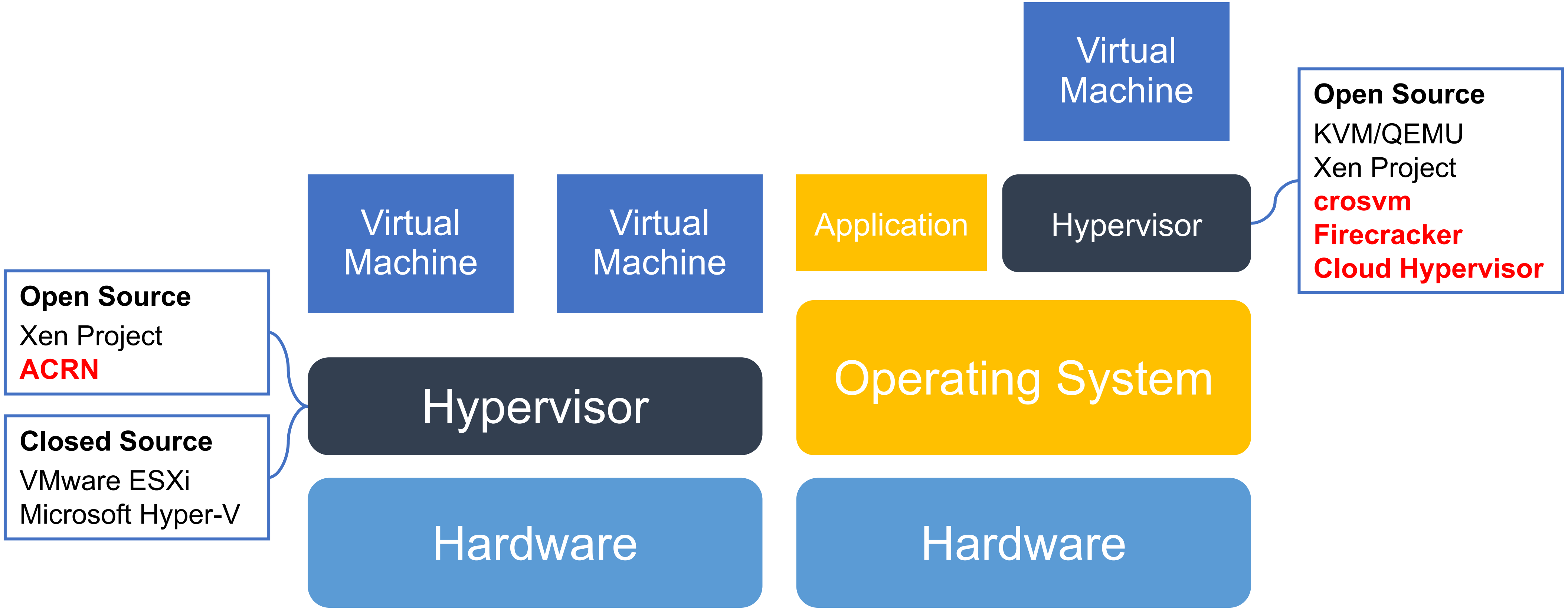
Cloud Hypervisor or Cloud Native Hypervisor

Agenda

- A Bunch of New Hypervisors and rust-vmm
- Cloud Hypervisor with Cloud Native
- Feature enabling in CLH: PMEM and vHost as example
- Community & Roadmap
- Cloud Native Hypervisor

A Bunch of New Hypervisors and rust-vmm

Hypervisors and Virtual Machines



CrosVM

- Android application sandboxing
- Rust implementation
- Strong focus on security
- Little emulation

CrosVM

April 2017

Firecracker

- AWS Lambda functions
- Rust implementation
- Strong focus on security
- **Very minimal emulation**

