ipdk Infrastructure Programmer Development Kit

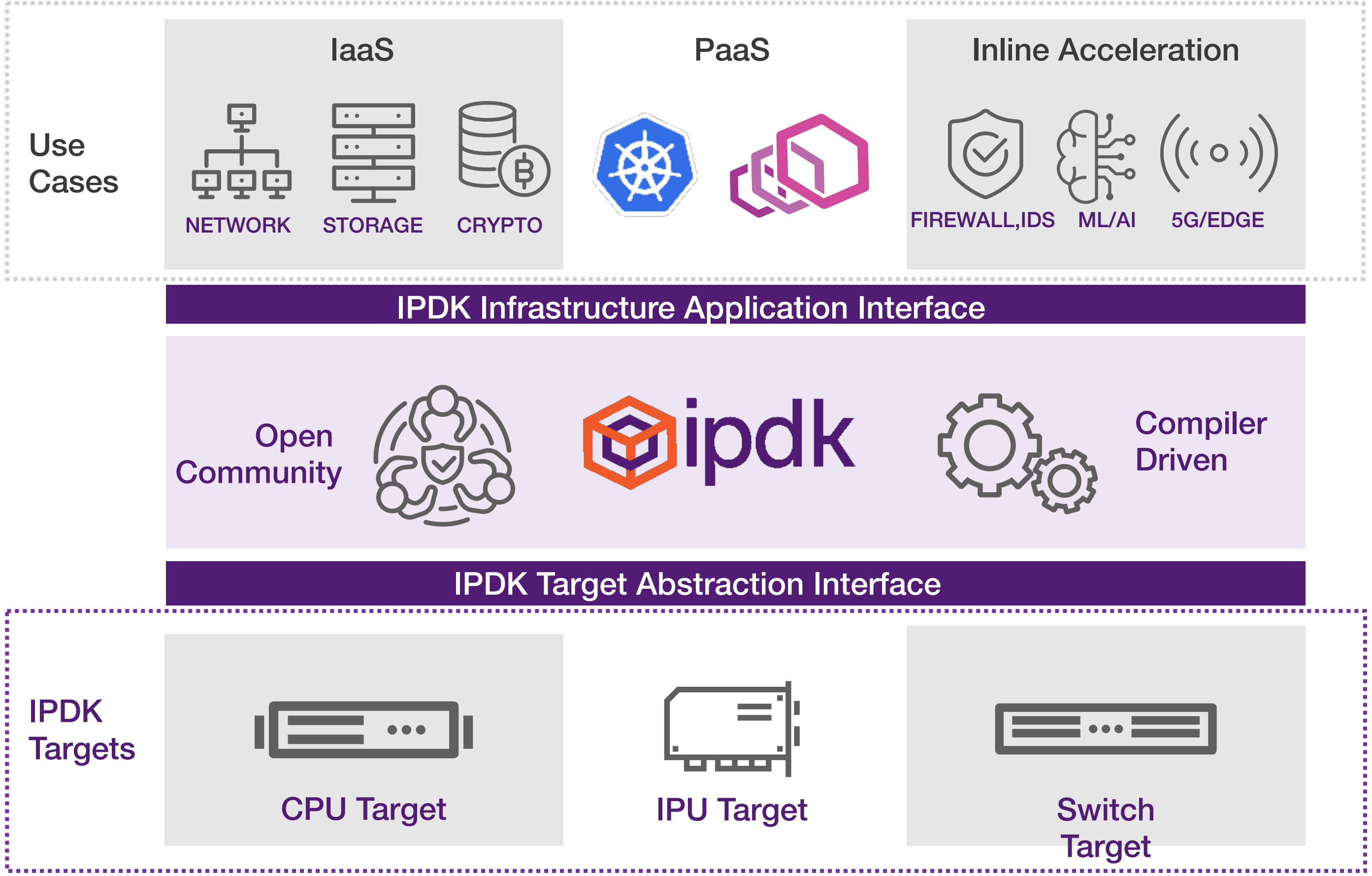
Introduction Infrastructure Programming

Presented by // DAN DALY

OCTOBER, 2021

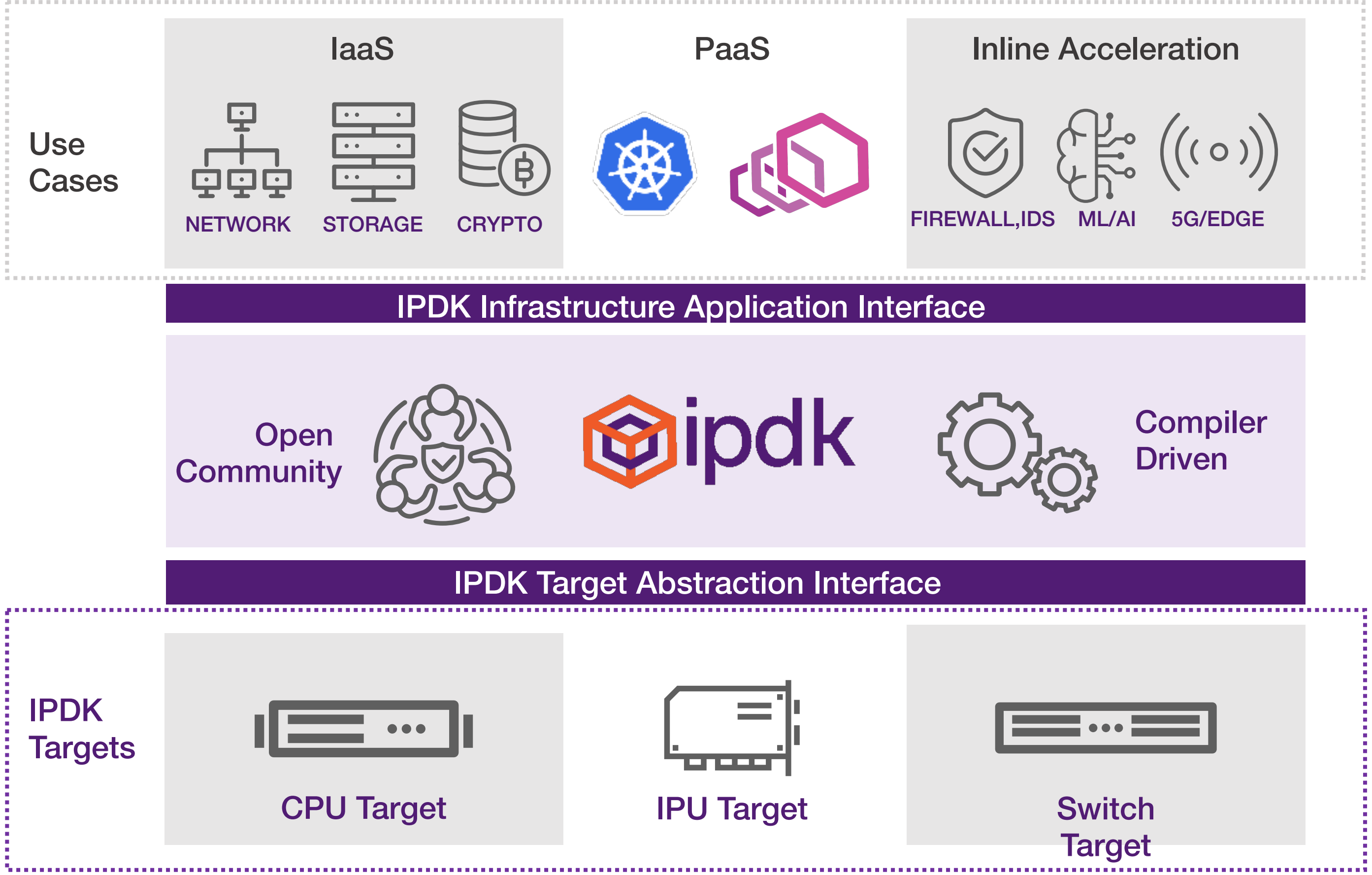
IPDK Overview

IPDK is a **community-driven target agnostic** framework for **infrastructure programming** that runs on a CPU, IPU, DPU, or switch.



Agenda

- Charter
- Approach
- Use cases
- Open-source development
  - Example: Infrastructure-as-a-Service
- Roadmap
- Call to action



IPDK Charter

Open Community:

Open-source project with standard interfaces

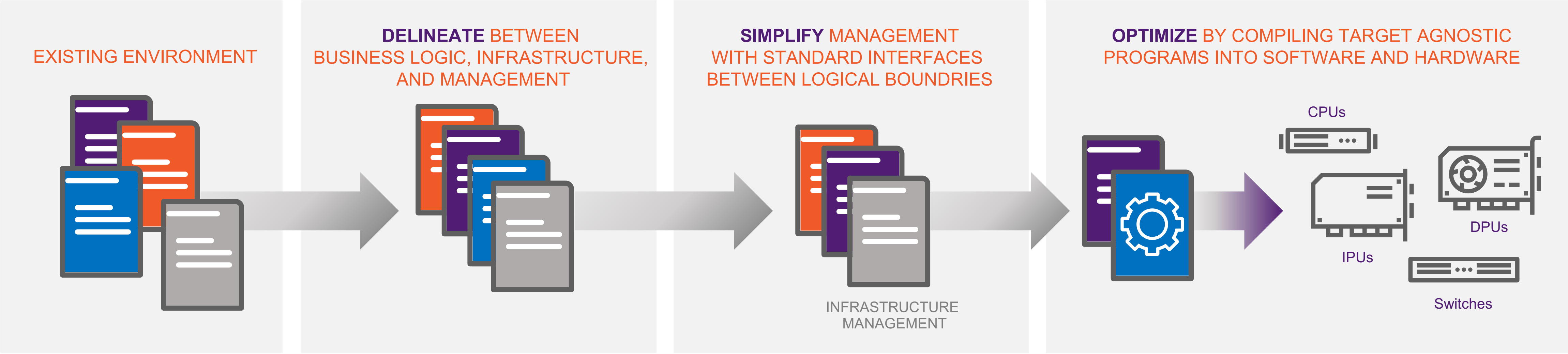
Target Agnostic:

Any CPU, IPU, DPU, or switch

Use-Case Driven:

