



# DATA PLANE AND INTEL<sup>®</sup> CLEAR CONTAINERS

Overview of networking POCs for Intel Clear Containers

[eric.ernst@intel.com](mailto:eric.ernst@intel.com), [manohar.r.castelino@intel.com](mailto:manohar.r.castelino@intel.com)

# Quick Introduction to Intel® Clear Containers

\*Brief\* look at some Dataplane Network Technologies:

- DPDK
- VPP\*
- SRIO-V

Using VPP, and SRIO-V with Intel Clear Containers

- Demonstration of SRIO-V
- Hands on Lab: playing with VPP and Clear Containers

# BRIEF OVERVIEW OF INTEL CLEAR CONTAINERS

```
samuel@zurbaran: [~]
```

```
$ docker pull ubuntu
```

```
Using default tag: latest
```

```
latest: Pulling from library/ubuntu
```

```
b6f8920042b Pull complete
```

```
55010f Pull complete  
2955fb Pull complete
```

```
3deef3fcbd30: Pull complete
```

```
cf9722e506aa: Pull complete
```

```
Digest: sha256:382452f82a8bbd34443b2c727650af46aced0f94a44463c62a9
```

```
848133ecb1aa8
```

```
Status: Downloaded newer image for ubuntu:latest
```

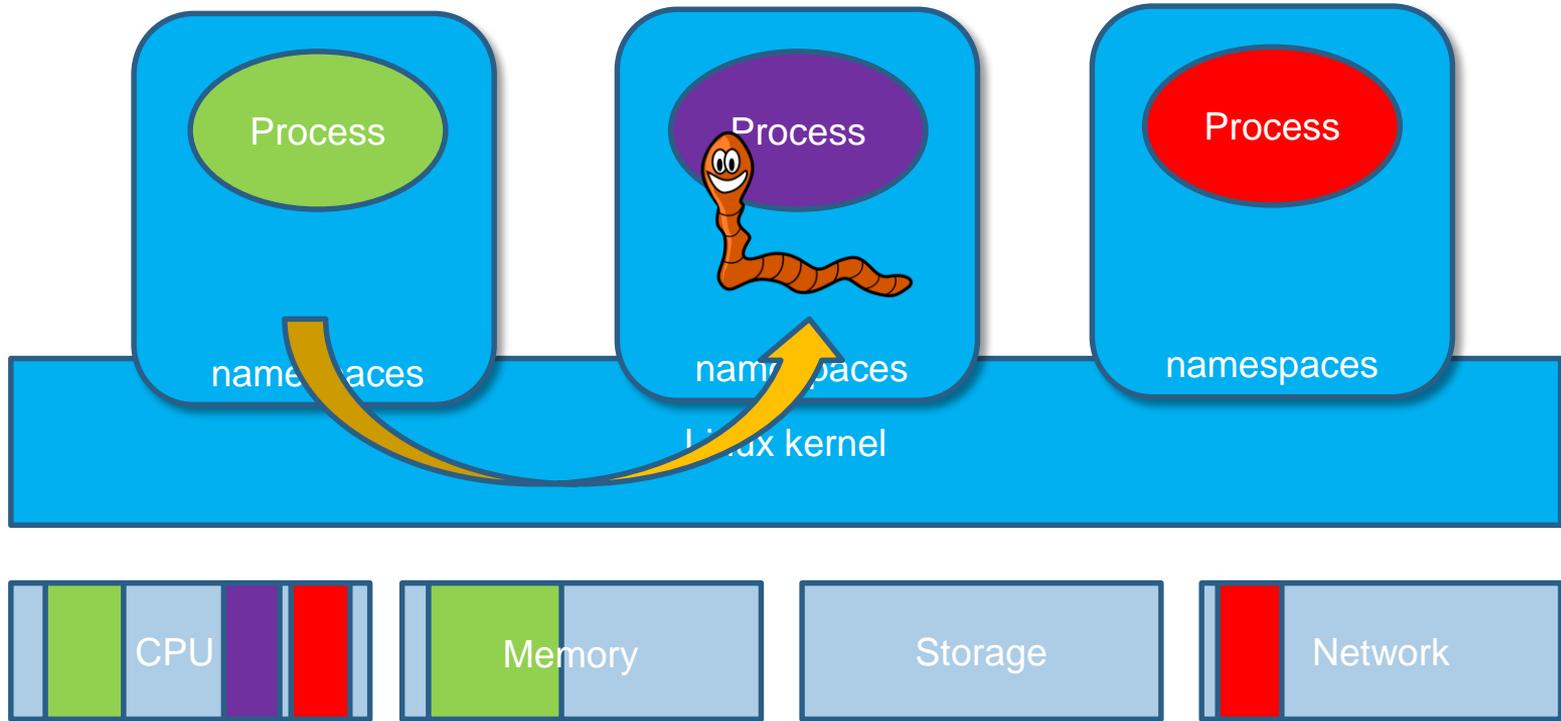
```
samuel@zurbaran: [~]
```

```
$ docker run -ti ubuntu
```

```
root@b3161c669676:/#
```

```
docker run (less than 1 sec)
```

