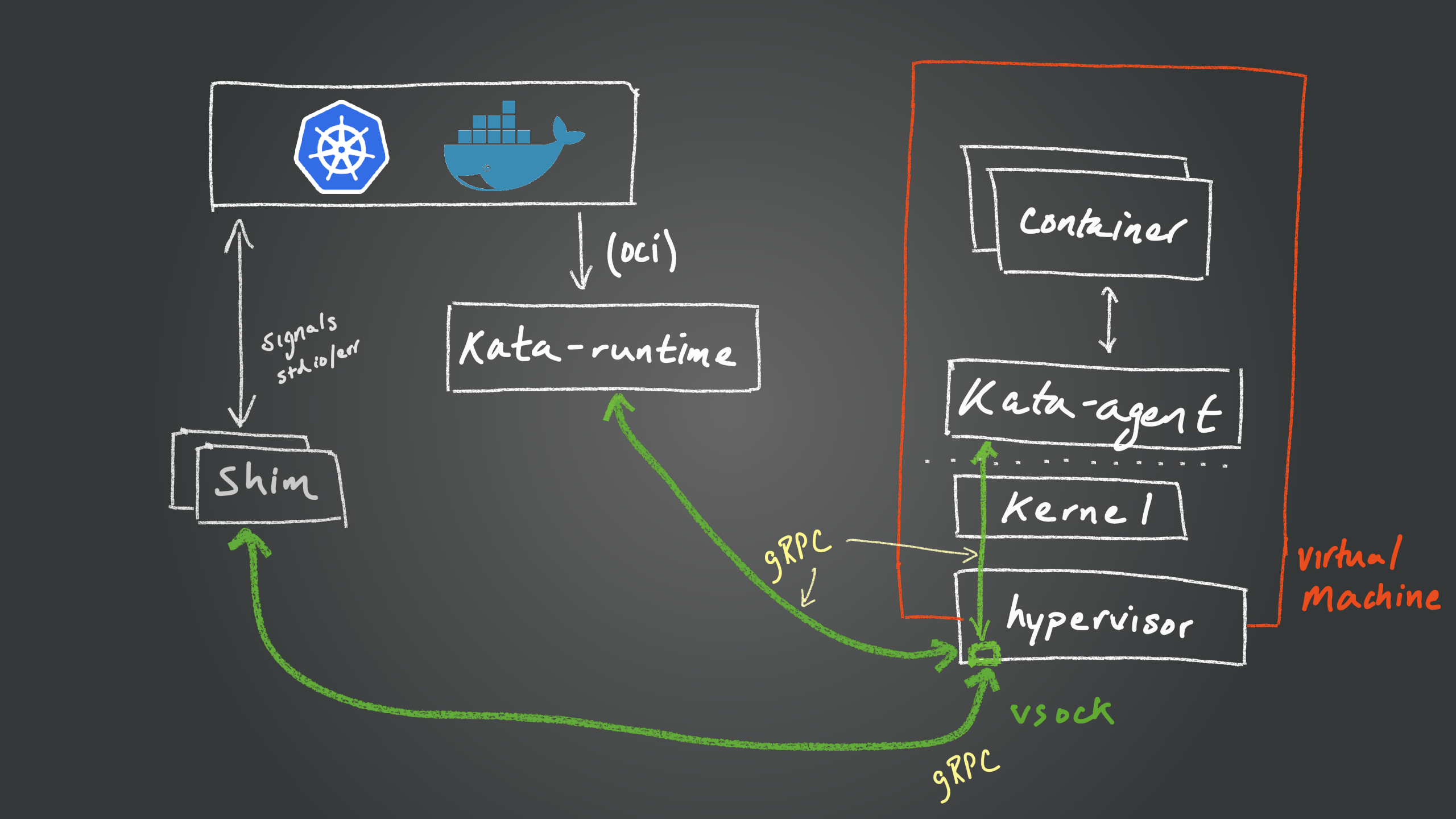

(oci)
Kata-runtime


(oci)
Kata-runtime
Kernel
hypervisor
virtual machine


(oci)
Kata-runtime
Kata-agent
Kernel
hypervisor
virtual machine
vsock


(oci)
Kata-runtime
Container
Kata-agent
Kernel
hypervisor
virtual machine
vsock

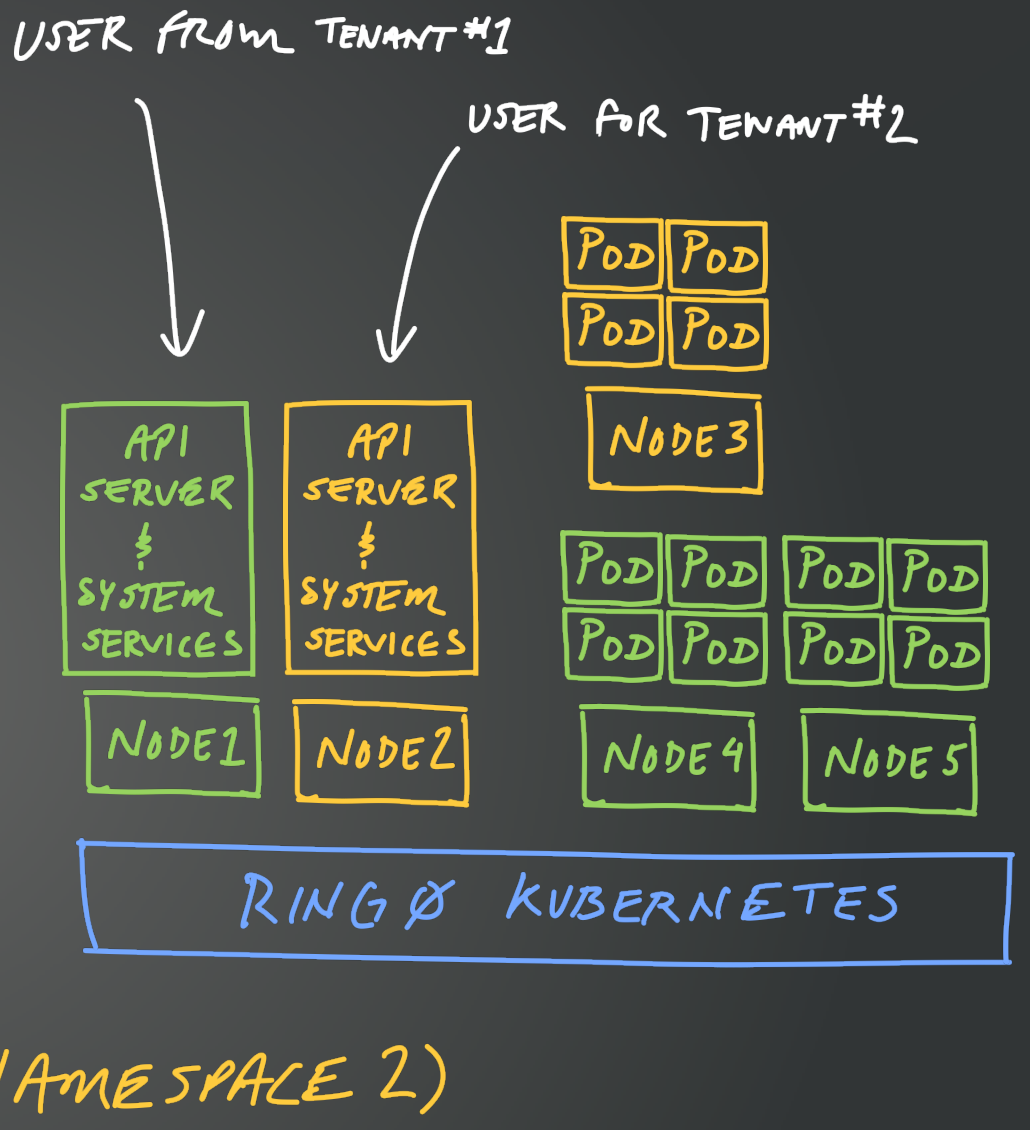

(oci)
Kata-runtime
signals
std io/err
Shim
Container
Kata-agent
Kernel
hypervisor
virtual machine
vsock


(oci)
Kata-runtime
signals
std io/err
Shim
Container
Kata-agent
Kernel
hypervisor
virtual machine
gRPC
gRPC
vsock
gRPC

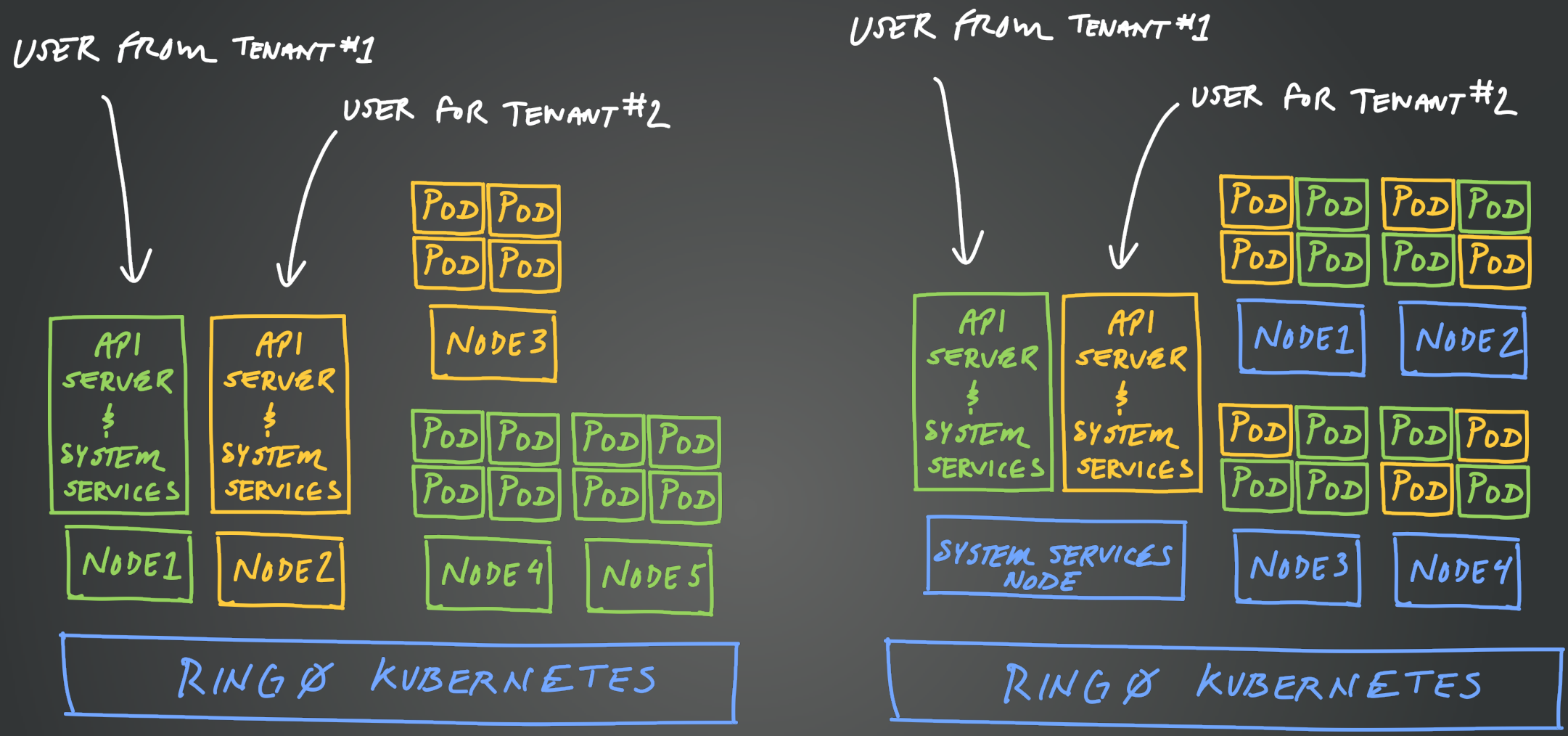
Nested Use Case

Nested Kata use case

- How many?
  - What's an appropriate pool size for tenant 1, or a particular workload?
  - Does each untrusted workload need its own VM?
- Infrastructure work:
  - Need to manage virtual machine creation, and tag each per workload/tenant.
  - May need to size virtual machines conservatively based on what a workload *could* need
  - SDN, SDS and fabric overheads with spinning up VMs