Use programmability for adaptability

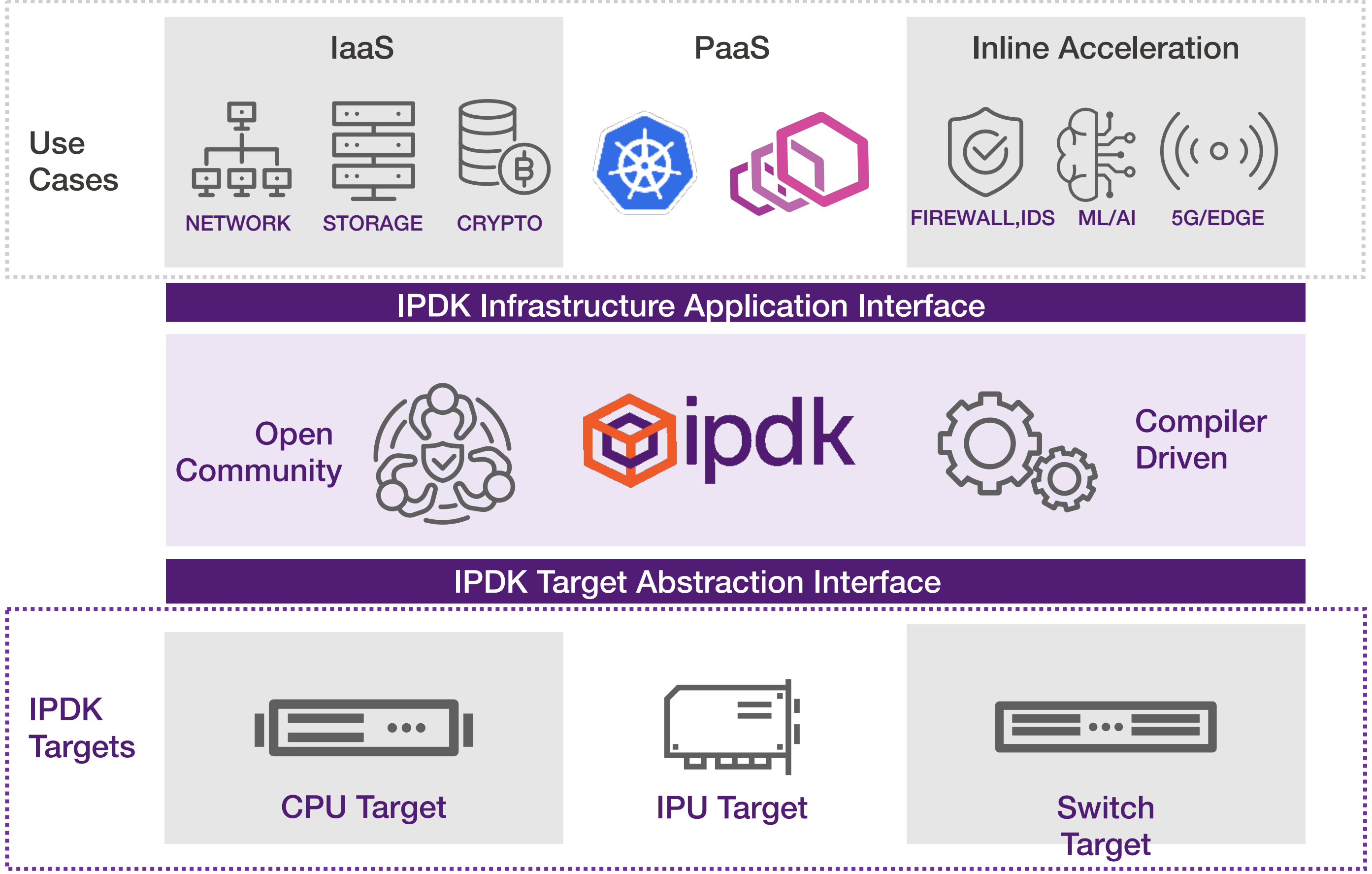
Approach


EXISTING ENVIRONMENT
DELINEATE BETWEEN BUSINESS LOGIC, INFRASTRUCTURE, AND MANAGEMENT
SIMPLIFY MANAGEMENT WITH STANDARD INTERFACES BETWEEN LOGICAL BOUNDRIES
INFRASTRUCTURE MANAGEMENT
OPTIMIZE BY COMPILING TARGET AGNOSTIC PROGRAMS INTO SOFTWARE AND HARDWARE
CPUs
DPUs
IPUs
Switches

1. **Delineate** Business Logic vs. Infrastructure
2. **Simplify** Infrastructure Management
3. **Optimize** using a Compiler-Driven Target Abstraction

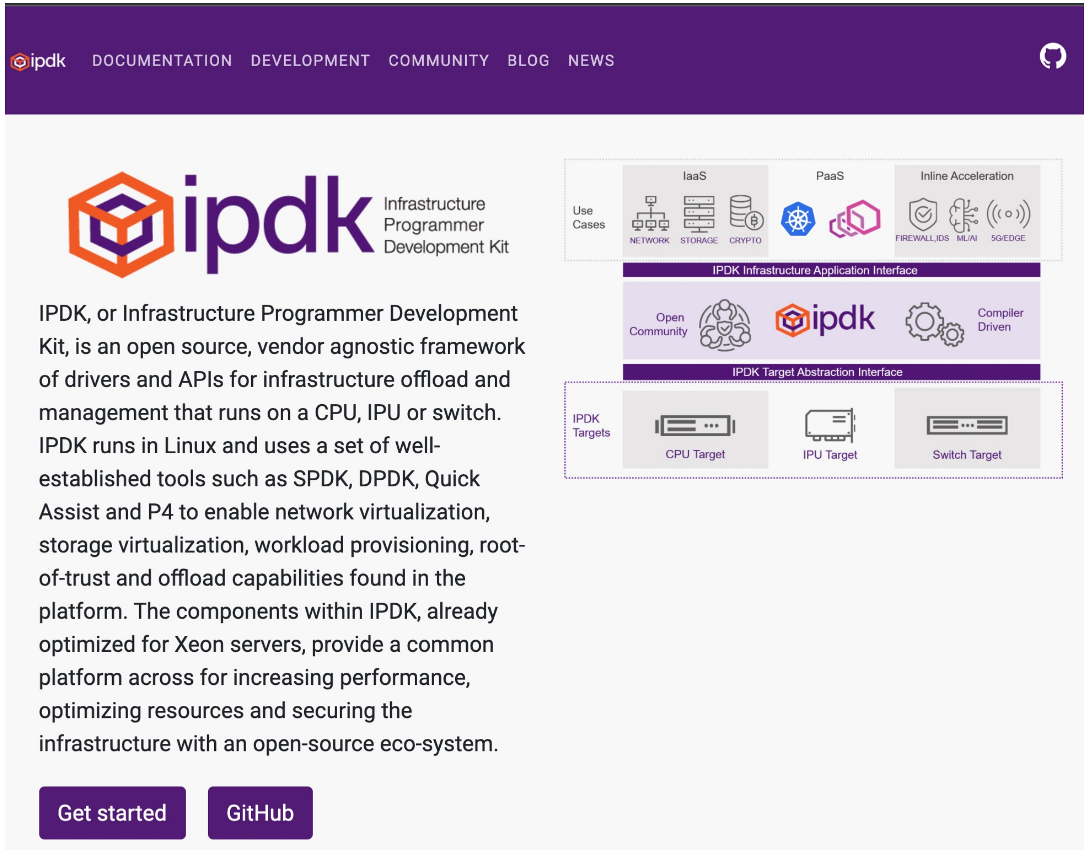
Use-Case Driven

1. **Infrastructure-as-a-Service**
   Virtual networking, storage and Crypto across VMs, containers and bare metal
2. **Platform-as-a-Service**
   Container Networking (Kubernetes) Sidecars (Envoy, MongoDB)
3. **Inline Acceleration**
   Firewall, IDS, Network Telemetry 5G/Wireless Infrastructure, AI/ML



Open-Source Development

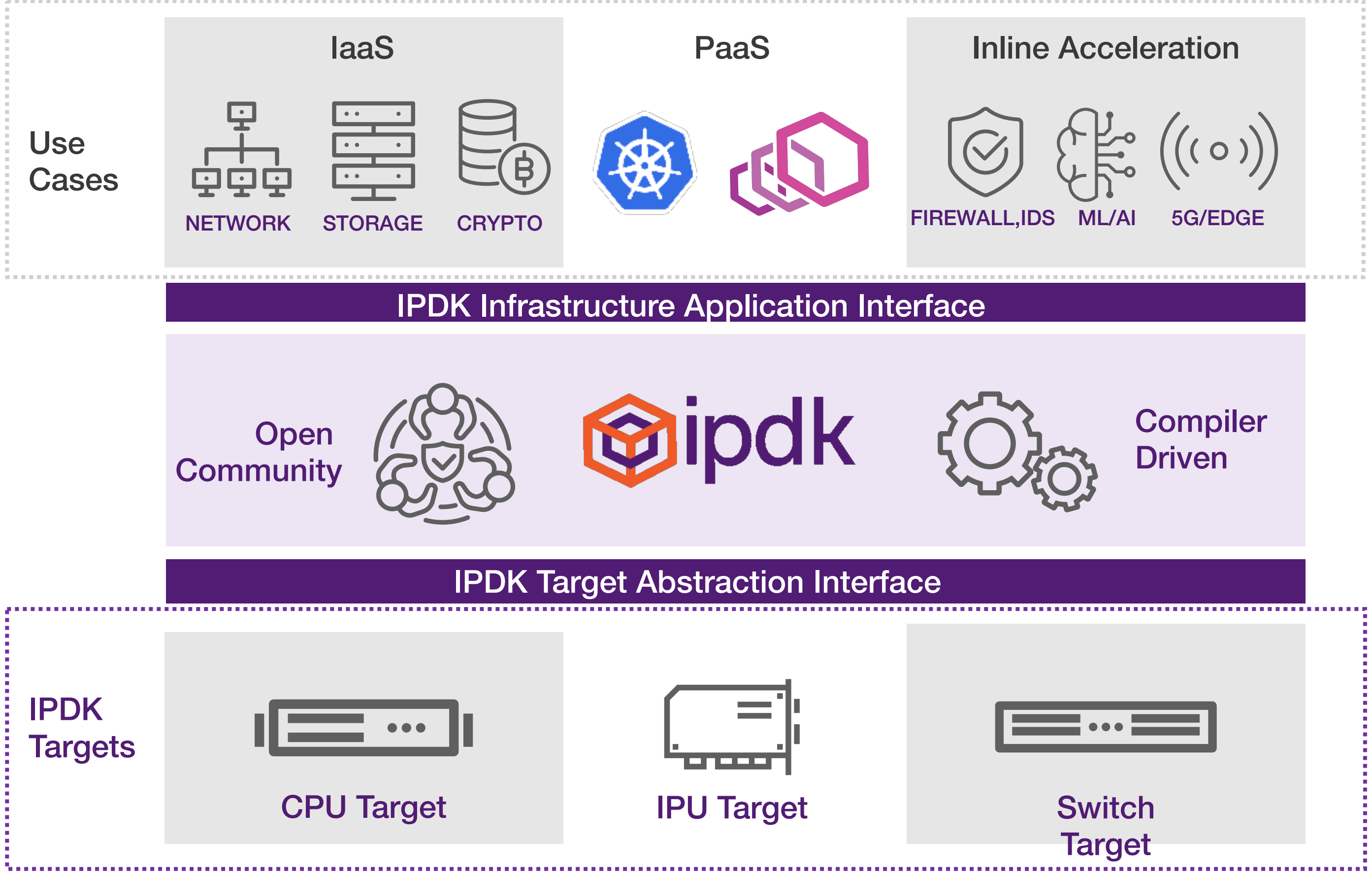
- **Recipes**
  Delineate, simplify, and optimize for each major use case and functional area
- **Open-Source Development & Governance**
  New patches, agents, and interface support
  Open-Source definitions of interfaces
  Dockerfiles and pre-built containers
- **Development has started, come join!**
  Collaborate on Slack , Github & IPDK.io



IPDK, or Infrastructure Programmer Development Kit, is an open source, vendor agnostic framework of drivers and APIs for infrastructure offload and management that runs on a CPU, IPU or switch. IPDK runs in Linux and uses a set of well-established tools such as SPDK, DPDK, Quick Assist and P4 to enable network virtualization, storage virtualization, workload provisioning, root-of-trust and offload capabilities found in the platform. The components within IPDK, already optimized for Xeon servers, provide a common platform across for increasing performance, optimizing resources and securing the infrastructure with an open-source eco-system.