CrosVM

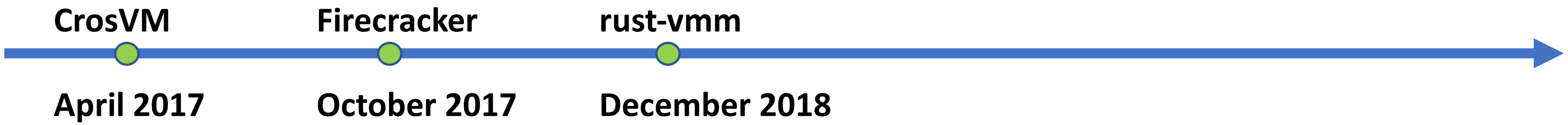
April 2017

Firecracker

October 2017

Common Virtualization Components

- KVM API wrappers
- Memory/Device model
- Virtio paravirtualization
- Kernel loader
- …



CrosVM

April 2017

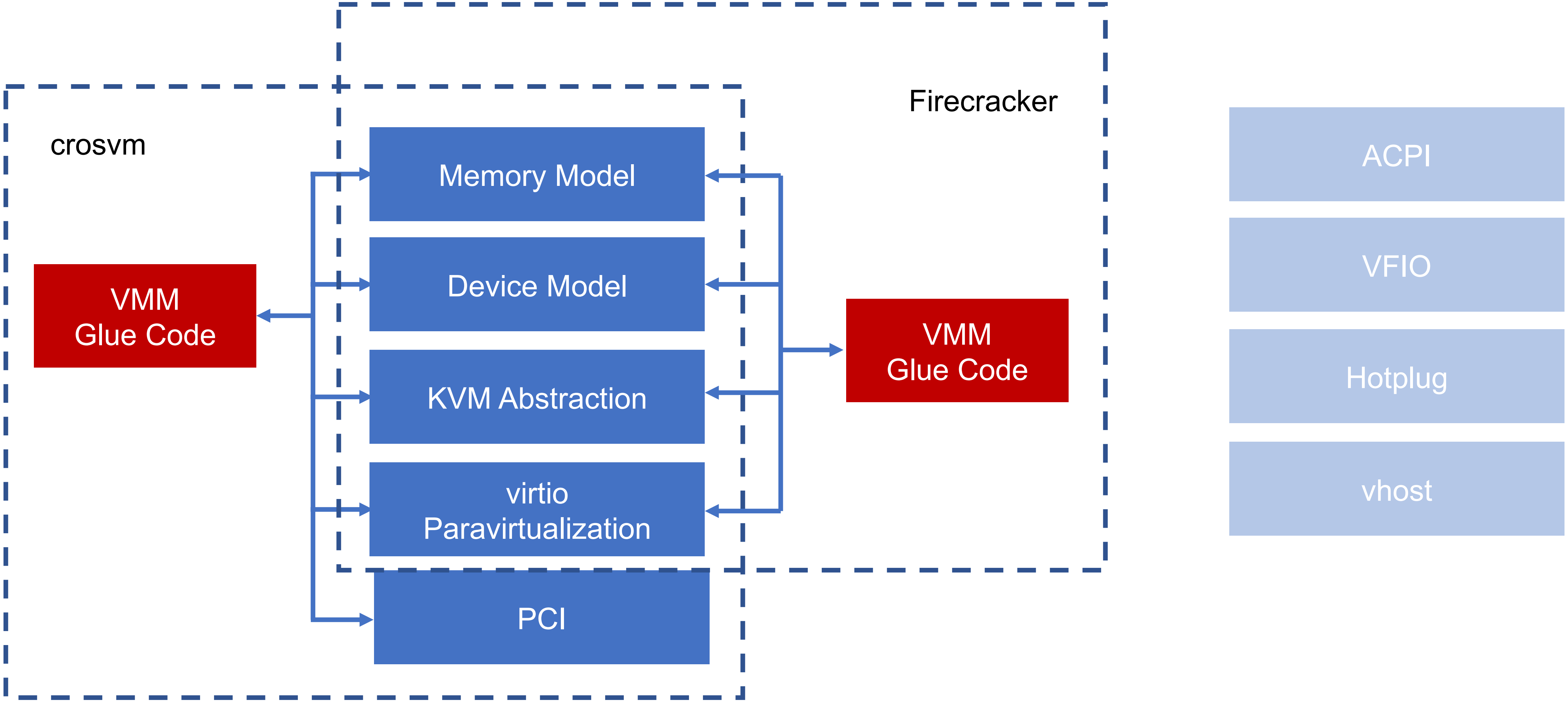
Firecracker

October 2017

rust-vmm

December 2018

Rust-VMM



crosvm
Firecracker
VMM
Glue Code
Memory Model
Device Model
KVM Abstraction
virtio
Paravirtualization
PCI
VMM
Glue Code
ACPI
VFIO
Hotplug
vhost