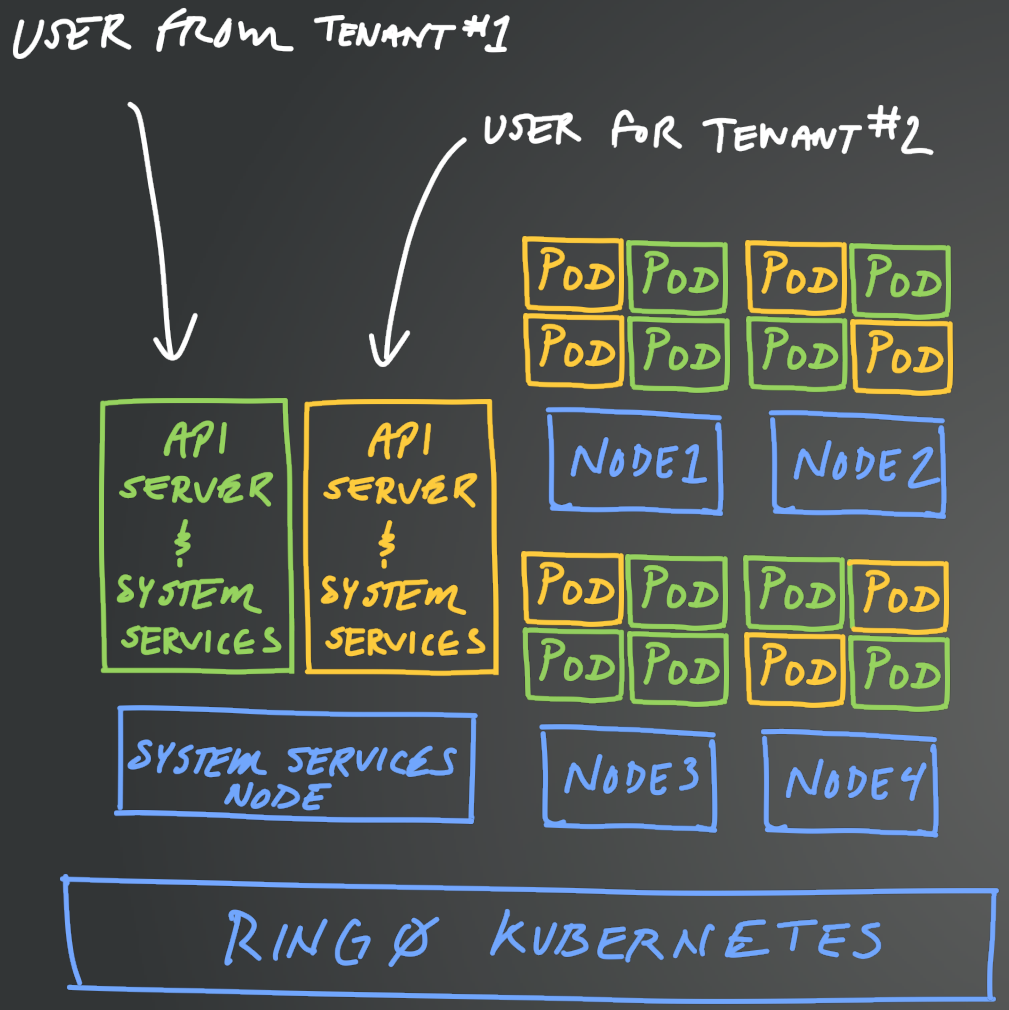


Nested Kata use case



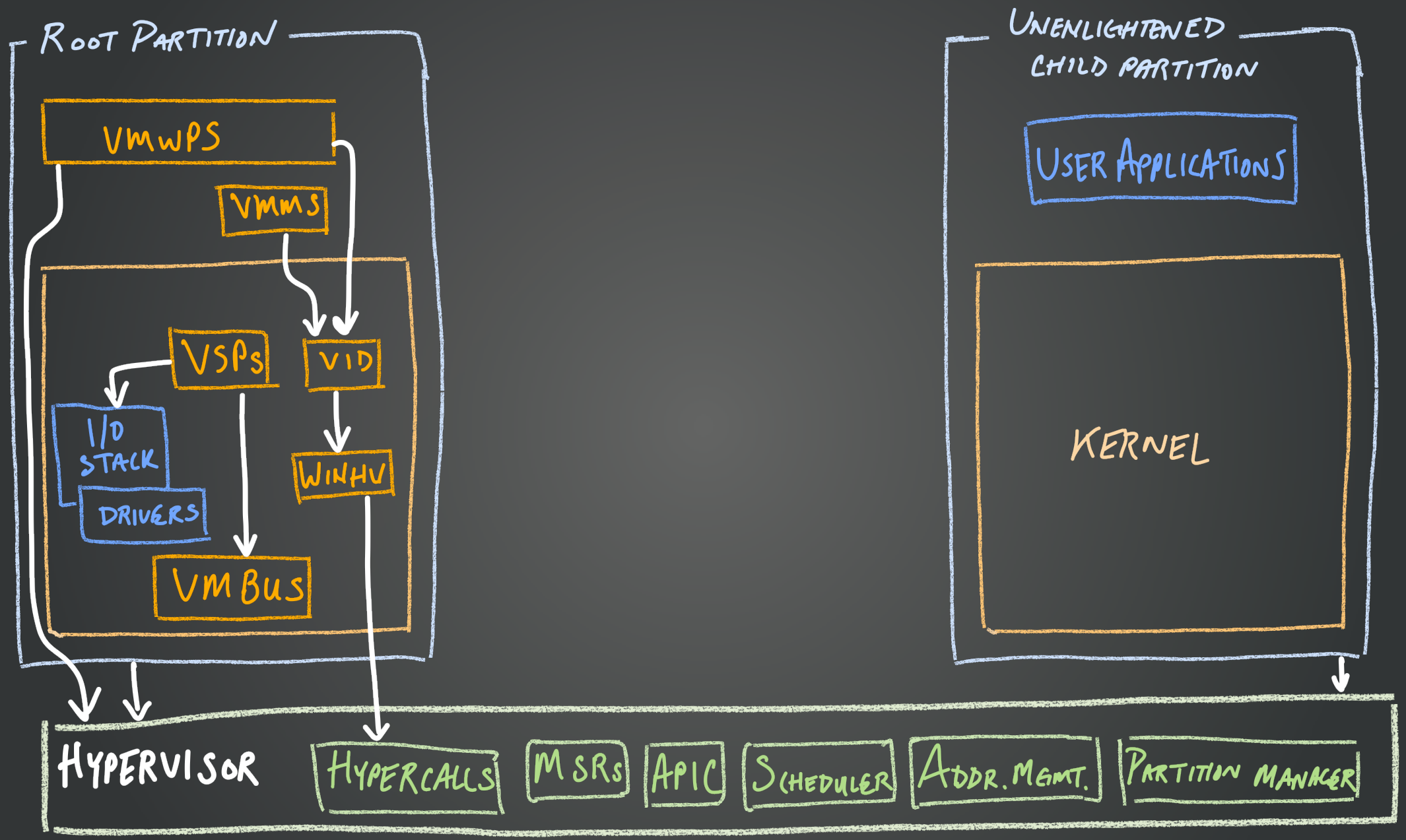
(NAMESPACE 1) (NAMESPACE 2)