IPDK Standard Interfaces

- **Infrastructure Application Interface**
  Devices & Services for Workload Apps
  Platform Capabilities Offered to Mgmt
- **Target Abstraction Interface (TAI)**
  Target Capabilities Offered to IPDK



Take-Home Demo

Download Containers

- IPDK Container
- Traffic Gen Container
- Storage Target Container

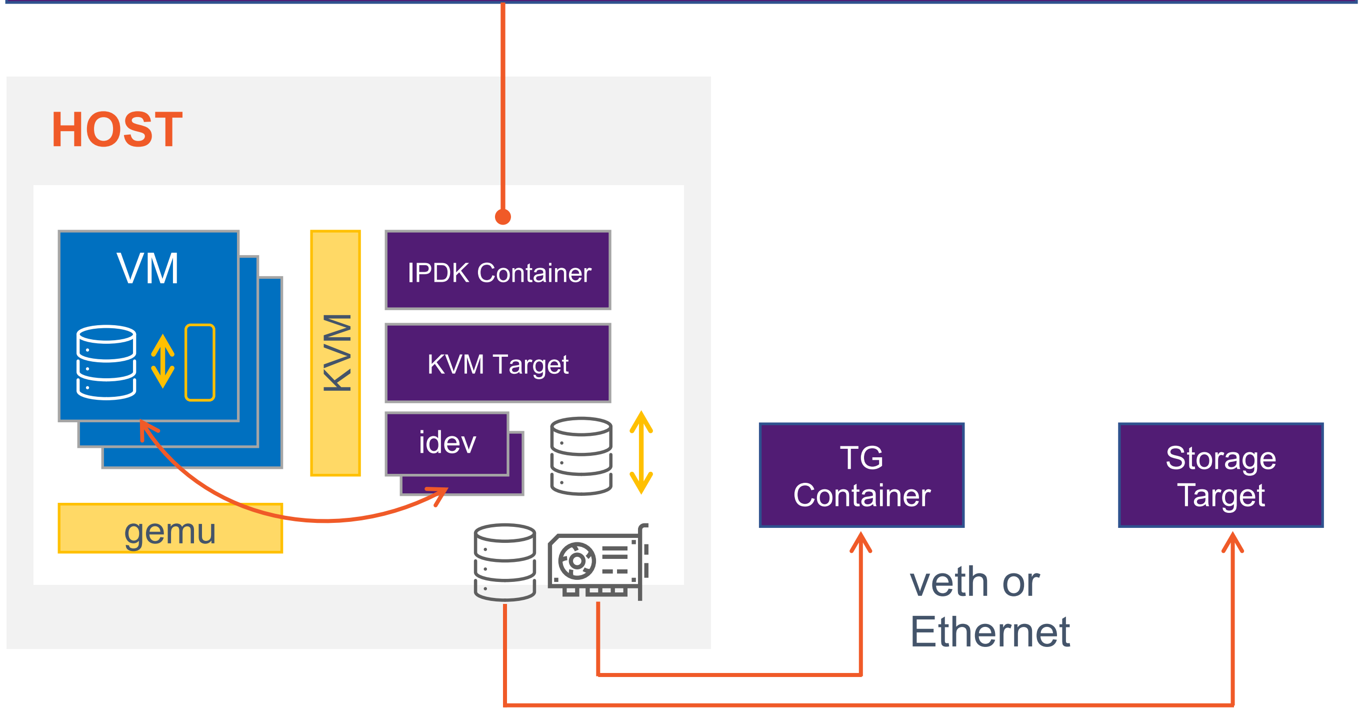
Create Virtual Devices

- Create virtual port
- Create virtual disk

Generate network traffic and storage read/writes

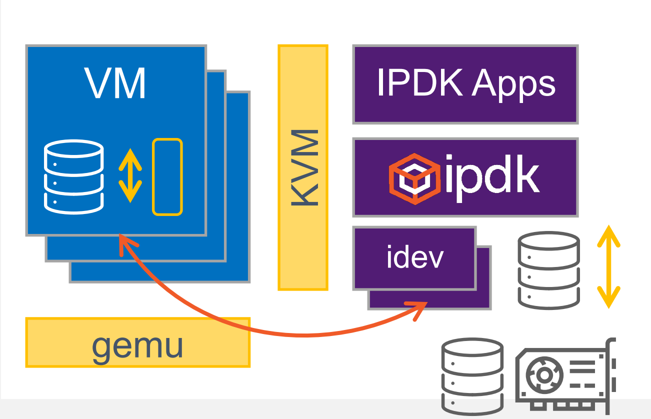
OpenConfig RPC or CLI

```
> create port host1 virtio-net 10Gbps netdev1
> create disk host1 virtio-blk 100KIOPS bdev1
```



Where Can IPDK Run?

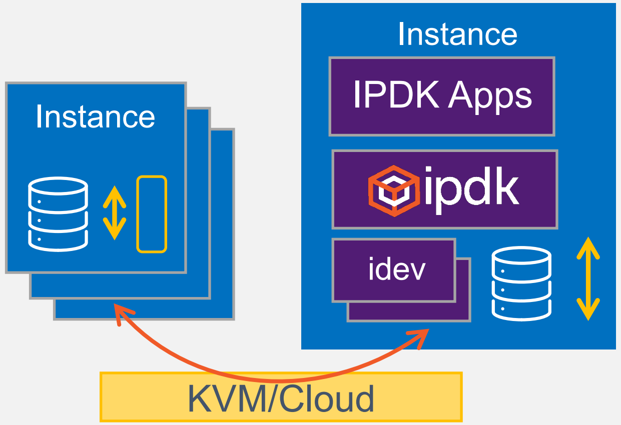
KVM HOST



- Insert devices into **VMs**
- Sockets in/out of **VM** apps/containers
- **Direct attached** devices (drives, NICs, Accel)

Software Hypervisor
Software Virtual Switch

VIRTUAL INSTANCE

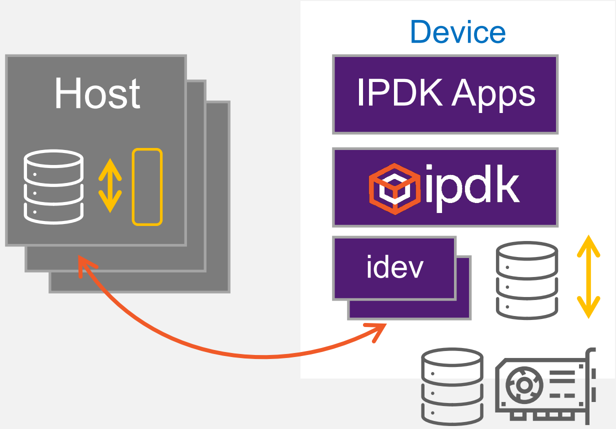


- Insert devices into **instances** (VM, bare metal)
- Sockets in/out of **instance** apps/containers
- **Virtual** devices (drives, NICs, Accel)

Cloud Hypervisor
Cloud Virtual Switch

KVM can also run in each instance

PHYSICAL DEVICE



- Insert devices into **hosts** (VM, bare metal)
- Sockets in/out of **host or VM** apps/containers
- **Direct and virtual** devices (drives, NICs, Accel)

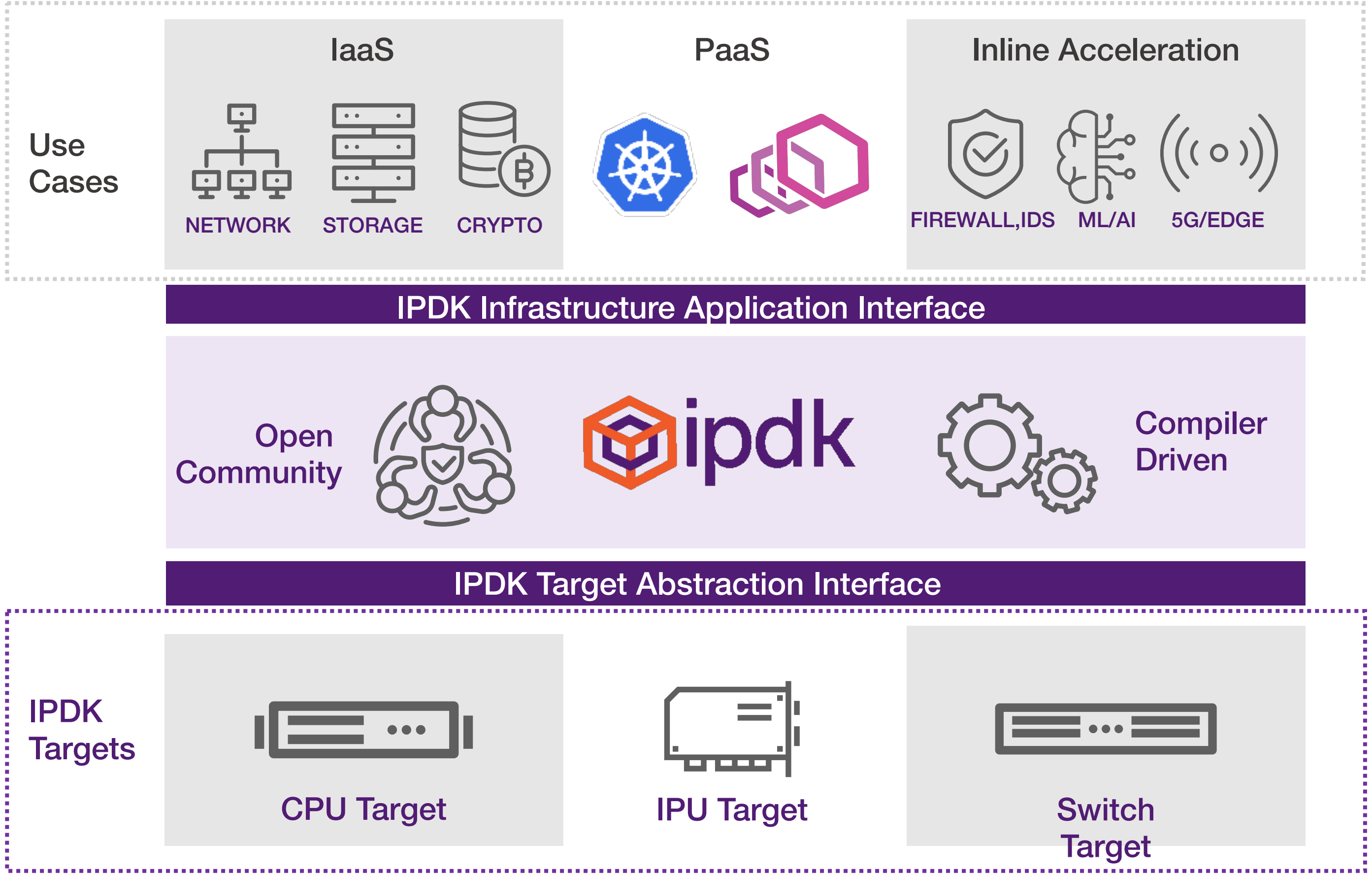
Hardware Hypervisor
Hardware Virtual Switch