Cloud Hypervisor with Cloud Native

Goals

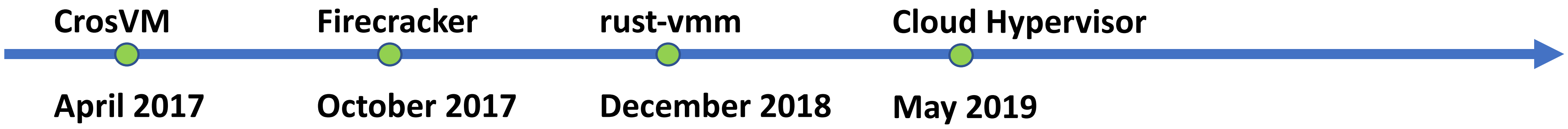
- Cloud workloads only
- No legacy hardware
- No platform emulation
- Security, simplicity, auditability
- Easy to be used in sandbox containers

Shared Pattern

- Narrow focus
- Security first
- Minimal emulation
- Hardware virtualization, no legacy
- Modularity
- rust-vmm instance for the cloud

Cloud Hypervisor

- A KVM-based Virtual Machine Monitor (VMM)
- Based on the rust-vmm crates
- Cloud workloads
  - Cloud images (Ubuntu, Centos, Windows)
  - Containers (Kata)
  - Functions
- Small, simple, secure and fast
  - Reduced footprint, boot time, TCB and code base
  - minimal emulation
  - Light and high-performance device model