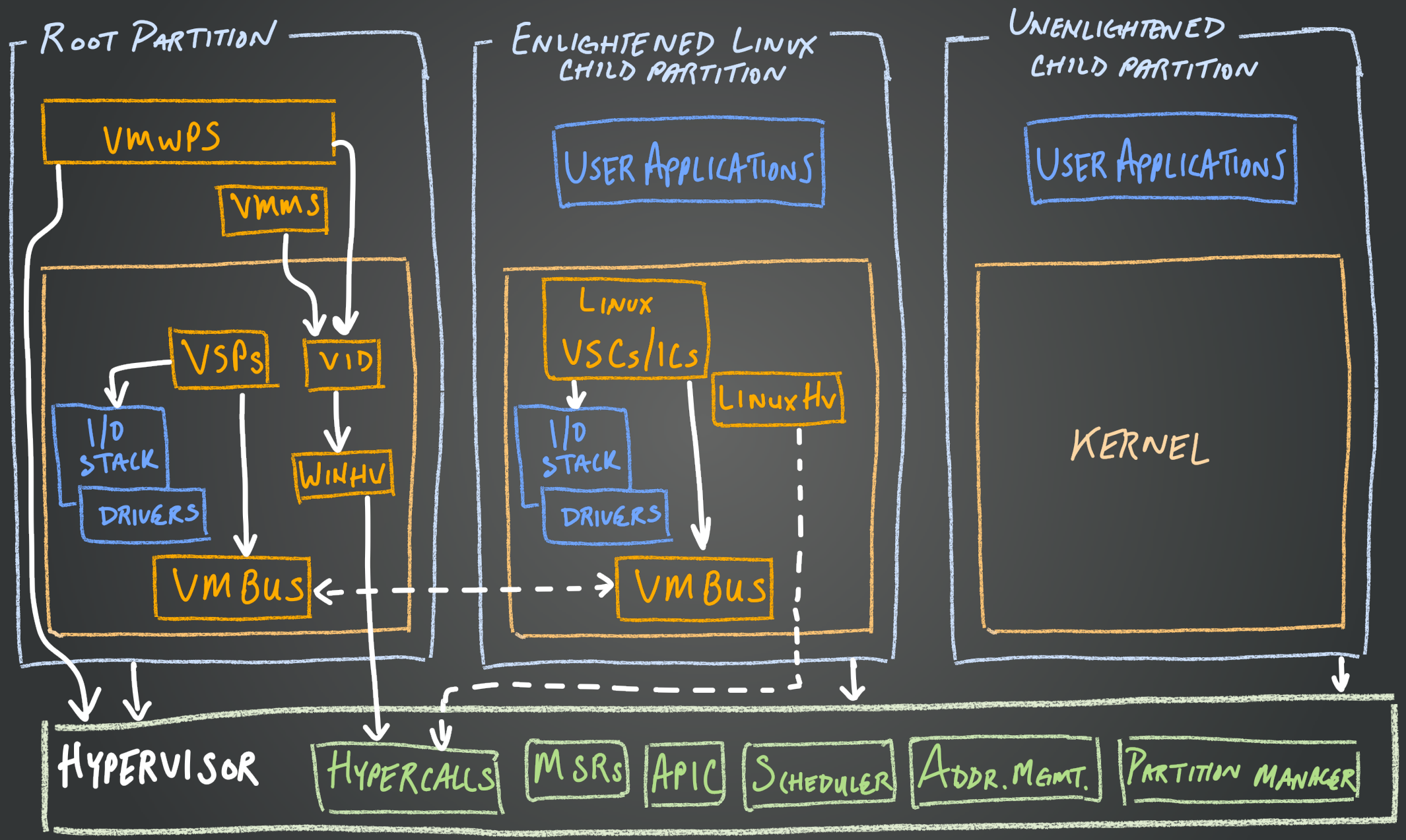
Nested Kata use case

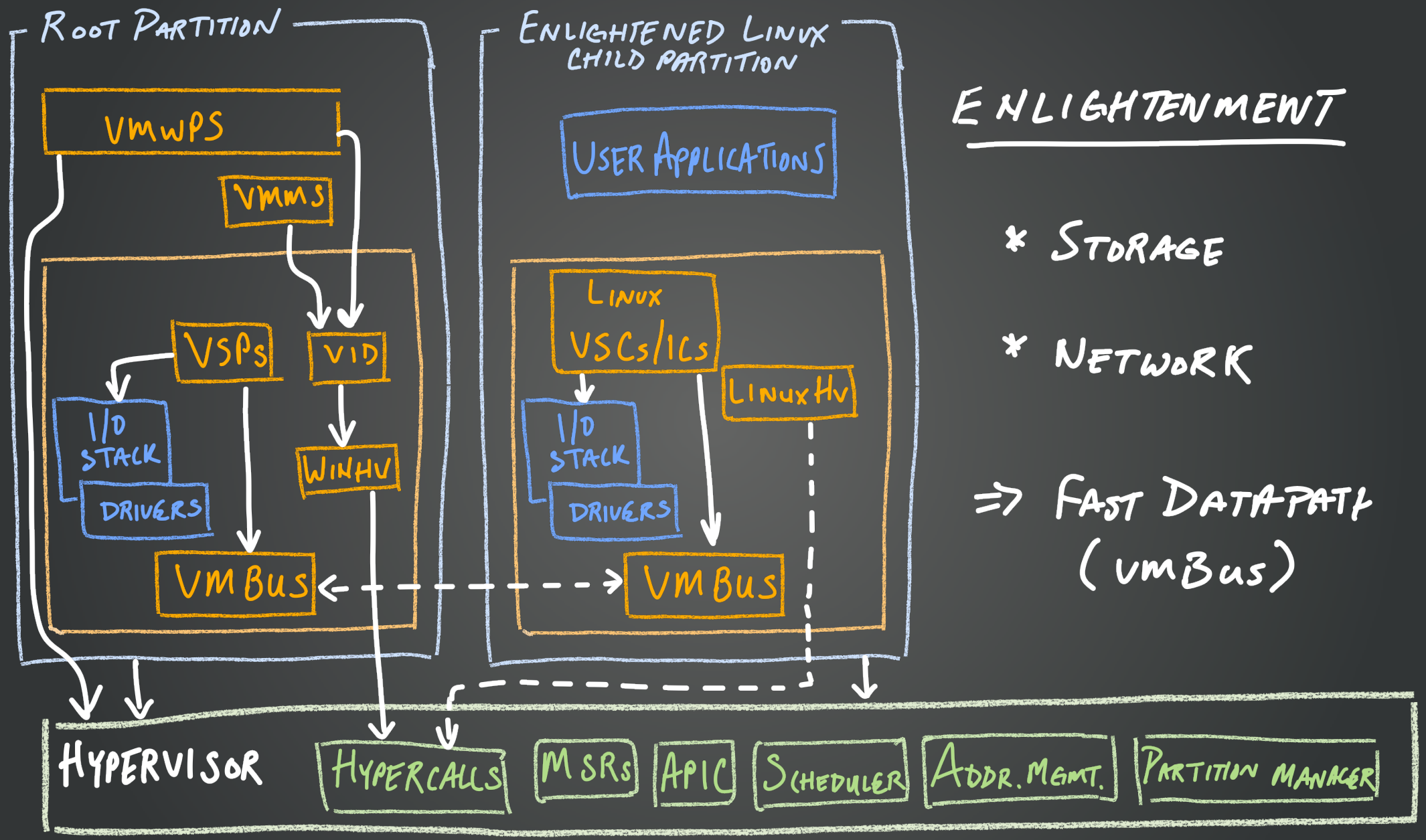


- Better utilization of resources:
  - No need to size VM based on workload needs
  - No need to size node pool based on potential tenants potential need
  - Pool is shared at a finer granularity

KVM on Hyper-V


Root Partition
VMWPS
VMMS
VSPs
VID
I/O STACK
DRIVERS
WINHV
VMBus
Unenlightened Child Partition
User Applications
KERNEL
HYPERVISOR
HYPERCALLS
MSRs
APIC
Scheduler
Addr. Mgmt.
Partition Manager

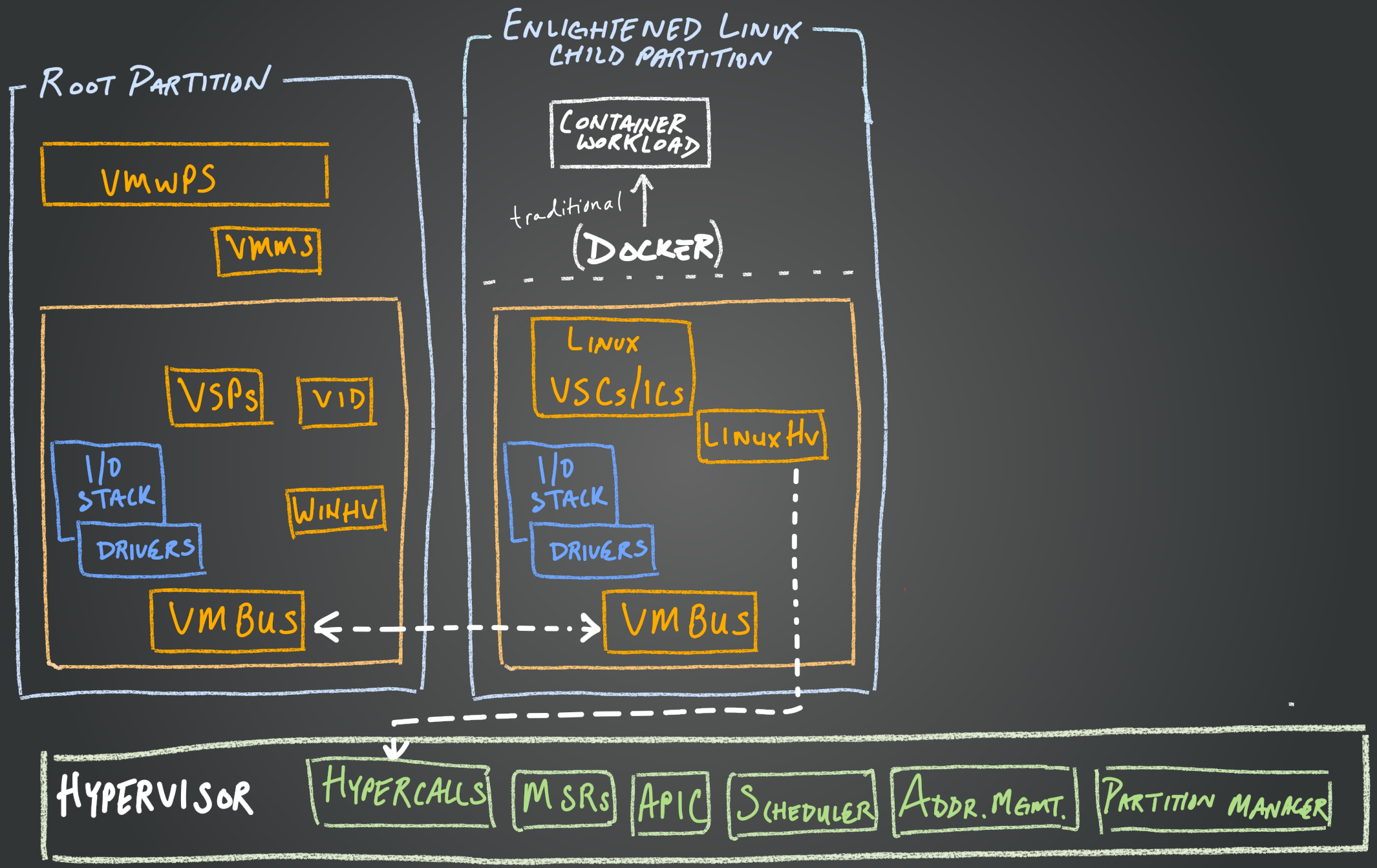

Root Partition
VMWPS
VMMS
VSPs
VID
I/O STACK
DRIVERS
WINHV
VM Bus
Enlightened Linux Child Partition
User Applications
Linux VSCs/ICs
I/O STACK
DRIVERS
LinuxHv
VM Bus
Unenlightened Child Partition
User Applications
KERNEL
HYPERVISOR
HyperCalls
MSRs
APIC
Scheduler
Addr. Mgmt.
Partition Manager