KVM can also run in each instance

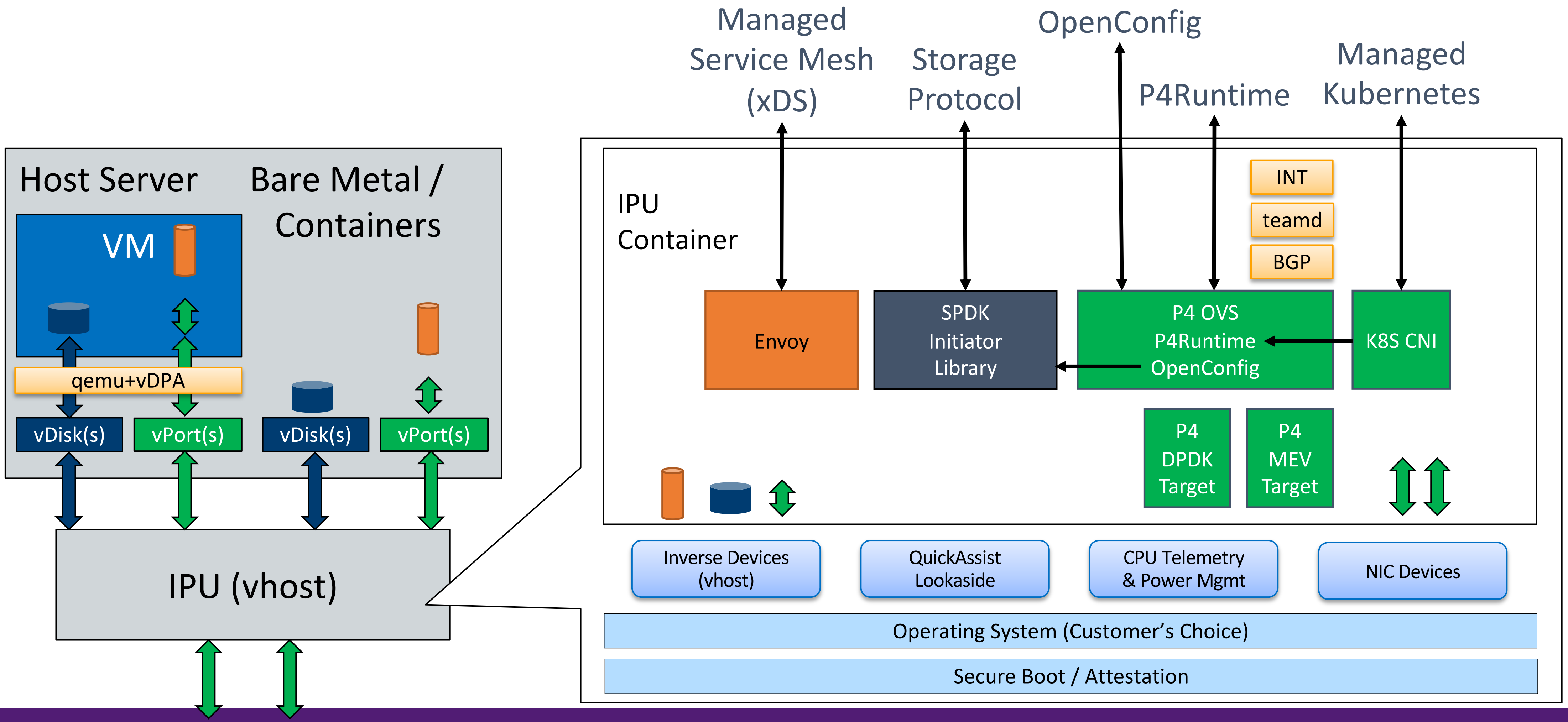
Call to Action

Come Join the Community

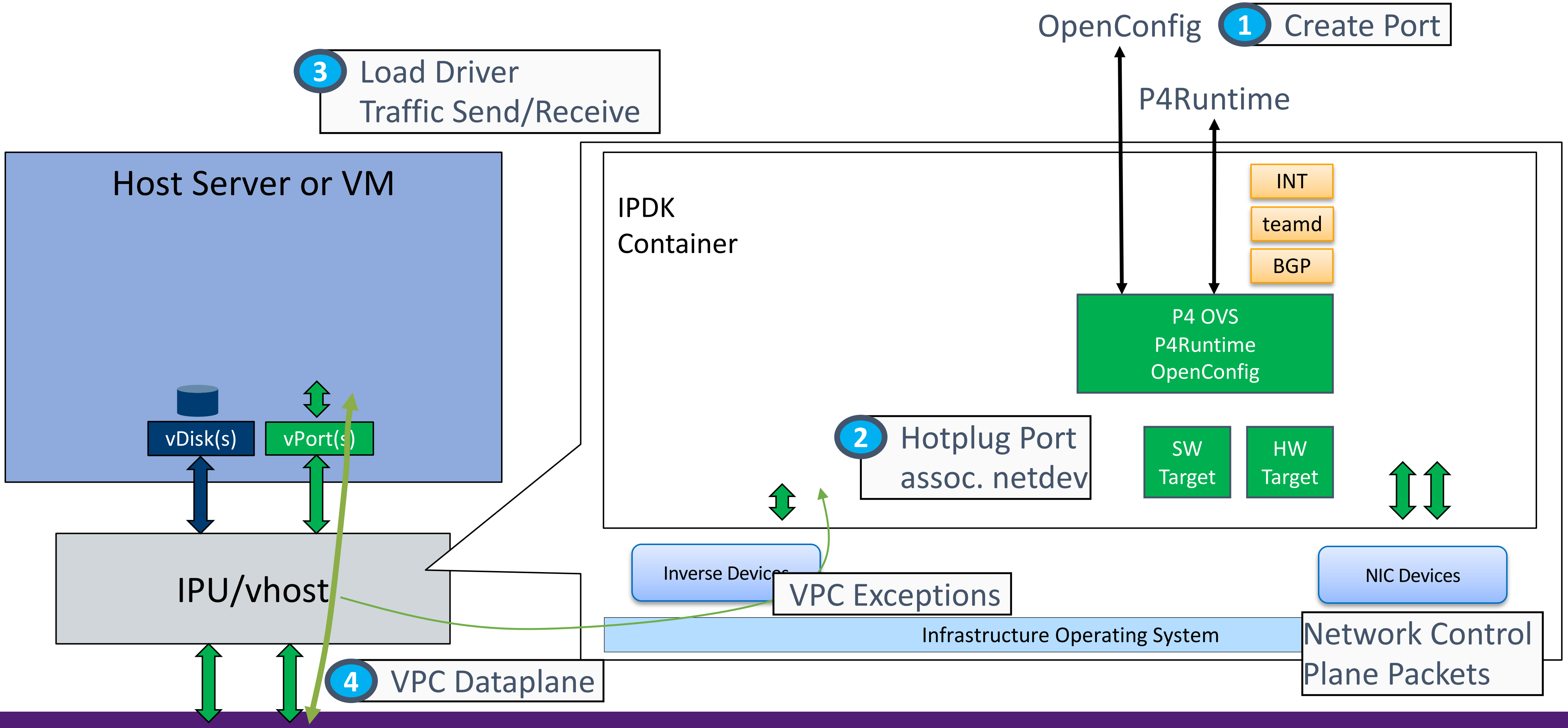
- Develop IPDK-compatible Applications
- Contribute an IPDK Target
  - Software, Switch, IPU, DPU, etc.
- Develop New Recipes
- Contribute to Existing Recipes
- Contribute to the Interfaces