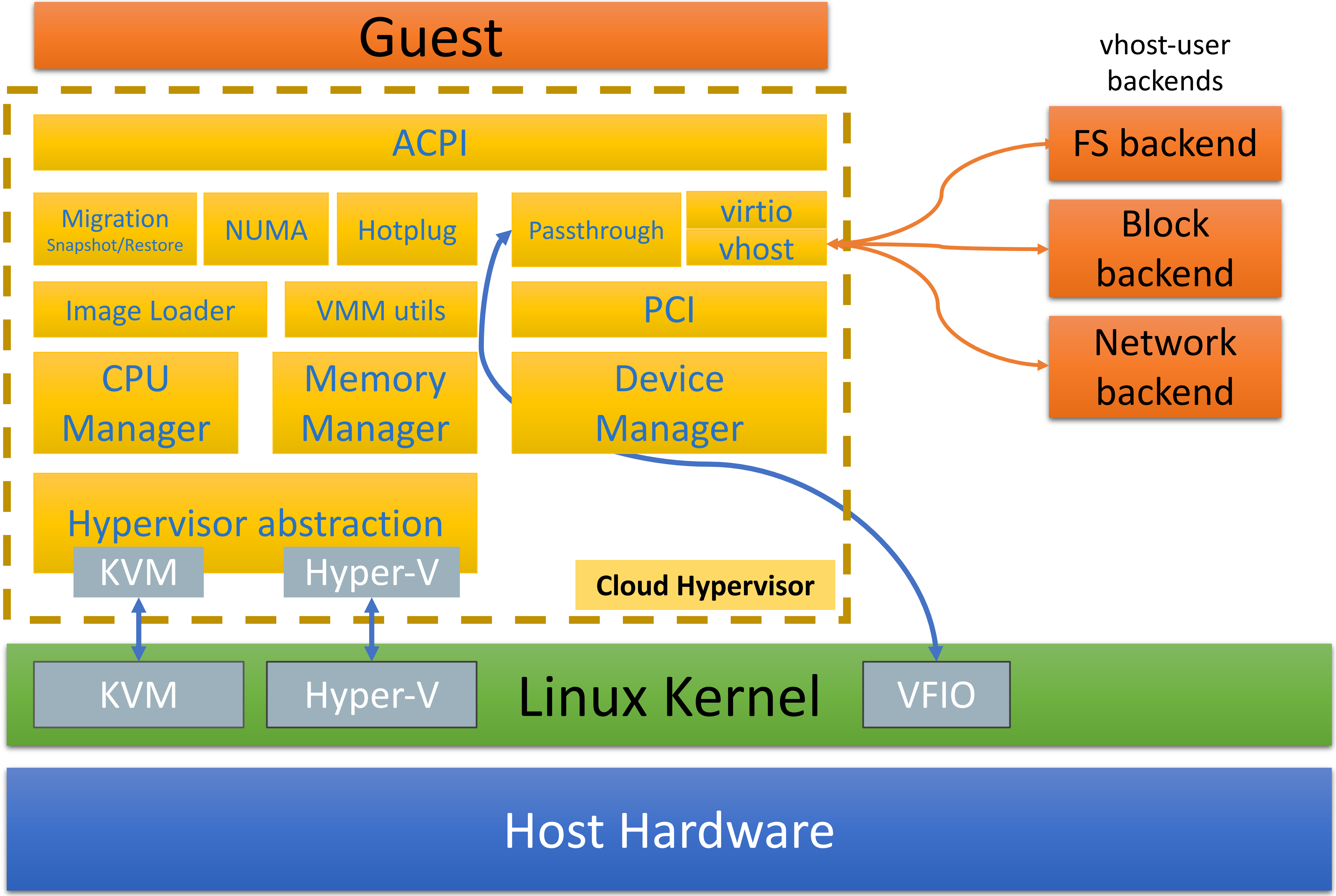


CrosVM
April 2017

Firecracker
October 2017

rust-vmm
December 2018

Cloud Hypervisor
May 2019


Guest
vhost-user backends
ACPI
Migration
Snapshot/Restore
NUMA
Hotplug
Passthrough
virtio
vhost
FS backend
Block backend
Network backend
Image Loader
VMM utils
PCI
CPU Manager
Memory Manager
Device Manager
Hypervisor abstraction
KVM
Hyper-V
Cloud Hypervisor
KVM
Hyper-V
Linux Kernel
VFIO
Host Hardware