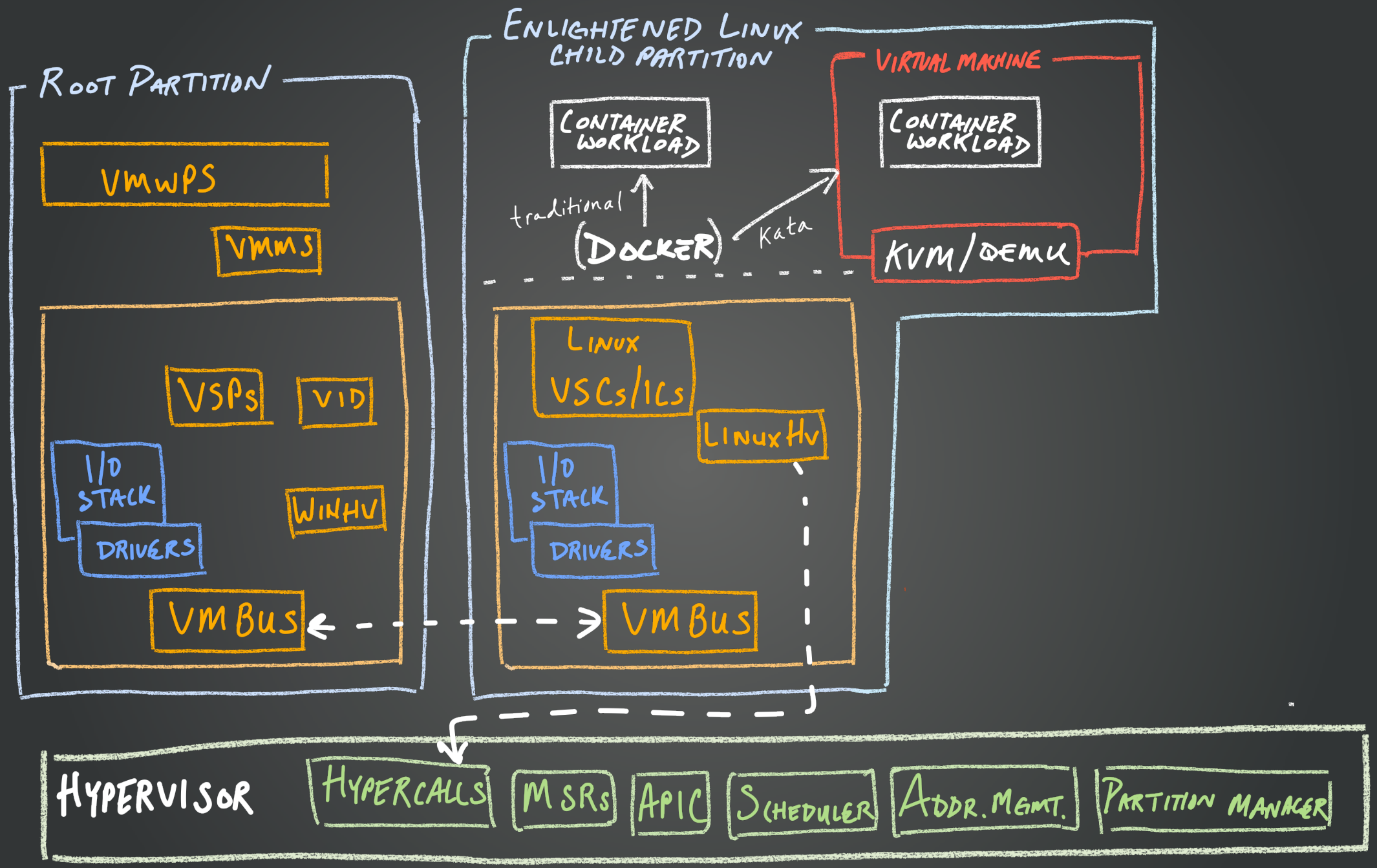


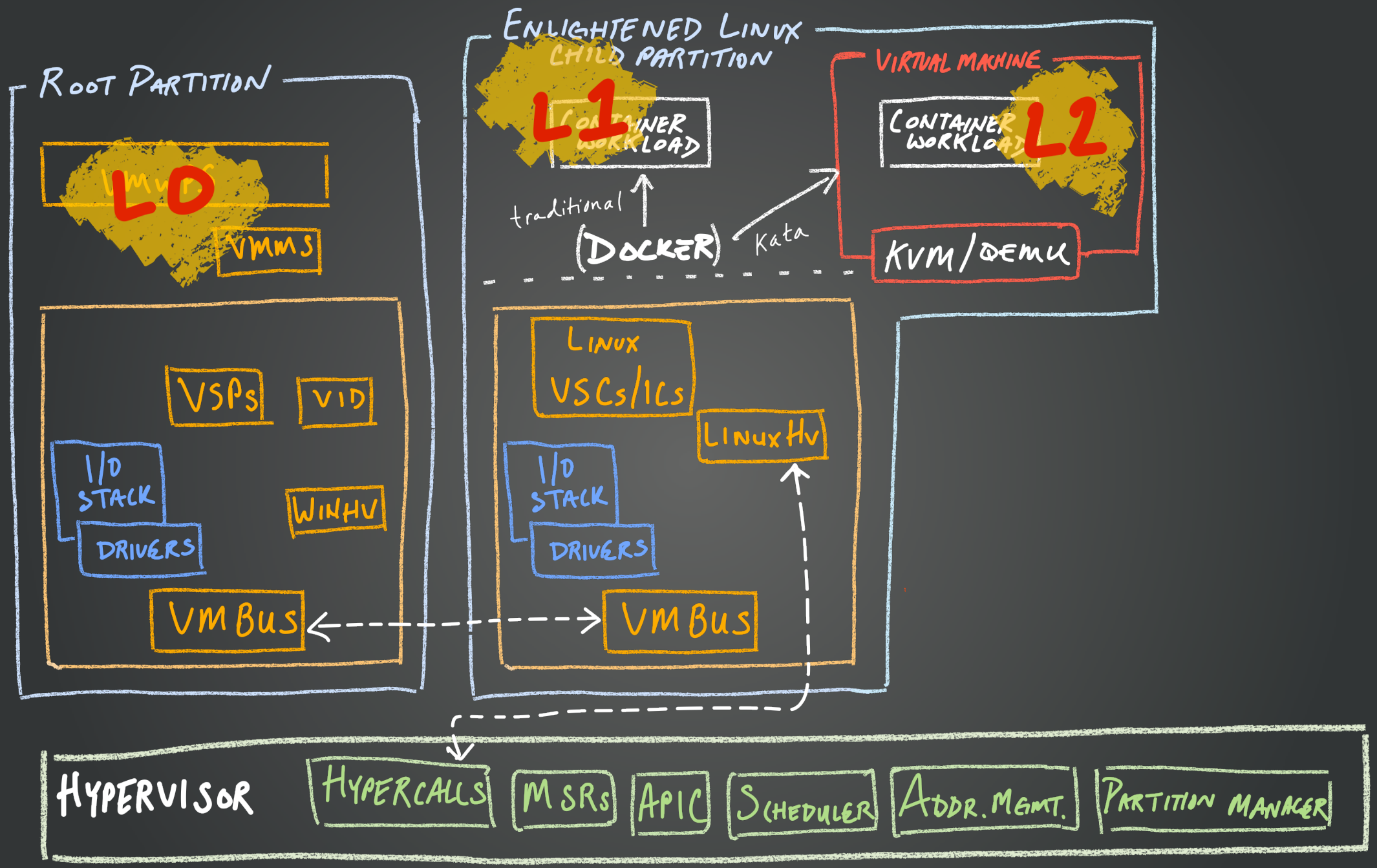
Enlightenment

* Storage
* Network

⇒ Fast Datapath (vmBus)

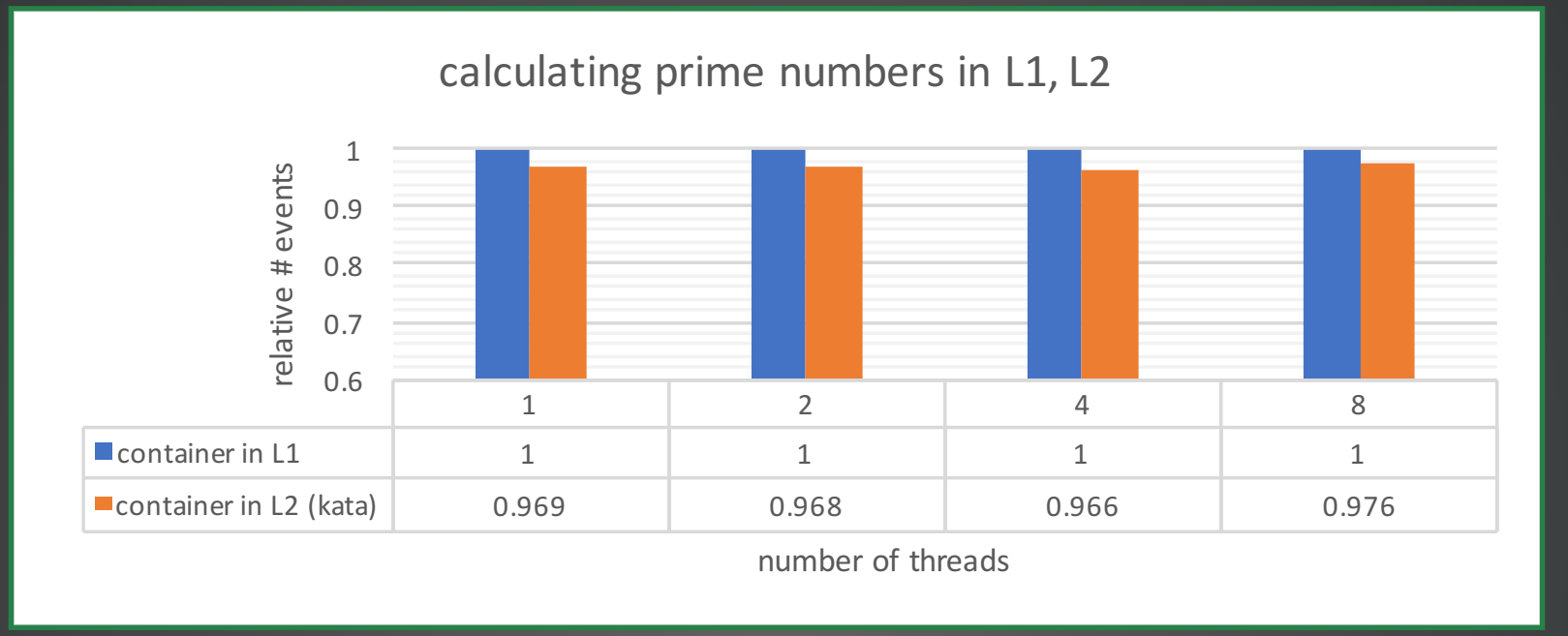

Root Partition
VMWPS
VMMS
VSPs
VID
I/O STACK
DRIVERS
WINHV
VMBus
Enlightened Linux Child Partition
Container Workload
traditional
(Docker)
Linux VSCs/ICs
LinuxHv
I/O STACK
DRIVERS
VMBus
Hypervisor
Hypercalls
MSRs
APIC
Scheduler
Addr. Mgmt.
Partition Manager


Root Partition
VMWPS
VMMS
VSPs
VID
I/O Stack
Drivers
WinHv
VMBus
Enlightened Linux Child Partition
Container Workload
traditional
(Docker)
Kata
Virtual Machine
Container Workload
KVM/QEMU
Linux VSCs/ICs
LinuxHv
I/O Stack
Drivers
VMBus
Hypervisor
Hypercalls
MSRs
APIC
Scheduler
Addr. Mgmt.
Partition Manager


ROOT PARTITION
L0
VMMS
VSPs
VID
I/O STACK
DRIVERS
WINHV
VMBUS
ENLIGHTENED LINUX CHILD PARTITION
L1
CONTAINER WORKLOAD
traditional
DOCKER
Kata
VIRTUAL MACHINE
CONTAINER WORKLOAD
L2
KVM/QEMU
LINUX VSCs/ICs
LINUX HV
I/O STACK
DRIVERS
VMBUS
HYPERVISOR
HYPERCALLS
MSRs
APIC
SCHEDULER
ADDR. MGMT.
PARTITION MANAGER