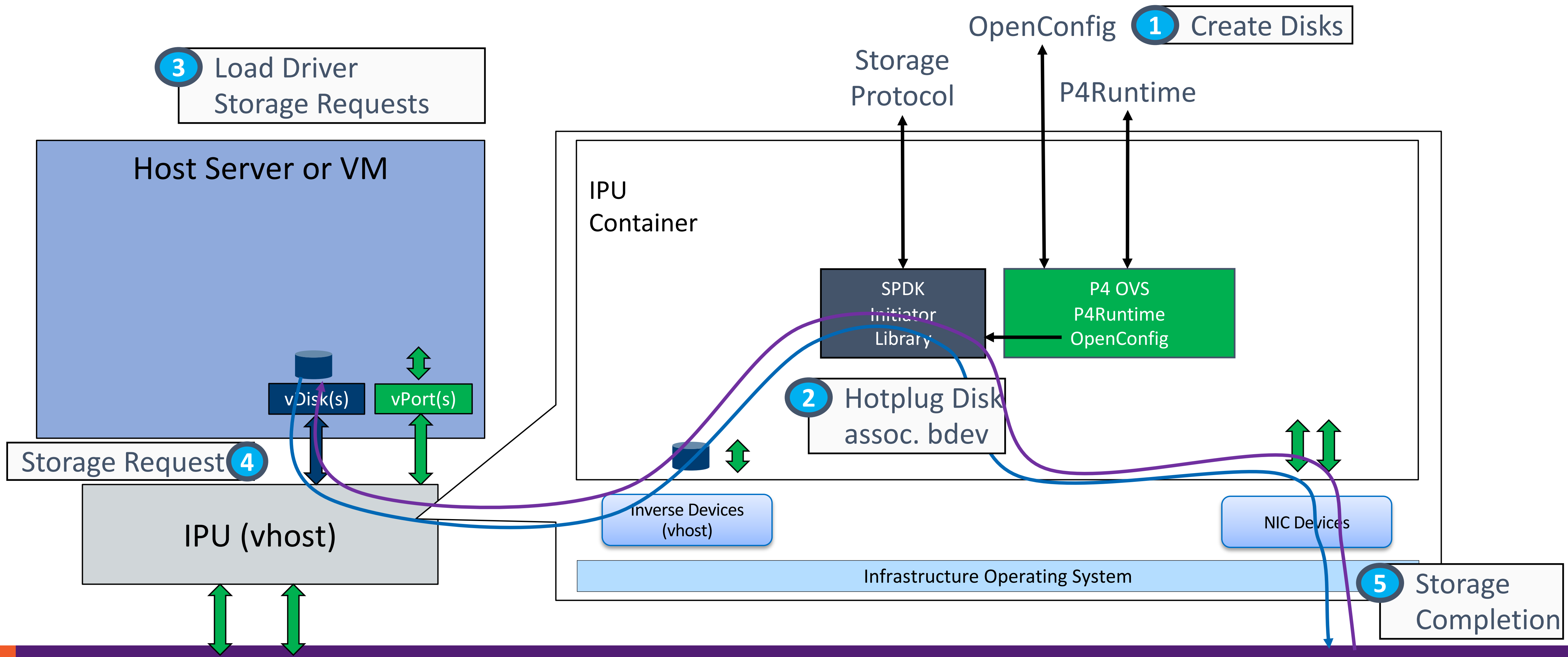
IPDK Container



IaaS – Network Virtualization

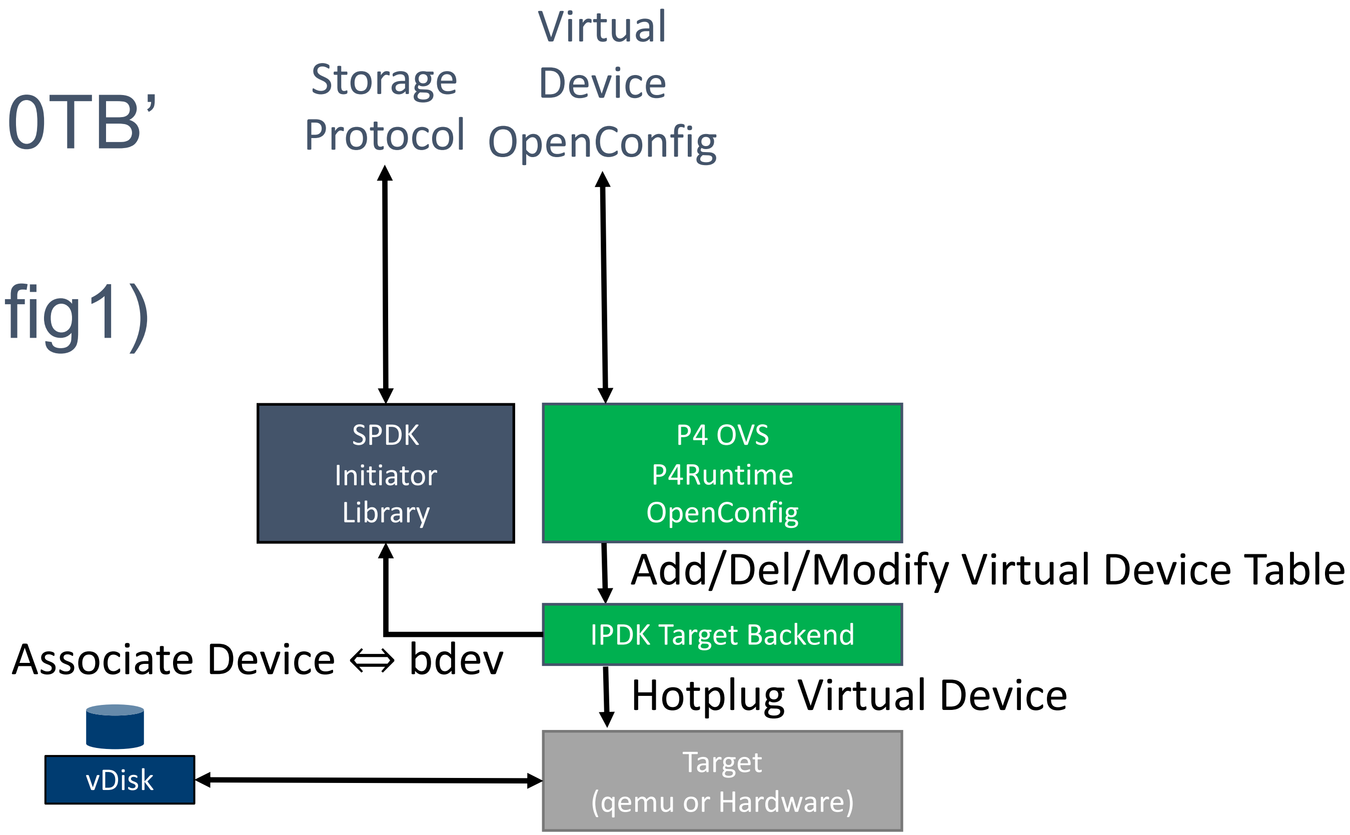


IaaS – Storage Virtualization

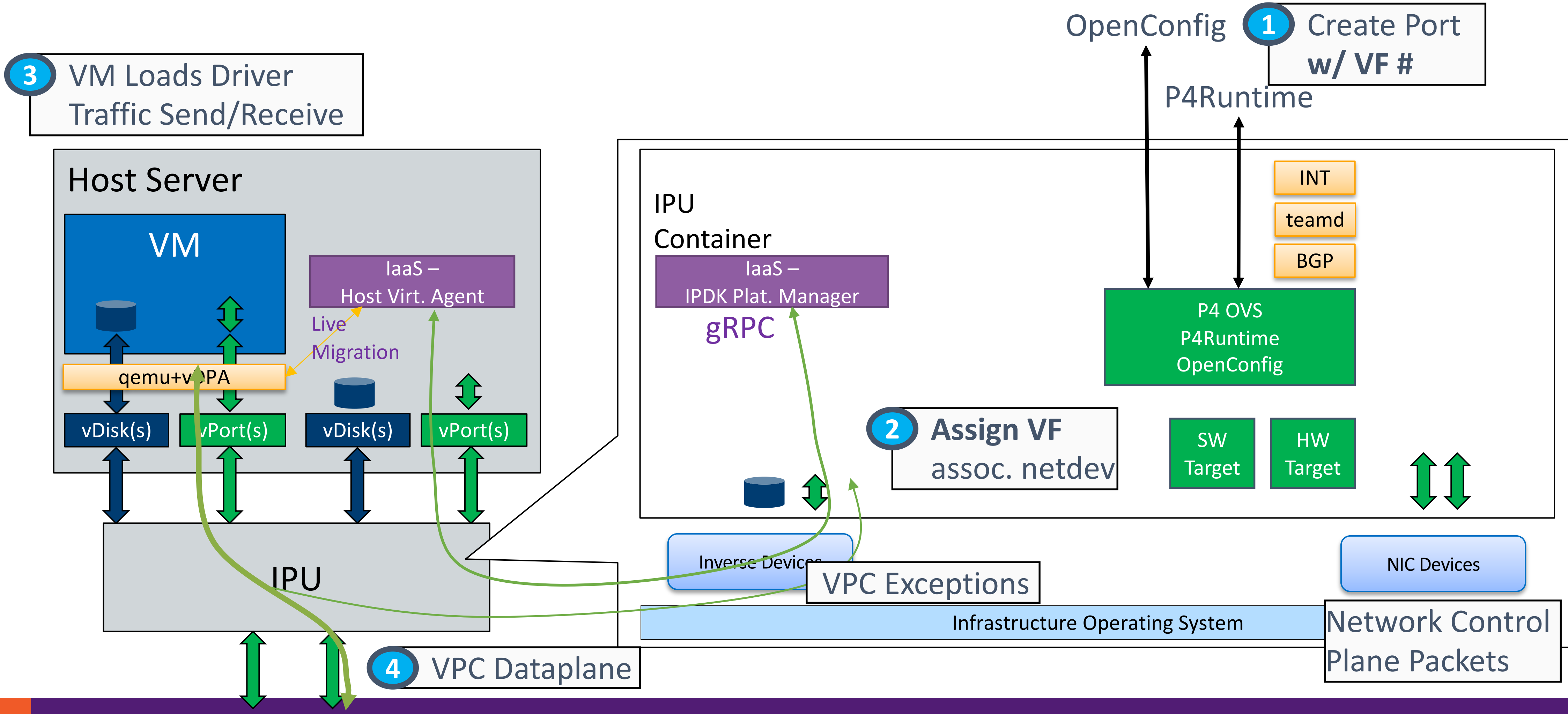


IaaS – Storage Virtualization - Interfaces

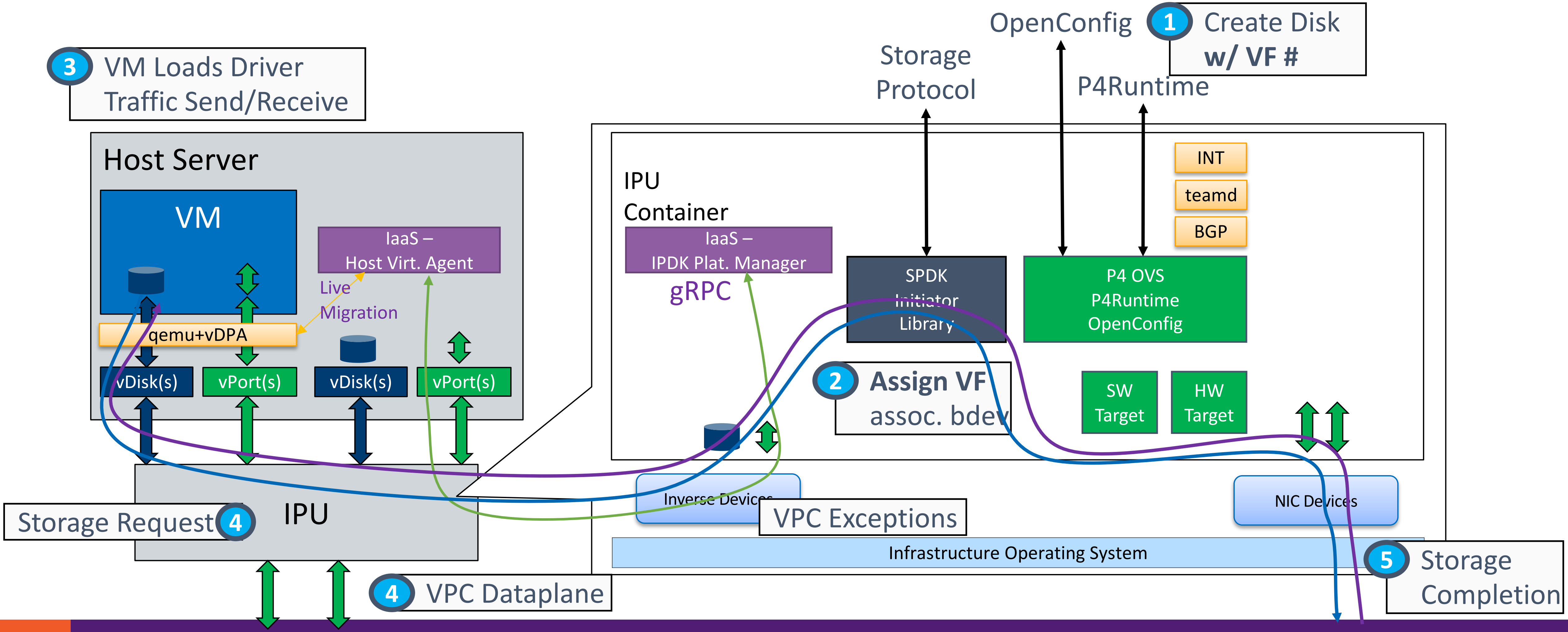
1. OpenConfig Operation
   ’add virtio-blk disk1 bdev5 10TB’
2. TDI (Table Driven Interface)
   ‘addEntry(vdev, “disk1”, config1)
3. Backend Device Operation
   - KVM: qemu hotplug
   - Canyon: pci hotplug
   - MEV: simulated hotplug
4. SPDK Operation
   associate device w/ ‘bdev5’


Storage Protocol
Virtual Device OpenConfig
SPDK Initiator Library
P4 OVS P4Runtime OpenConfig
Add/Del/Modify Virtual Device Table
IPDK Target Backend
Associate Device ⇔ bdev
Hotplug Virtual Device
vDisk
Target (qemu or Hardware)