# Linux\* namespace containers





# HARDWARE SECURITY

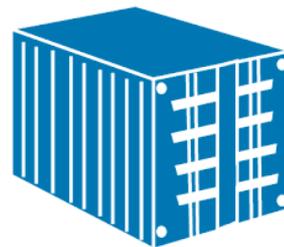
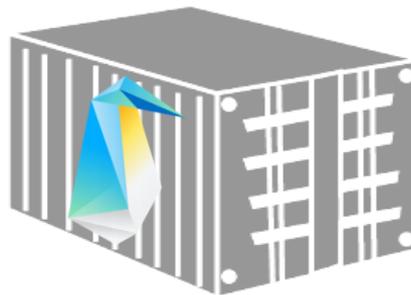


# SOFTWARE SECURITY

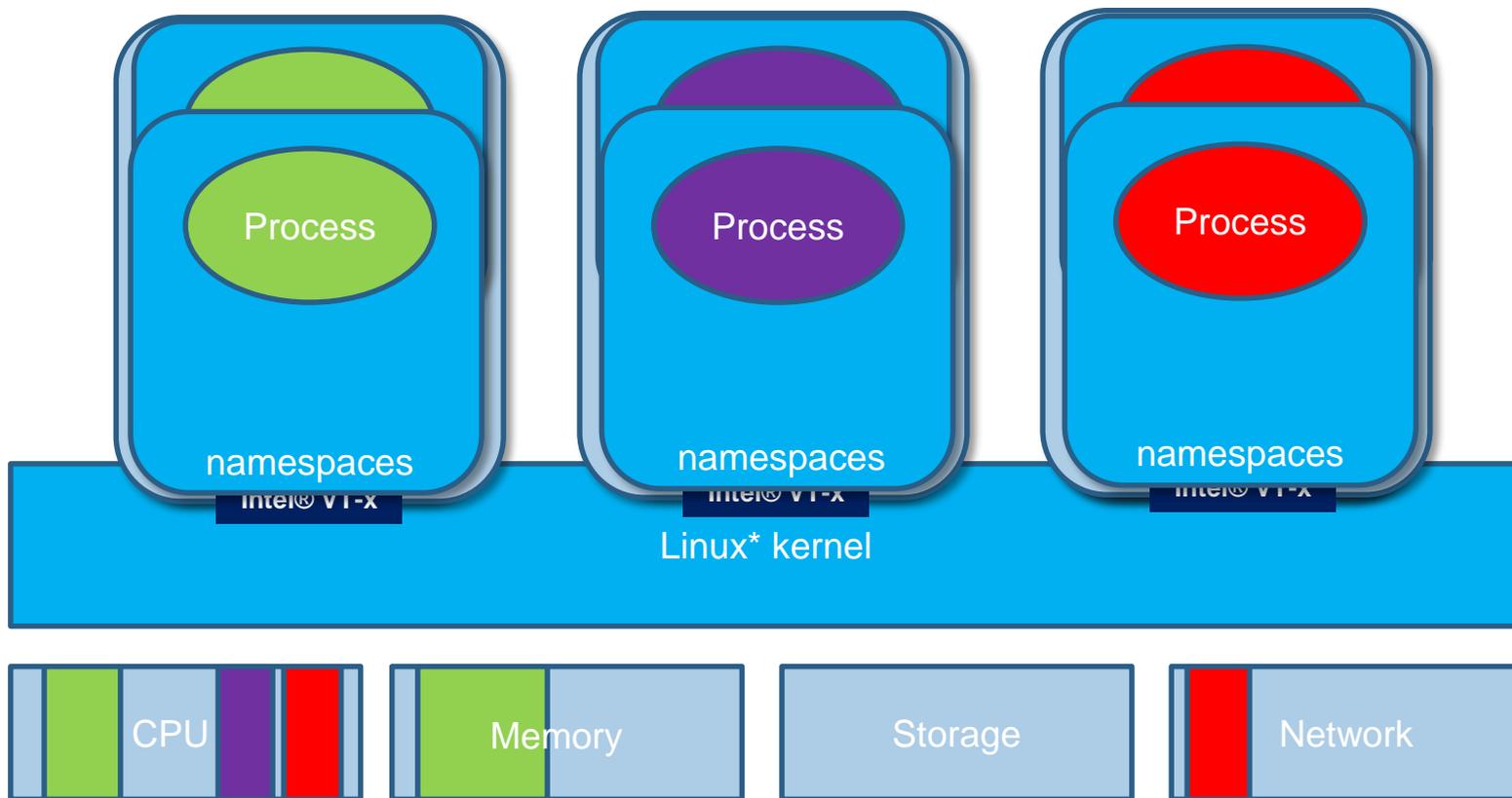
Security