Nested Kata

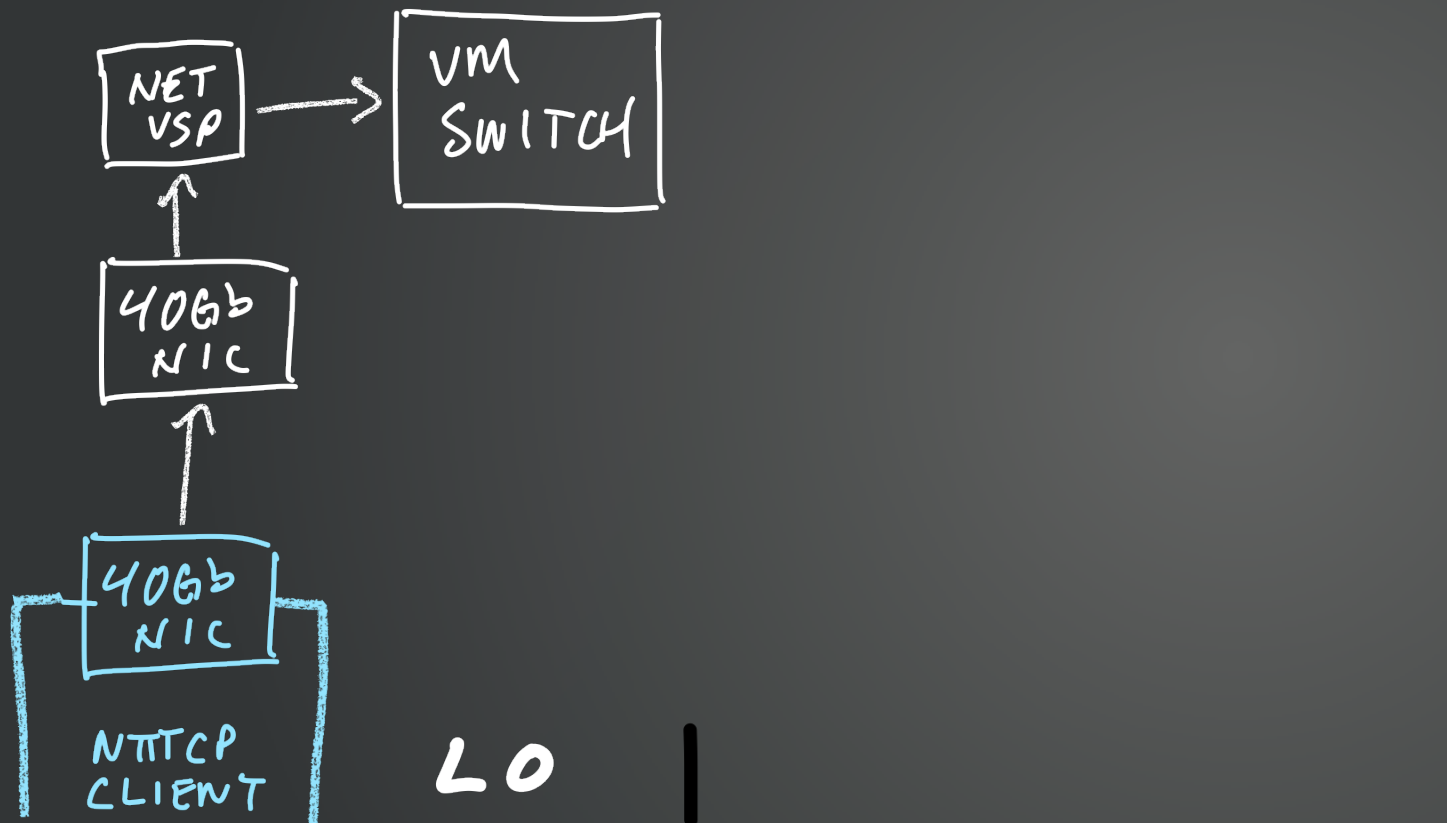
Nested Kata: CPU Measurements



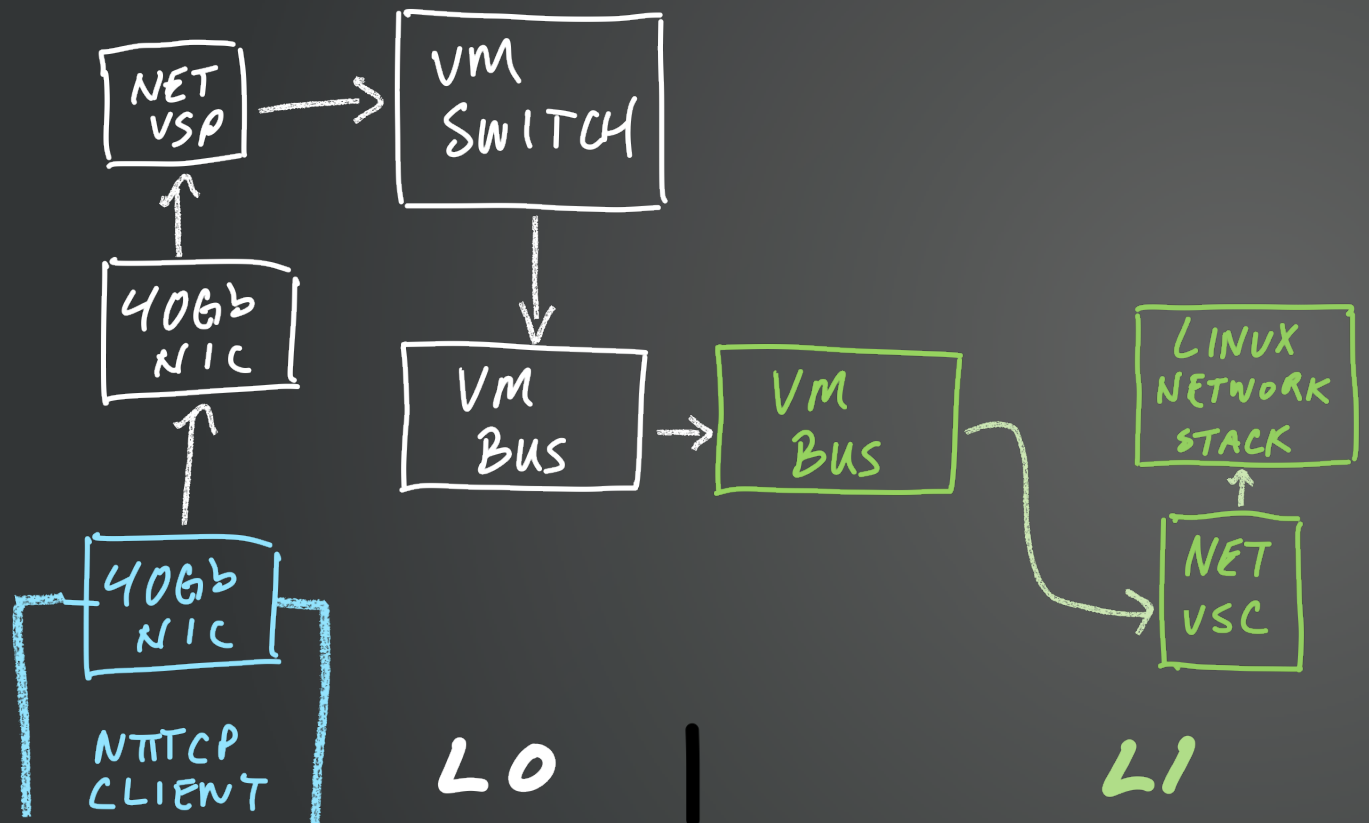
	1	2	4	8
container in L1	1	1	1	1
container in L2 (kata)	0.969	0.968	0.966	0.976

Approximately 3% degradation seen when running with varying Number of threads on prime number calculation workload