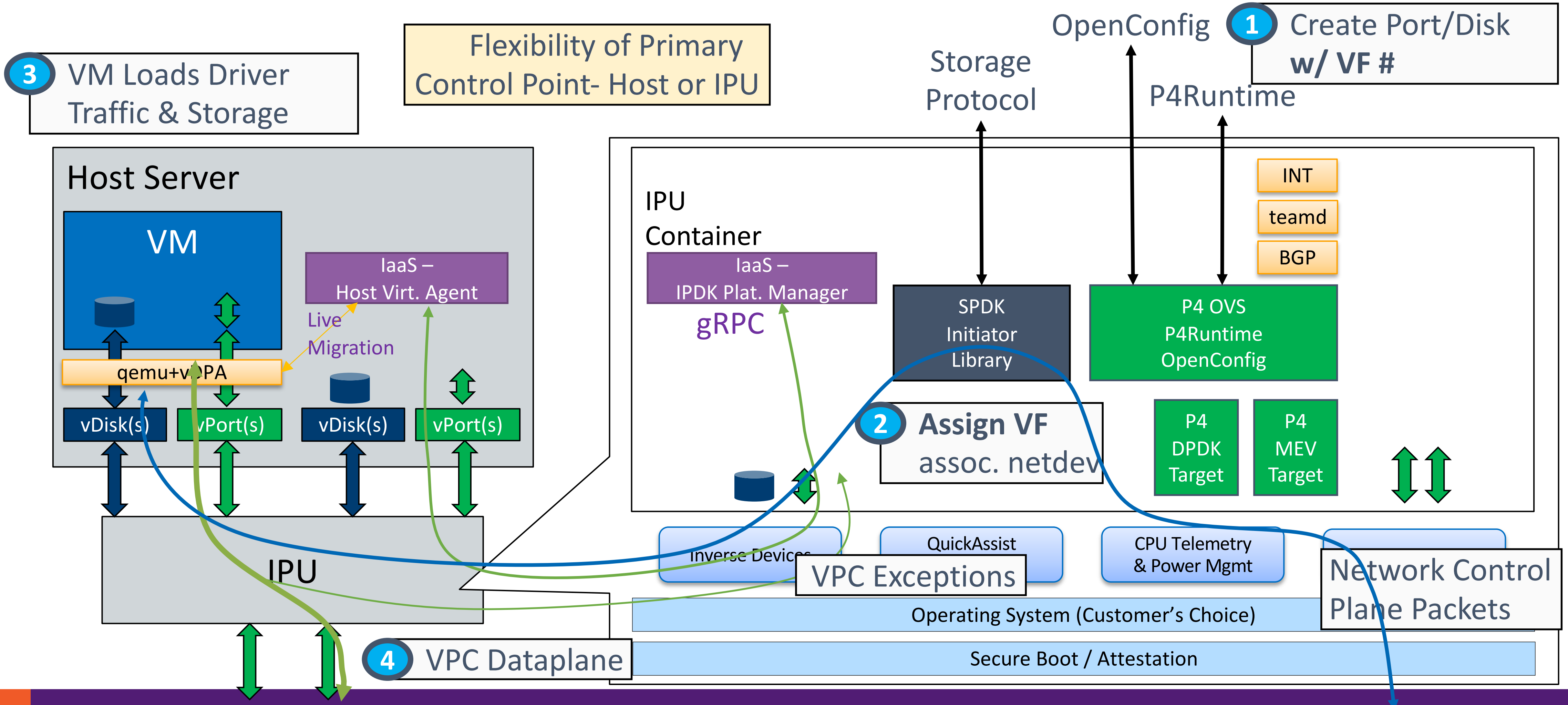
IaaS – Host Virtualization – Network



IaaS – Host Virtualization – Disk



IaaS – Host Virtualization – Net+Disk

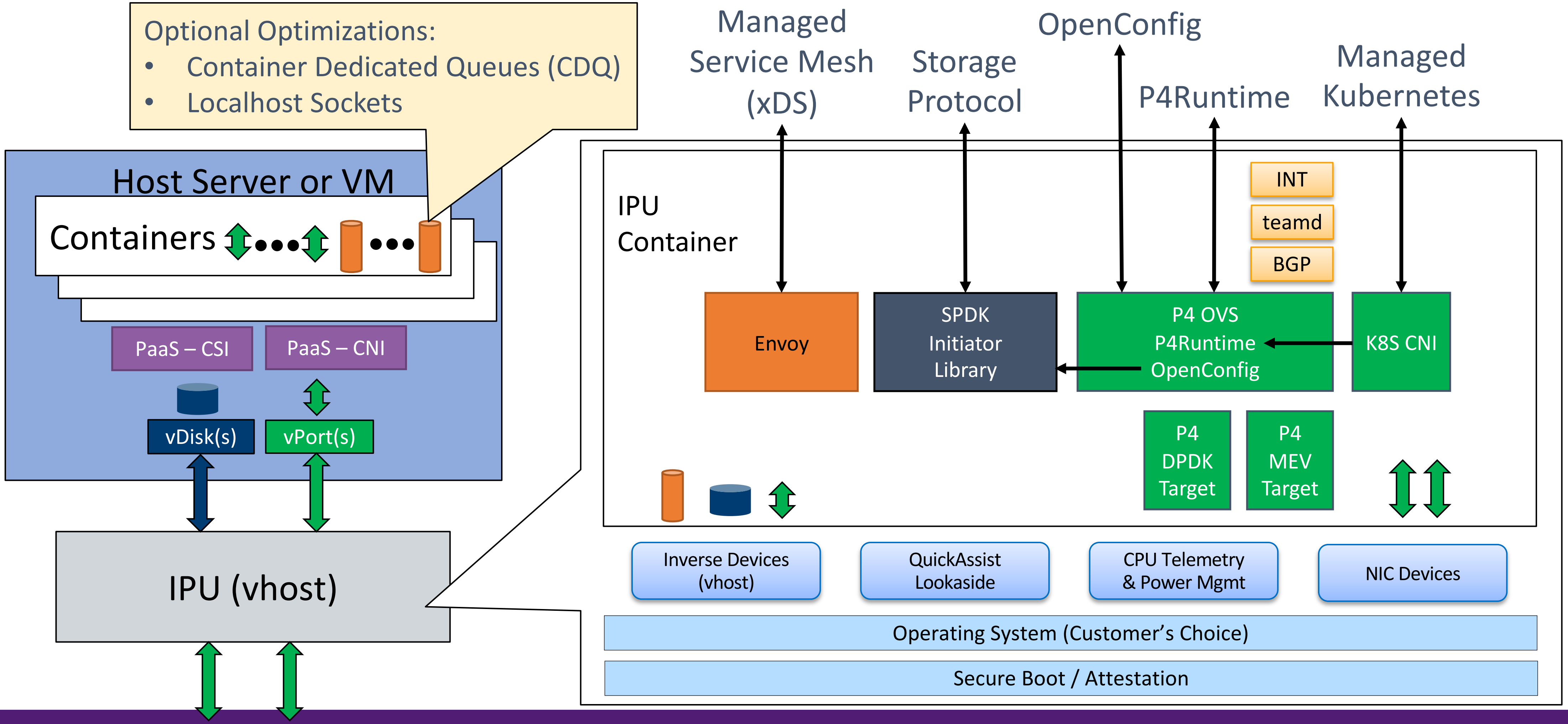


3 VM Loads Driver Traffic & Storage
Flexibility of Primary Control Point- Host or IPU
Storage Protocol
OpenConfig
1 Create Port/Disk **w/ VF #**
P4Runtime
Host Server
VM
IaaS – Host Virt. Agent
Live Migration
qemu+vDPA
vDisk(s)
vPort(s)
vDisk(s)
vPort(s)
IPU
IPU Container
IaaS – IPDK Plat. Manager
gRPC
SPDK Initiator Library
P4 OVS P4Runtime OpenConfig
INT
teamd
BGP
2 **Assign VF** assoc. netdev
P4 DPDK Target
P4 MEV Target
Inverse Device
QuickAssist
VPC Exceptions
CPU Telemetry & Power Mgmt
Network Control Plane Packets
Operating System (Customer's Choice)
Secure Boot / Attestation
4 VPC Dataplane