Virtual Machines



Speed



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

# INTERESTING, SO WHAT?

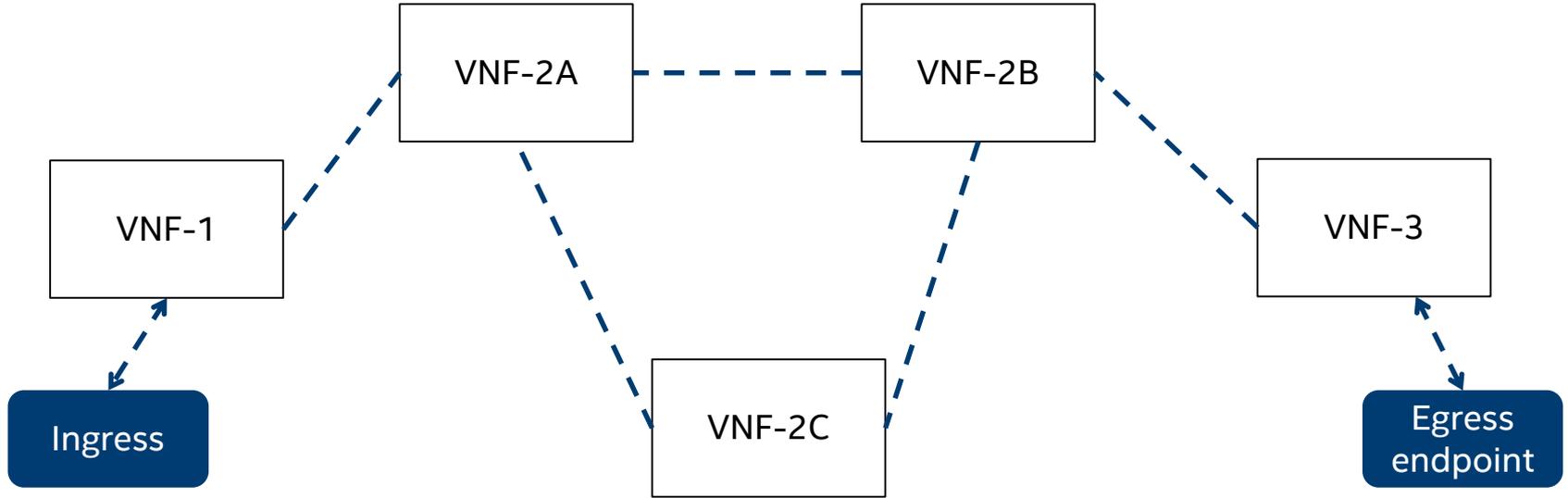
Telecommunications and their applicability to Intel Clear Containers