Cloud Hypervisor Features

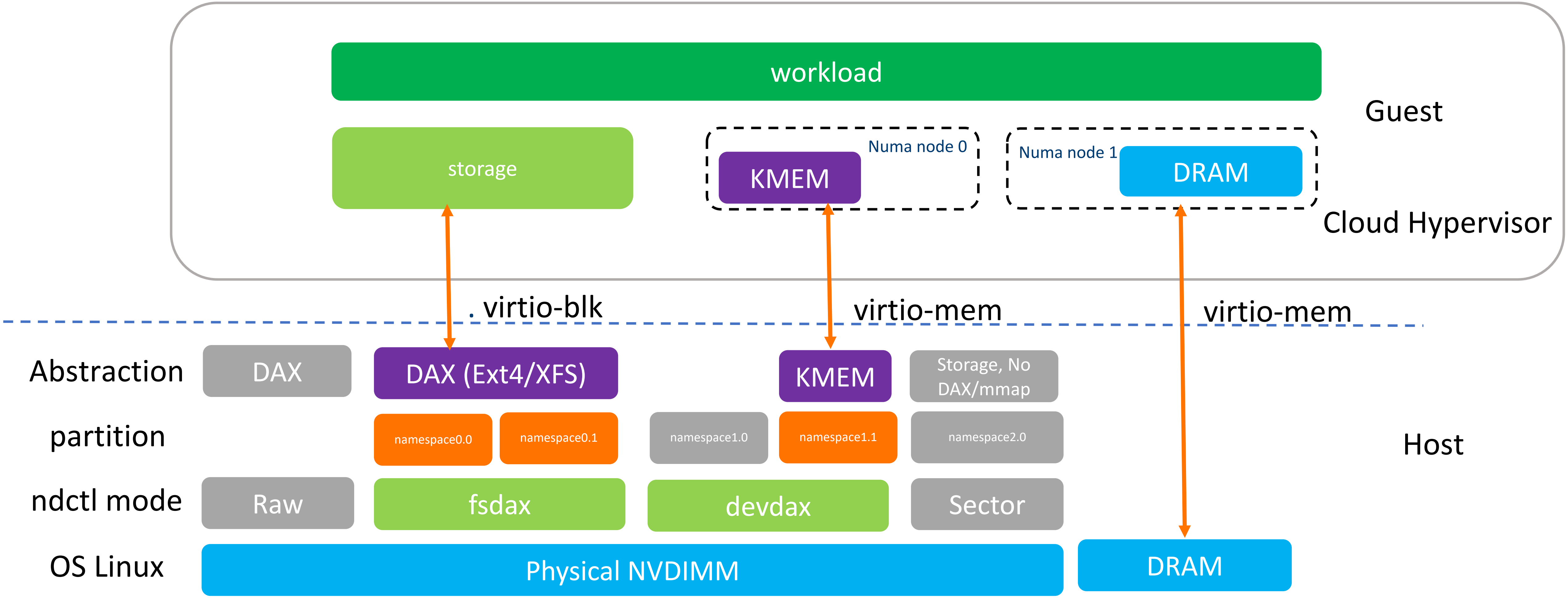
- x86_64 and aarch64
- Linux and windows guest
- Hardware-reduced ACPI
- Snapshot/Restore and Initial Live migration
- Guest NUMA topology(CPU/MEMORY AFFNITY)
- Virtio-mem with multiple NUMA nodes
- Guest Persistent memory allocation
- Nested guests (including VT-d)
- *seccomp* rules contained
- ACPI-based hot plug (CPU, memory and devices)
- REST API control interface
- Test Driven Development flow, Azure-based integration tests

Cloud Hypervisor Device Model

- PCI-based
- Virtio-mem
    - memory hotplug and resize
    - multiple numa supports
    - Different memory types including PMEM
- Virtio-fs for container image sharing
- Vhost-user for fast block/net transport with SPDK/DPDK
- Paravirtualization
    - console, iommu, mem, pmem, rng, vsock
    - *virtio* (in VMM) and vhost-user=true
    - *vhost-user* (Rust backends)
    - Multi-queue, multi-threaded
- Device passthrough through VFIO
- IO_uring support
- Minimal legacy devices support
    - Serial, CMOS, ACPI virtual device

Feature enabling in CLH: PMEM and vHost as example

Feature enabling in CLH: PMEM



Community & Roadmap

Cloud Hypervisor Project Status

- Currently at version 0.11.0
  - One new release every 6 weeks
  - Under the independent cloud-hypervisor github organization
- Intel, ARM, Alibaba, Red Hat, Oracle, Microsoft, Coder, Phytium, etc
- New governance model
  - Inspired by Kata Containers model
  - Architecture committee
  - Distributed commit access (Not only Intel)

Cloud Hypervisor Roadmap

- TDX and Total Memory Encryption
- Live Migration optimization
- VMM live update
- VM monitoring
- Net and block IO rate-limiting
- …