PaaS – CSI & CNI

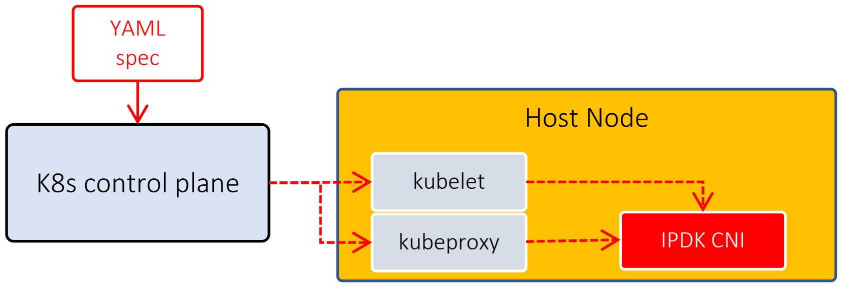
Optional Optimizations:

- Container Dedicated Queues (CDQ)
- Localhost Sockets

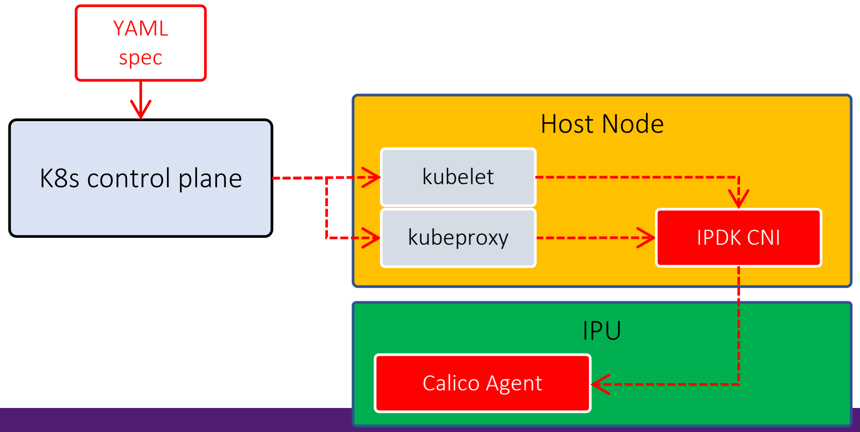


PaaS – K8s Delineations

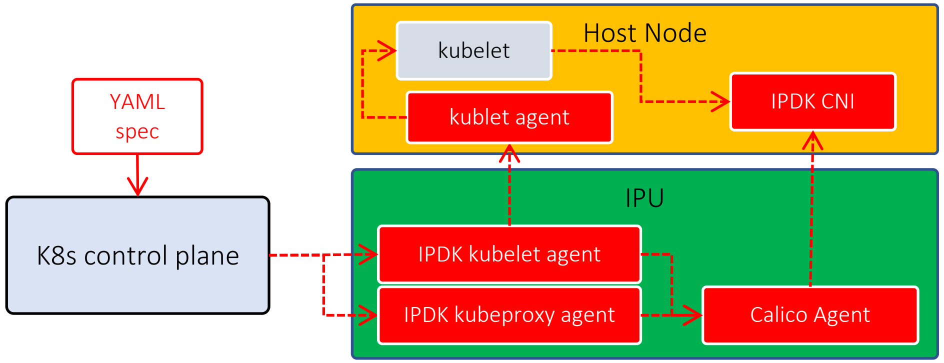
0. Bare Metal, no IPU



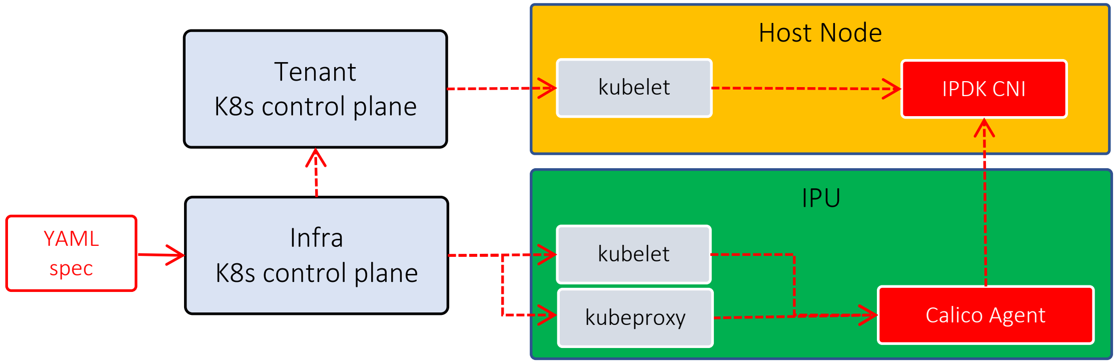
1. Trusted host anchored control (Self Managed)



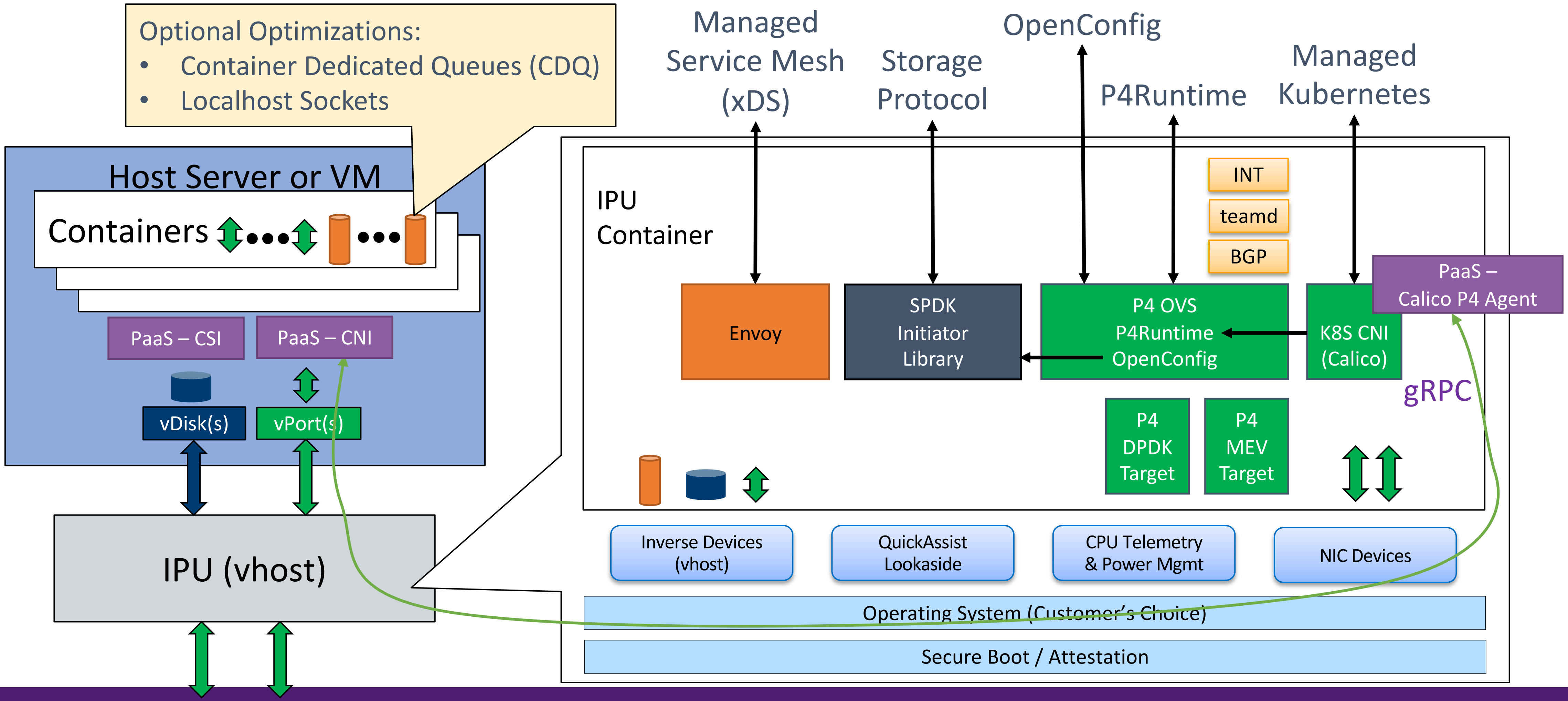
2. IPU anchored control (Managed Services)



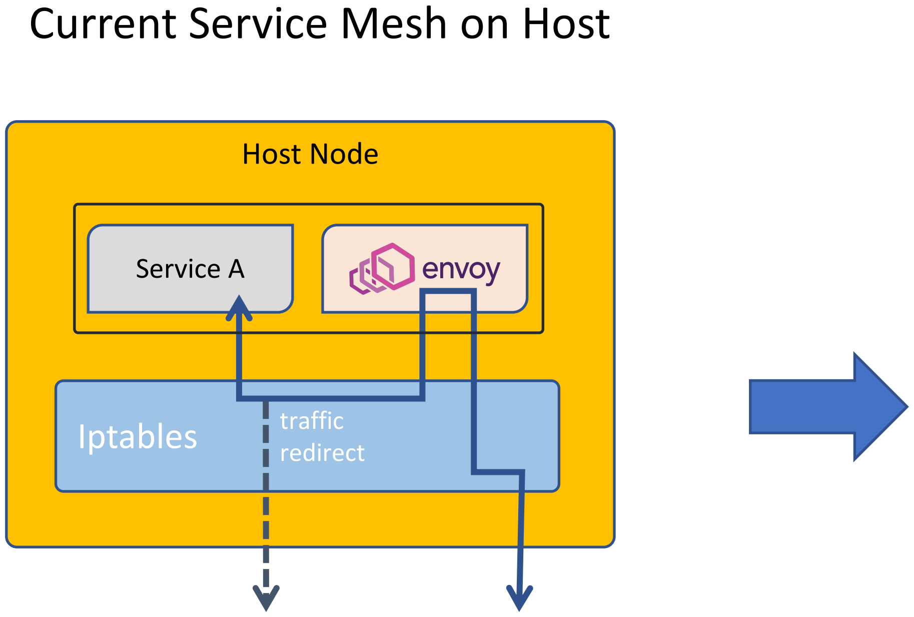
3. Split Tenant & Managed Infrastructure