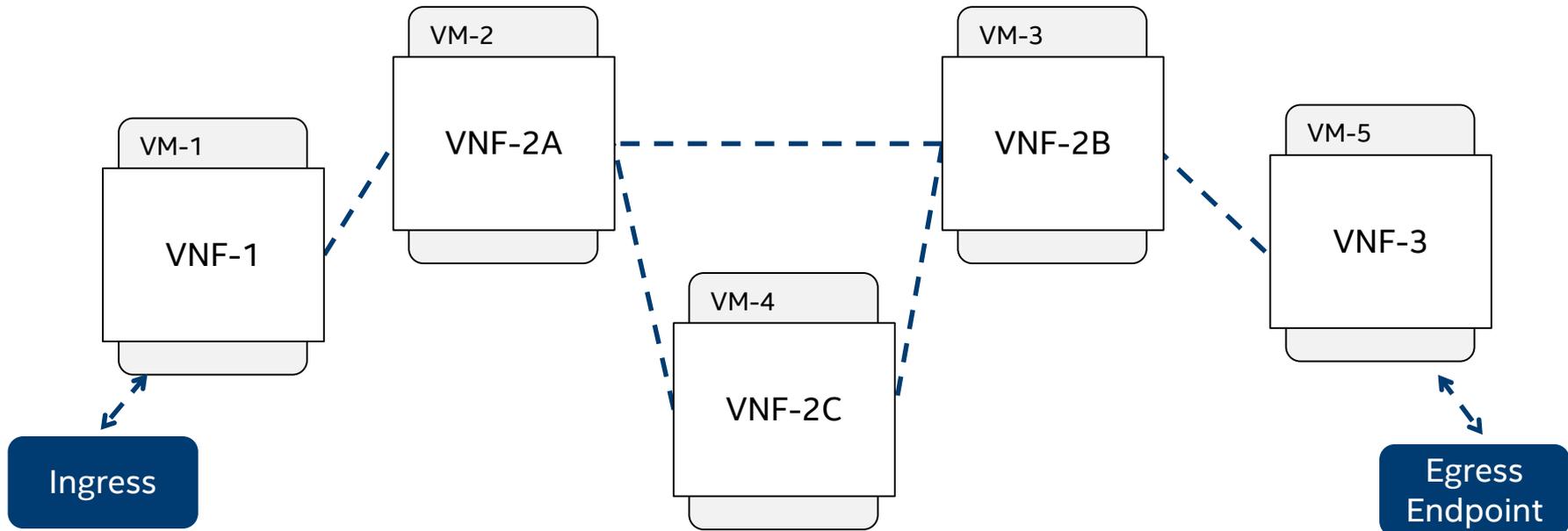
# Historically a strong place for virtual machines (VMs)

Telco usage and requirements tends to favor VMs over containers, at least for the time being.

- Reliability: five 9s:
  - Computing infrastructure working 99.999% of the time.
- Highly customized, optimizations in their workload's stack, including in kernel.
- Use cases aren't necessarily throw-away, short-lived workloads in the cloud; lower downtime.

Intel Clear Containers provides a unique feature set relative to standard containers.