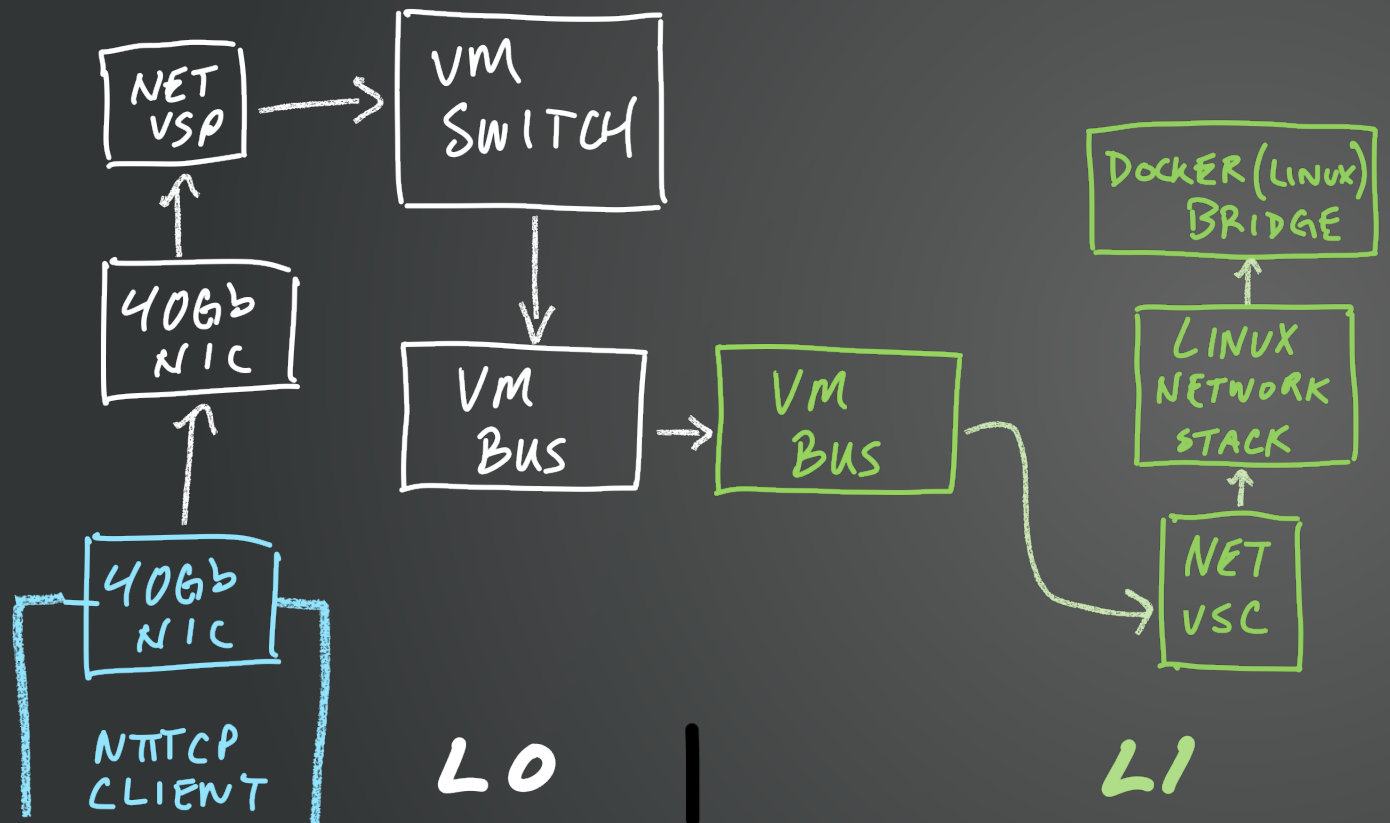
Nested Kata: Network I/O - setup



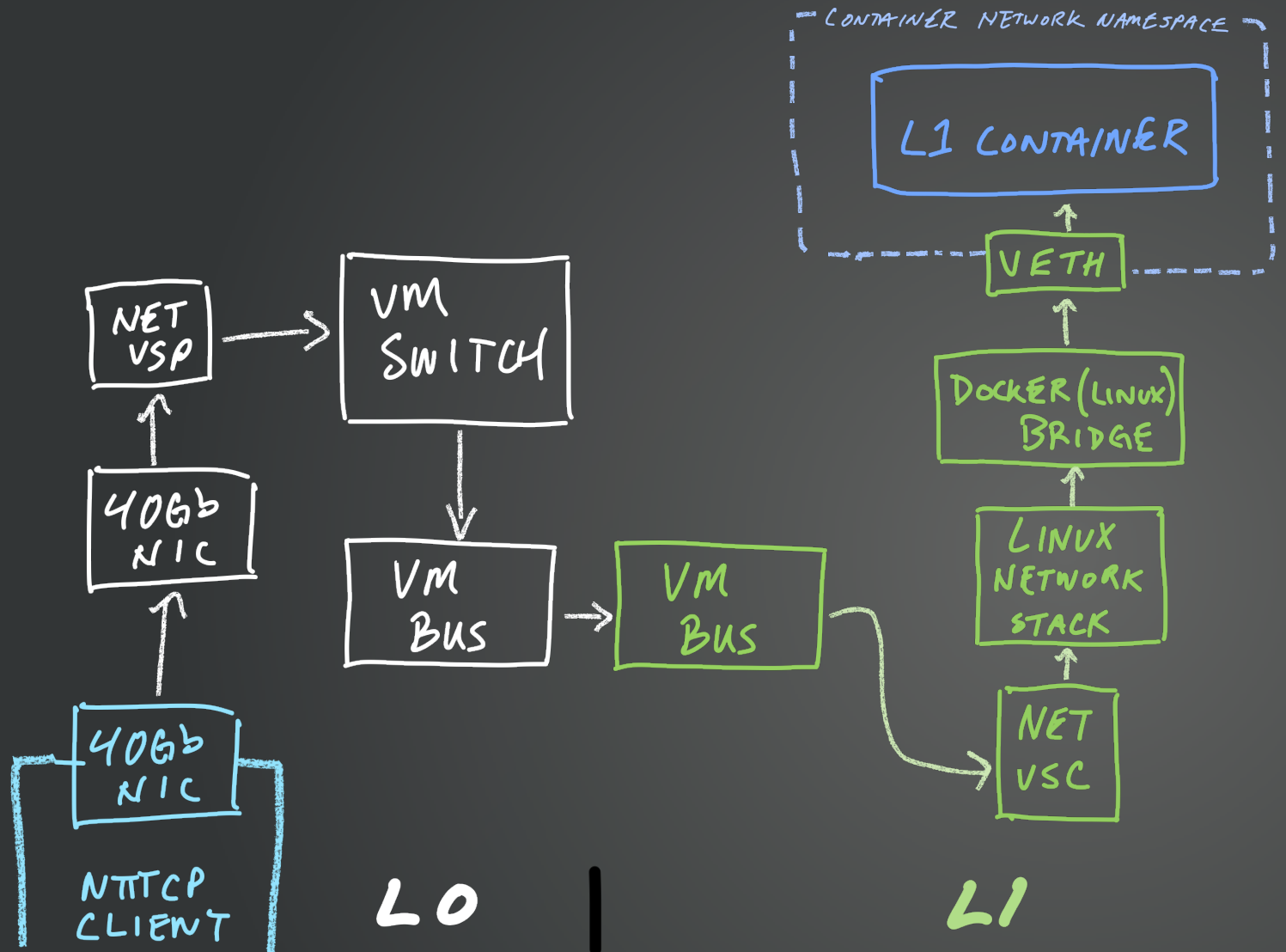
Nested Kata: Network I/O - setup



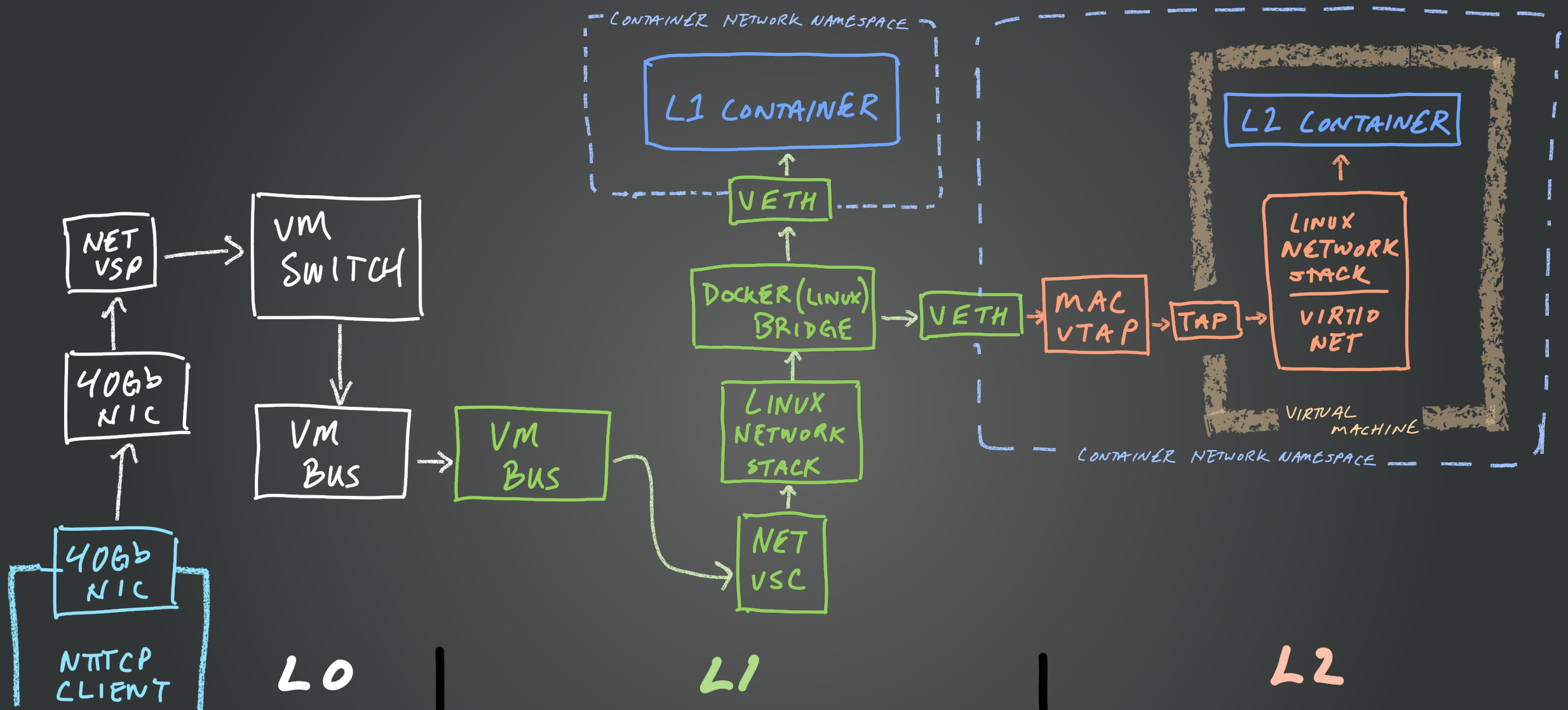
Nested Kata: Network I/O - setup



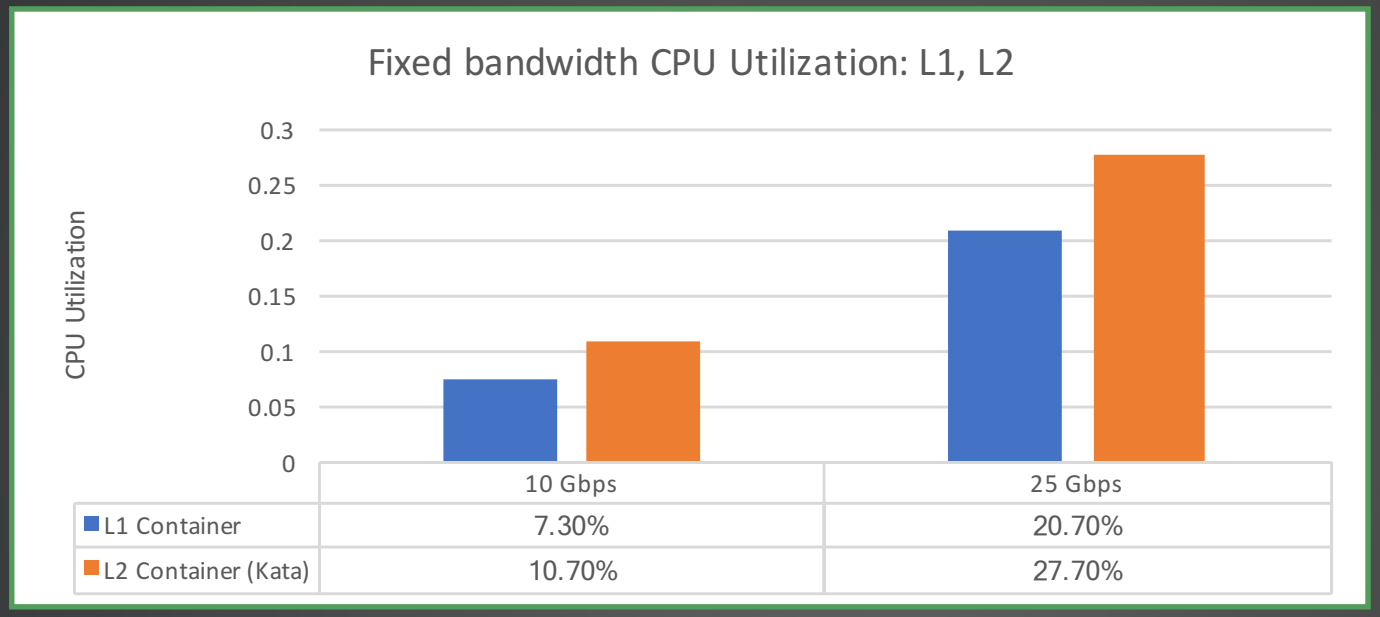
Nested Kata: Network I/O - setup



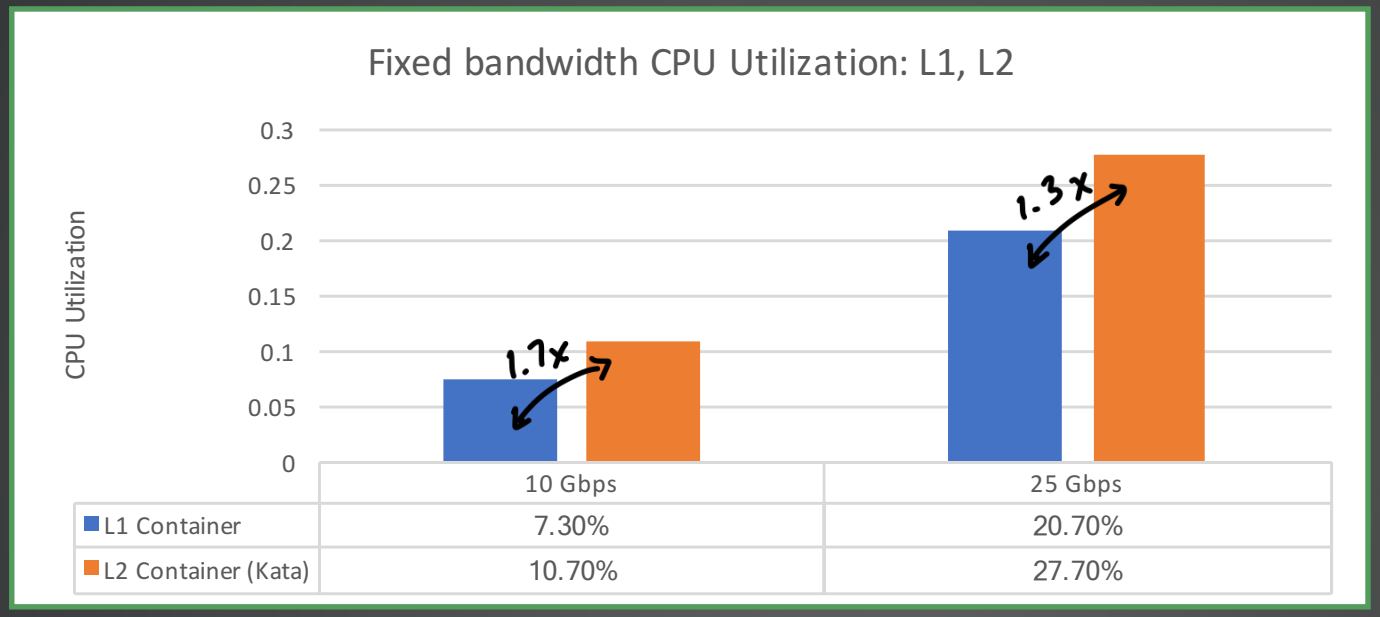
Nested Kata: Network I/O - setup



Nested Kata Network I/O