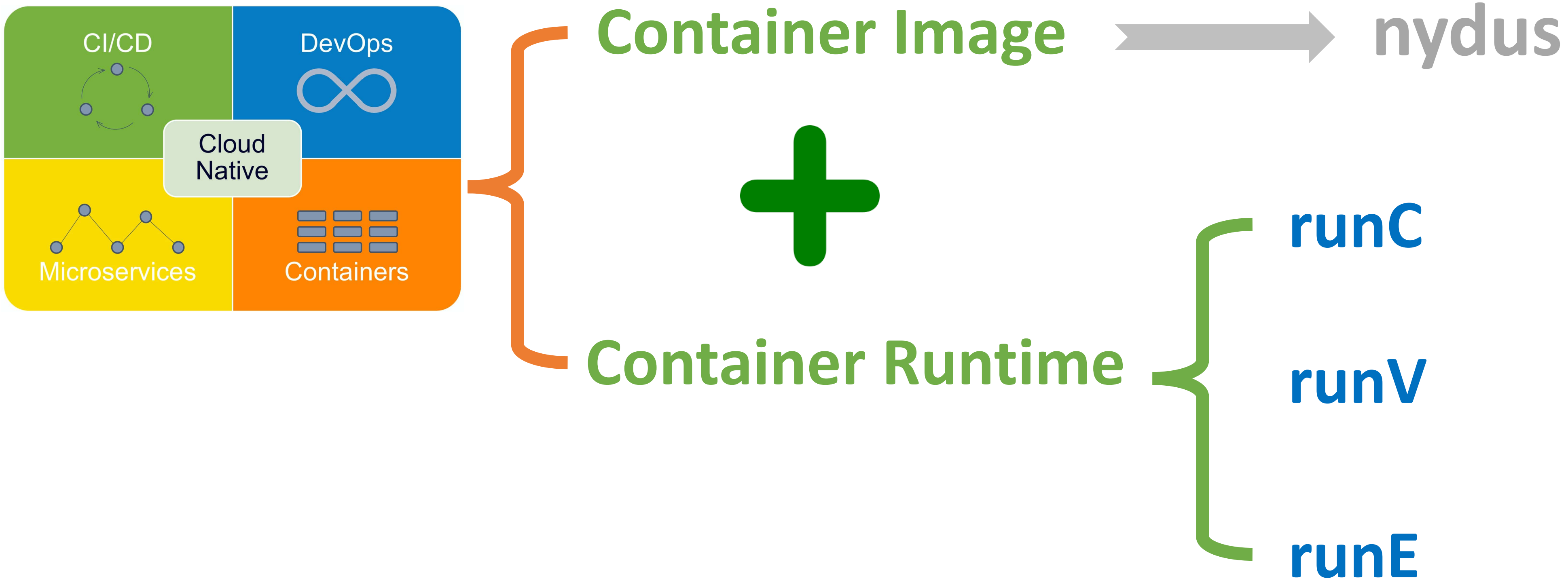
Cloud Native Hypervisor

Cloud
Hypervisor

VS

Cloud
Native

When Cloud Hypervisor falls in love with Cloud Native, they become “Cloud Native Hypervisor”



CI/CD

DevOps

Cloud Native

Microservices

Containers

Container Image → nydus

+

Container Runtime

runC

runV

runE

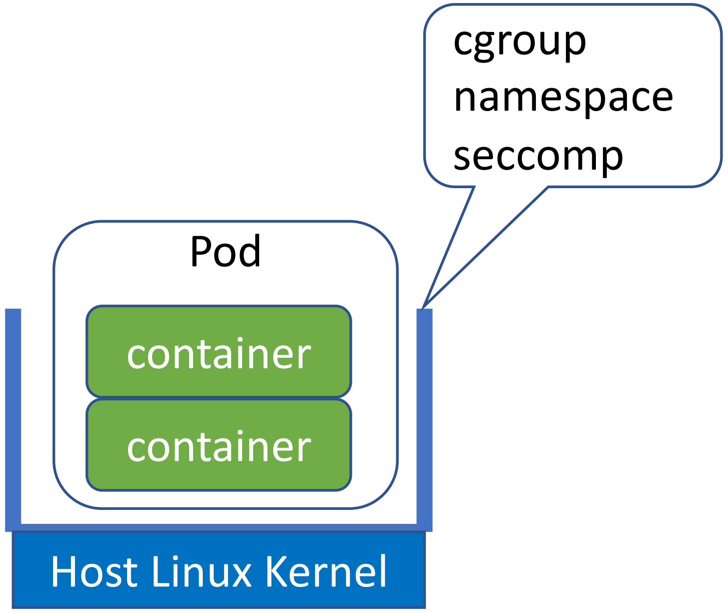
When OS virtualization cannot satisfy Cloud Native’s requirements, what the plan?