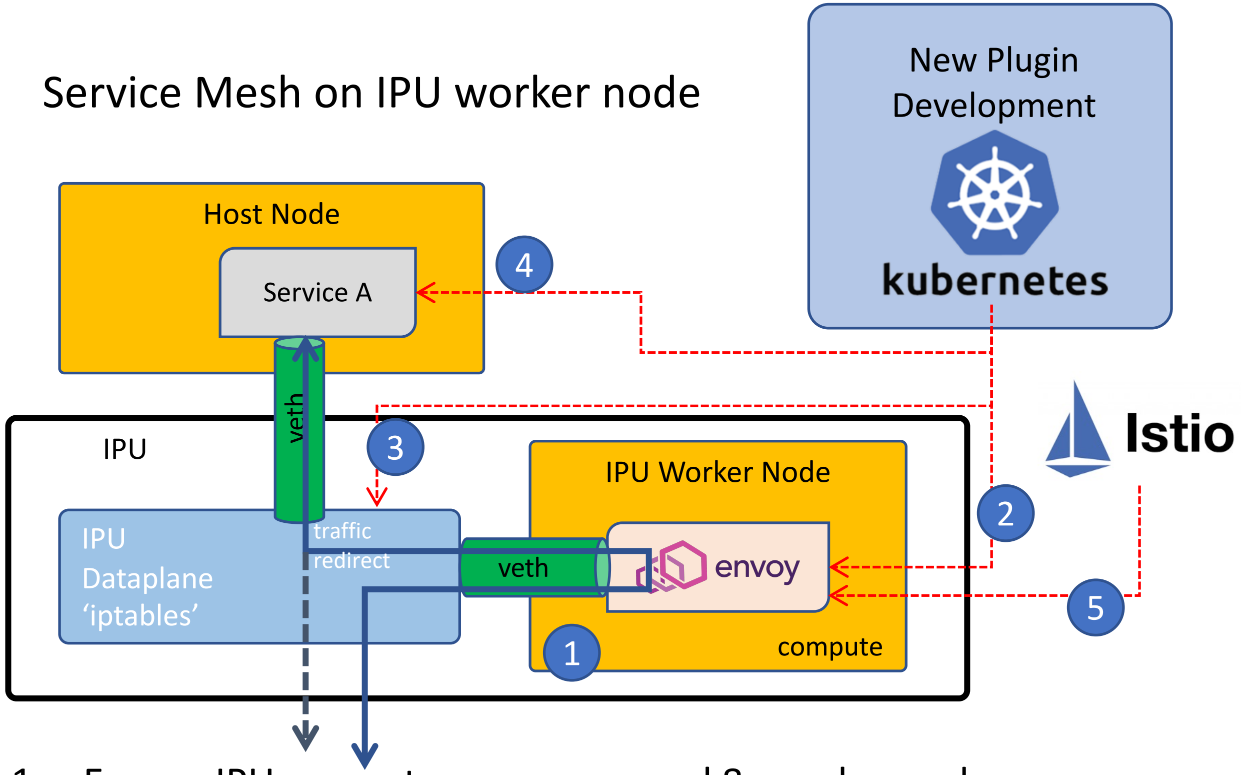


PaaS – Calico Running on IPU



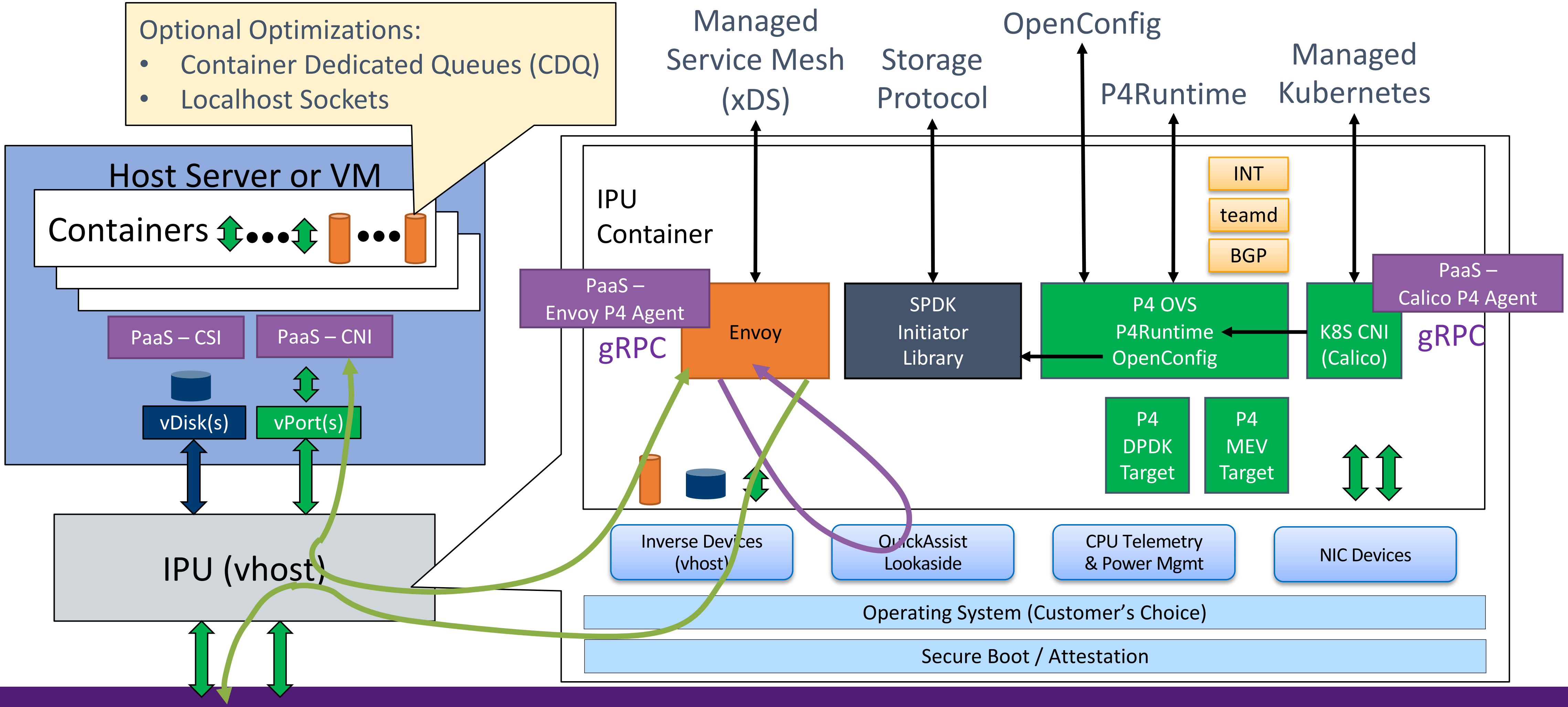
PaaS – Envoy Delineations





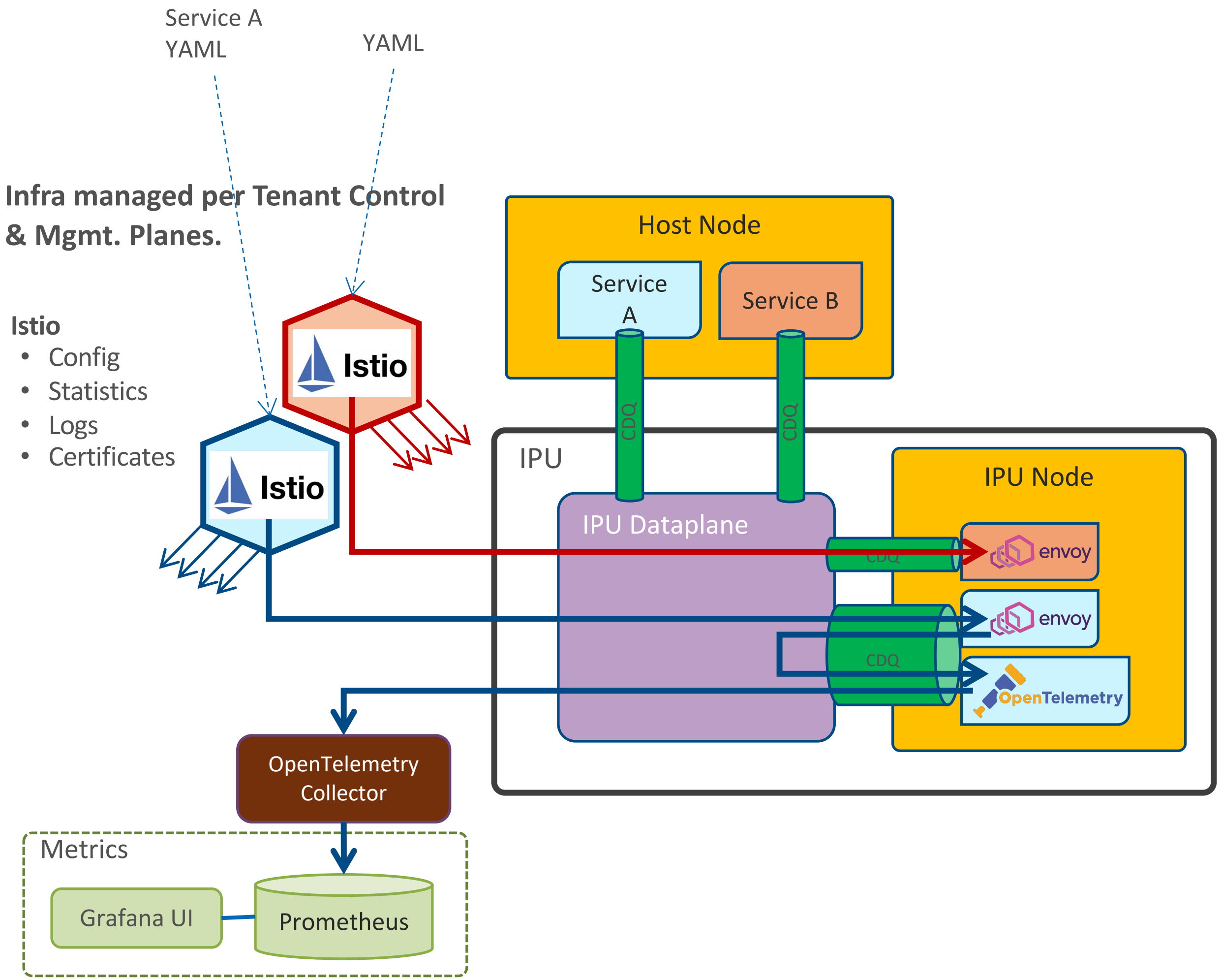
1. Expose IPU compute resources as a k8s worker node
2. Modify k8s orchestration to inject Envoy to IPU Worker Node
3. Configure IPU dataplane offload for 'iptables' traffic redirect
4. Orchestrate tenant service
5. Istio configuration of Envoy

PaaS – Envoy on IPU



PaaS – Multi-Tenancy (Service:Sidecar)

- IPU dataplane used for traffic redirection to Envoy proxy and offload of CNI
- 1:1 Service Mesh (Envoy) containers
  - Resource allocation & life cycle
  - CDQ enable scheduling & BW allocations
  - Separation of mTLS Certificates
- No restriction on Istio deployment
  - Support various deployments such as multiple Envoy instances, soft multi-tenancy, & RBAC namespaces.
- Additional services (OpenTelemetry) can be orchestrated
- Optional to use Istio soft multi-tenancy



Thank you.

IPDK.io: Infrastructure Programmer Development Kit
Collaborate with the community on Github & Slack