Nested Kata Network I/O

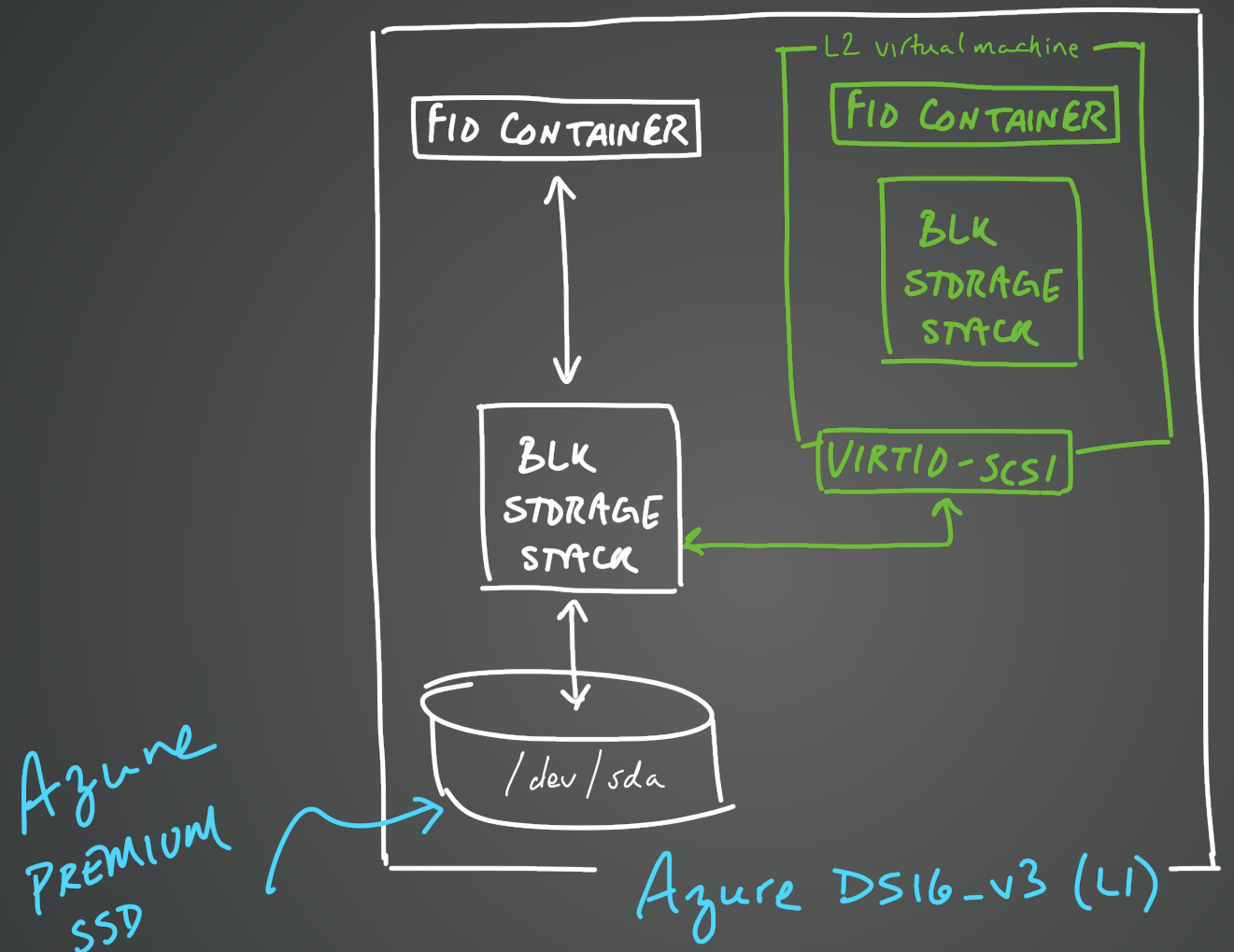


CORES USED

	10Gbps	25Gbps
L1:	3.2	9.1
L2:	4.7	12.2

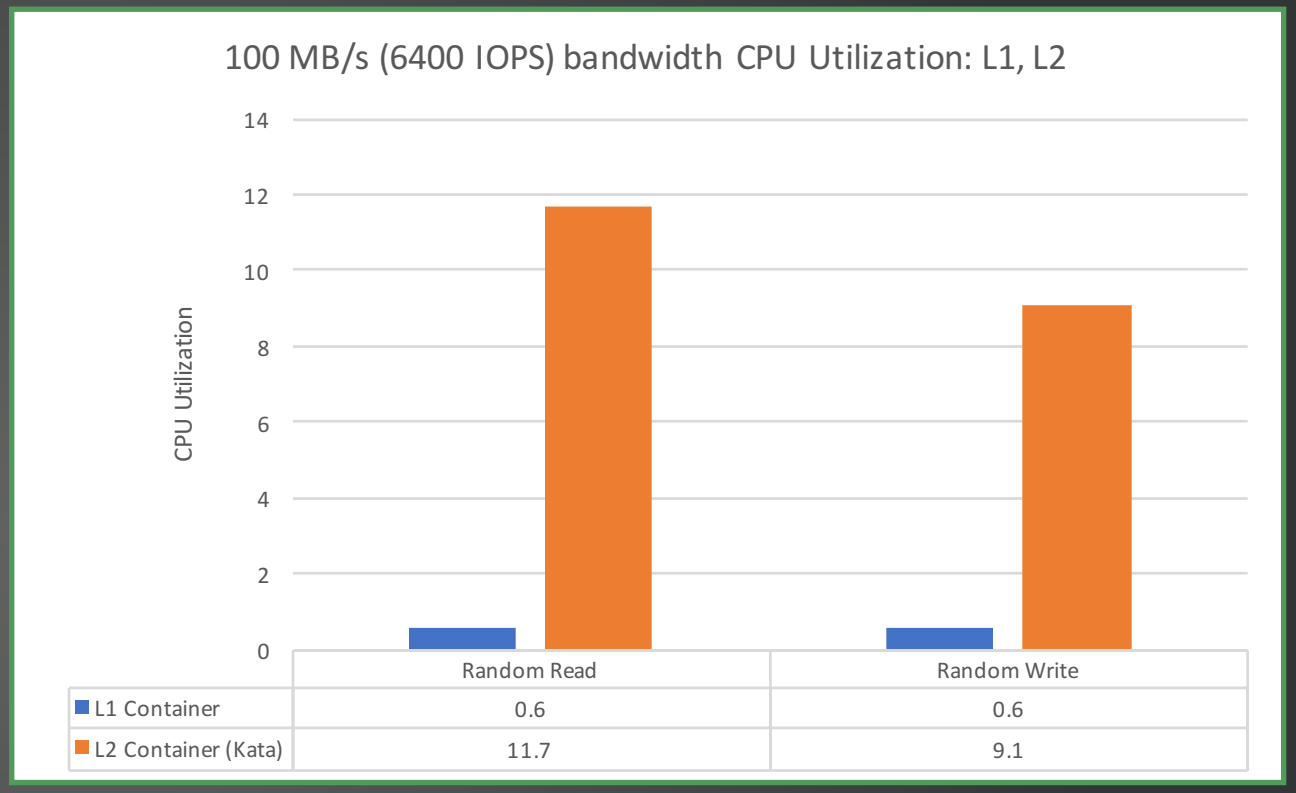
* measured utilization in L0

Nested Kata: Storage I/O - setup



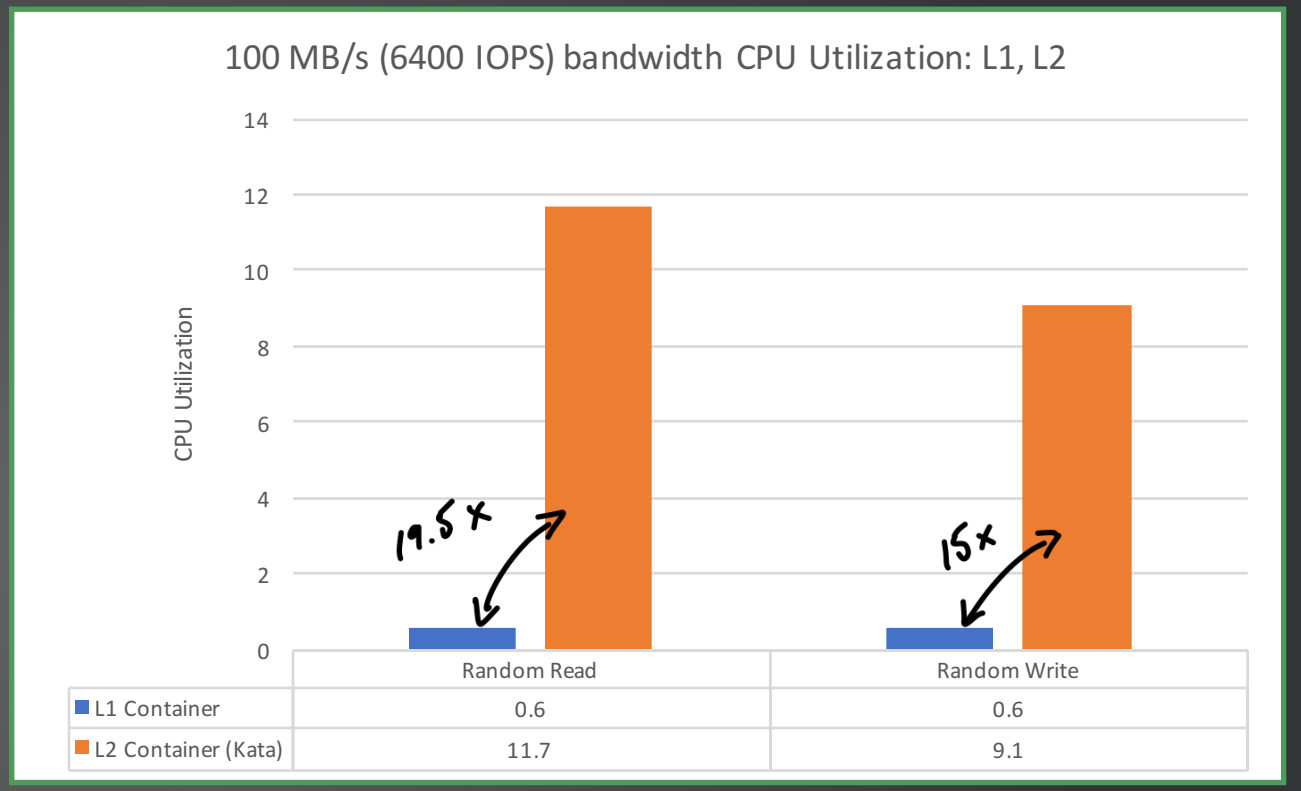
Nested Kata Storage I/O

- Nested is relatively expensive
- High amount of iowait observed in L1 during L2 random read testing