OS
Virtu ✗ ion
(runC)

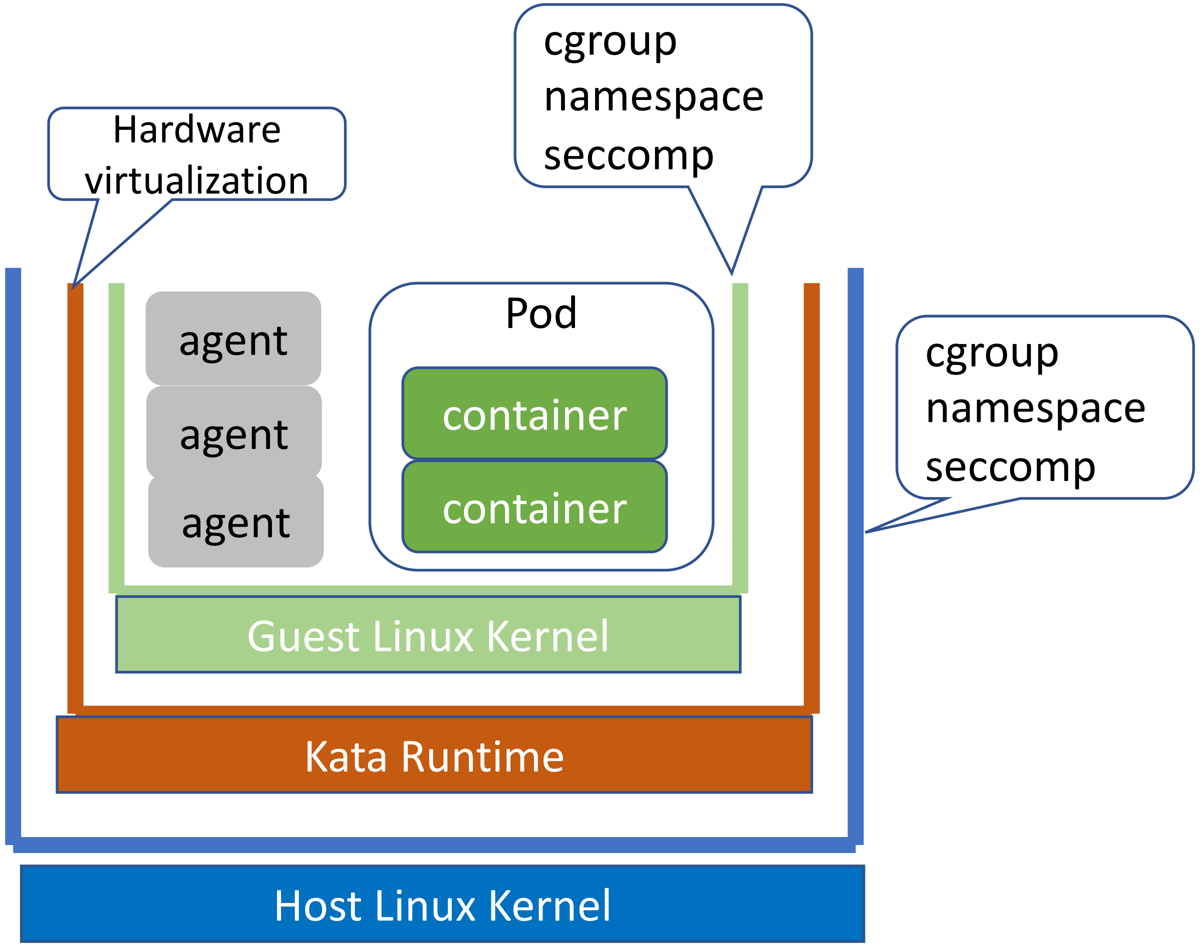
→

Hardware
Virtualization
(runV/Kata Containers)

Is runV(Kata Containers) what the user needs? Maybe not!

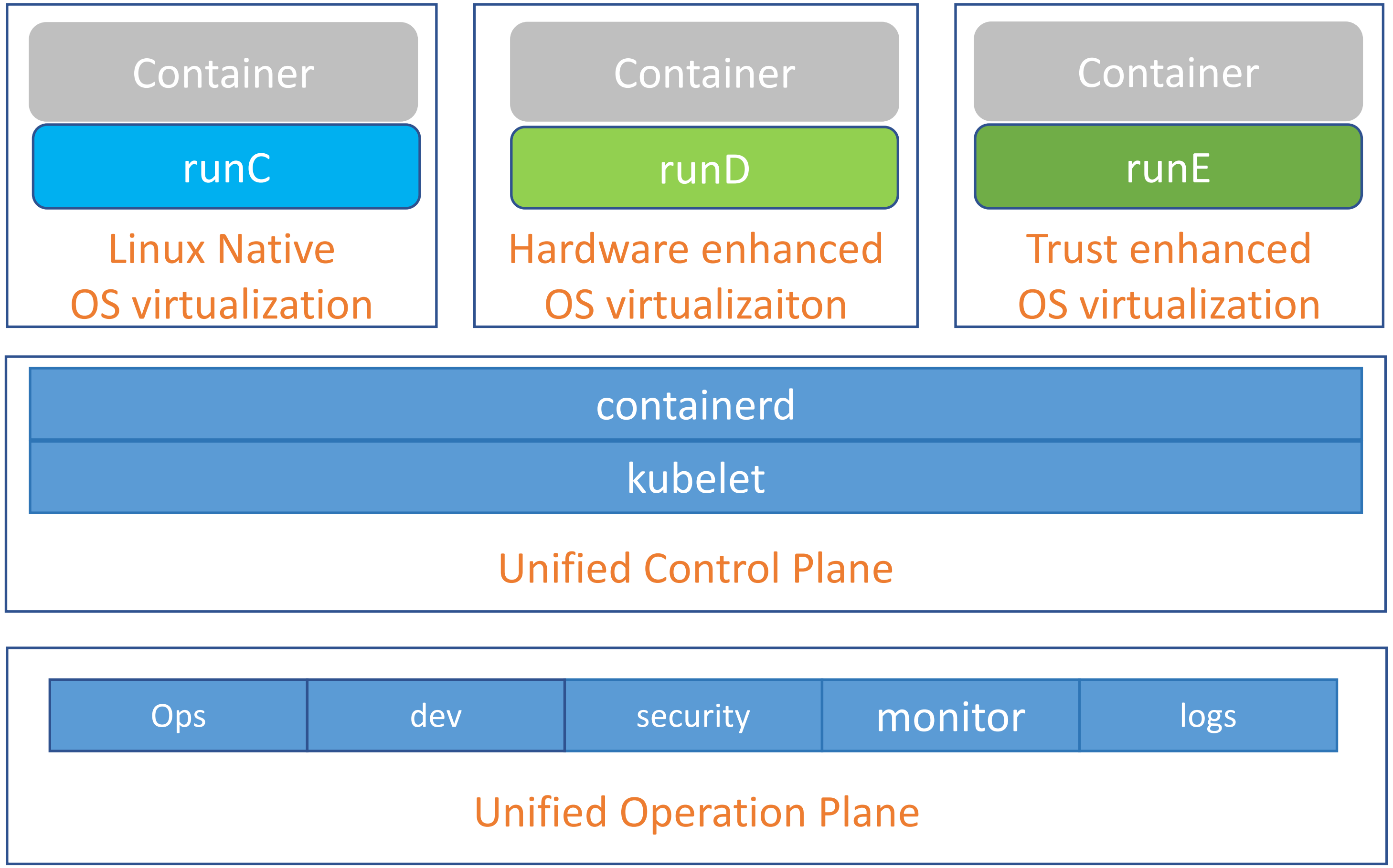


runC



Kata Containers

OS virtualization based container runtime



Let’s talk about

Runtime Virtualization

OS Virtualization

- Linux runC
- Solaris zone
- Windows Server Container

Hardware Virtualization

- Qemu/kvm
- HyperV
- PowerVM
- Vmware ESXi

Hardware Virtualization + VTx = Hardware Enhanced Hardware Virtualization

OS Virtualization + VTx = Hardware Enhanced OS Virtualization

Then, what’s Cloud Native Hypervisor?
It’s up to you to define it☺

Q & A