Nested Kata Storage I/O

- Nested is relatively expensive
- High amount of iowait observed in L1 during L2 random read testing



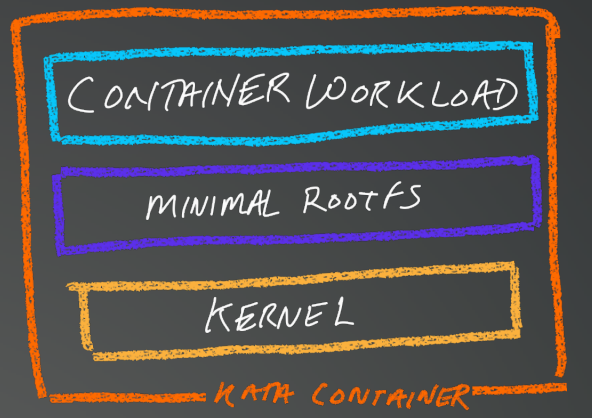
RR RW

L2: 1.8 cores, 1.5 cores

*measured utilization in L1

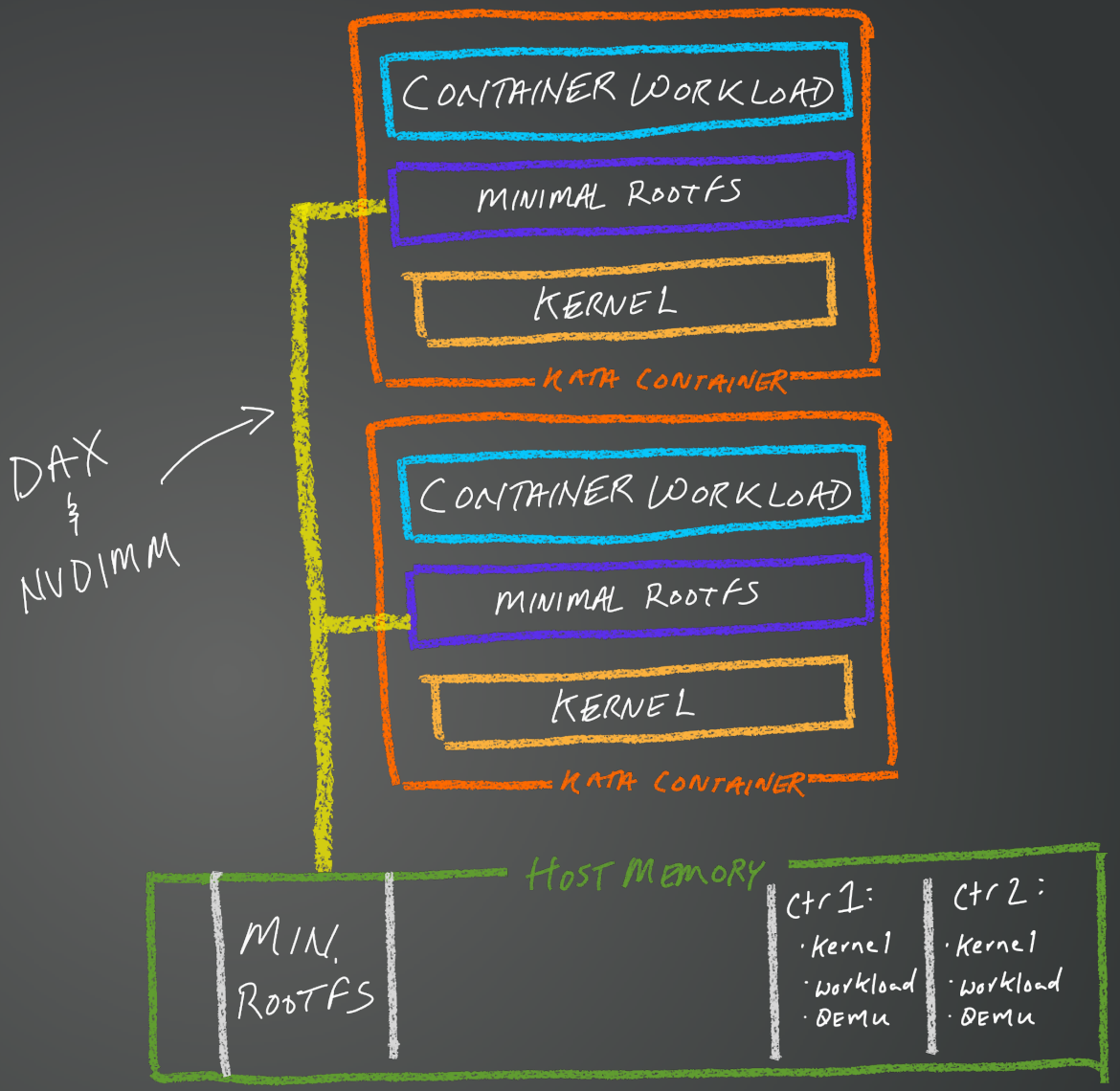
Reducing Kata's footprint

- Minimal kernel
- Minimal rootfs
- Minimally configured QEMU



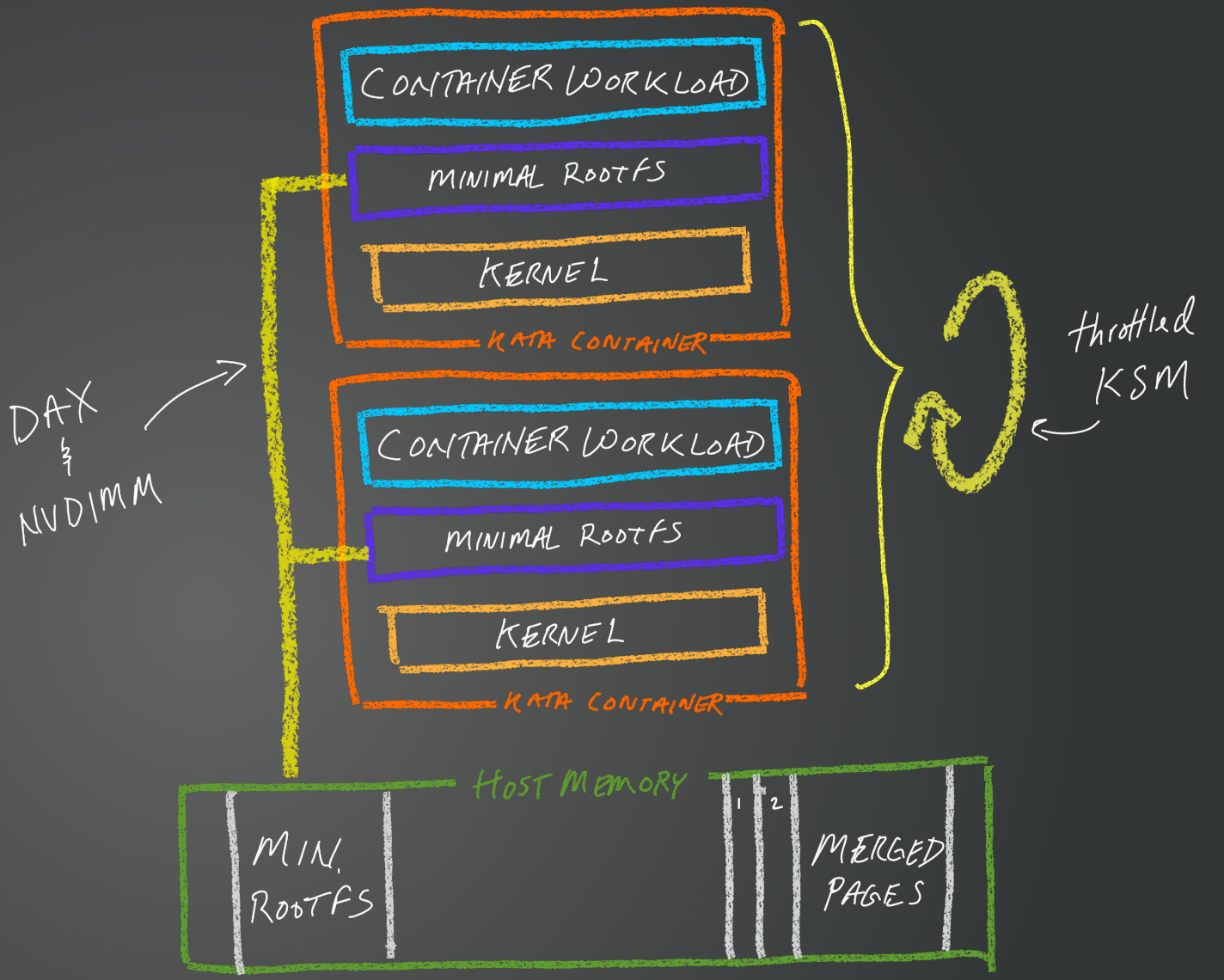
Reducing Kata's footprint

- Minimal kernel
- Minimal rootfs
- Minimally configured QEMU
- DAX/NVDIMM



Reducing Kata's footprint

- Minimal kernel
- Minimal rootfs
- Minimally configured QEMU
- DAX/NVDIMM
- De-duplicating memory



Nested Kata container density

Dockerhub workload	Memory footprint		Containers/GB	
	Kata	**runc**	**Kata**	**runc**
busybox (small)	93.2 MB	682 KB	11	1535.7
mysql (medium)	135.5 MB	160.8 MB	7.6	6.5
elasticsearch (large)	2.5 GB	2.2 GB	0.4	0.5

“it depends”

Summary, next steps

Next Steps

- Nesting:
  - Continued improvements for KVM on Hyper-V
  - Optimizations for L2:
    - Investigate more efficient L2 storage options
    - General efficiency improvements to minimize nested cost
- Kata:
  - Improvements on density as well as security
  - Released support for NEMU

Where can you get Kata?

- Dockerhub:
  - katadocker/kata-deploy
- Packages:
  - Clear linux, Snap
  - Built for CentOS, Fedora, SLES, RHEL, Ubuntu
- Running on public Cloud:
  - ACS-engine support in Azure
  - Anywhere bare-metal or nested virtualization is supported, including AWS, Azure, GCP, packet.net, Vexxhost

THE LINUX FOUNDATION

OPEN SOURCE SUMMIT

EUROPE

THE LINUX FOUNDATION