# WHY WE CANNOT JUST USE LINUX\* NETWORKING

Motivation for data plane acceleration

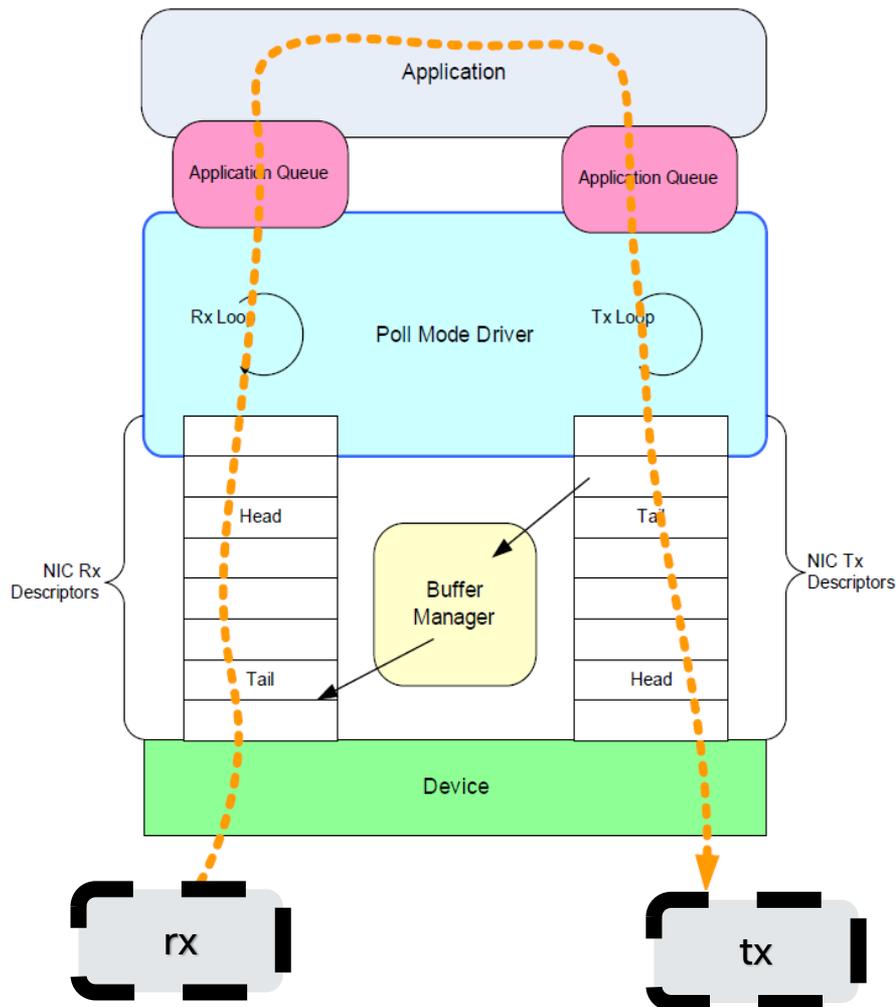
# Time budget for processing

	64 B packet
1 GB	672 ns
10 GB	67.2 ns
40 GB	16.8 ns
100 GB	6.72 ns

- Process one packet:
  - 67.2 ns -- ~201 cycles @ 3 GHz
- Cache:
  - L2 / L3 access: 12 cycles / 44 cycles
  - Cache hit in other core: 60 cycles
  - Memory: ~70 ns: 210 cycles!

# DPDK

Data Plane Development Kit



- Is a framework for fast packet processing.
- Put/get directly to from DMA area to avoid copies using huge pages.
- Works in Poll Mode (no interrupt context switch/latency). Directly poll data from the NIC.
- Uses huge pages to avoid dealing with paging.

# VPP\*

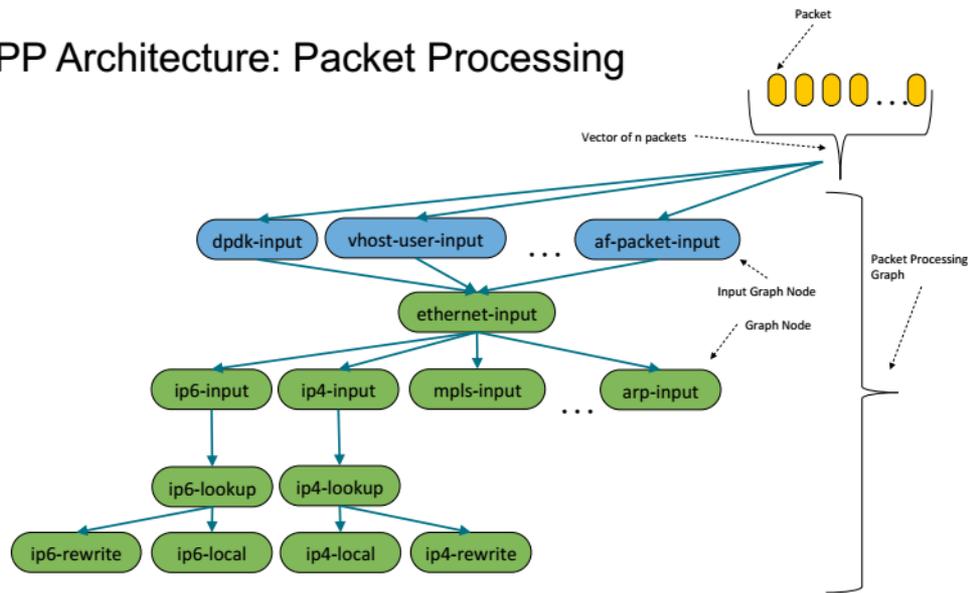
Vector Packet Processing Platform



# VPP\* quick overview

- Built on a packet processing graph.
- Modular: It is a framework to which you can add your own graph nodes.
- Takes in a vector of packets at a time to amortize the cost of a D-cache miss and mitigates read latency. Nodes optimized to keep L-cache hot.
- All user space - makes use of DPDK.

VPP Architecture: Packet Processing



# VPP\*: Learn more

- <https://wiki.fd.io/view/VPP>
- Overview, including architecture and performance:
  - <http://events.linuxfoundation.org/sites/events/files/slides/fdio%20-%20Universal%20Dataplane.pdf>
- CNM plugin for Intel Clear Containers:
  - <https://github.com/clearcontainers/vpp>



# SRIO-V

Single Root I/O Virtualization

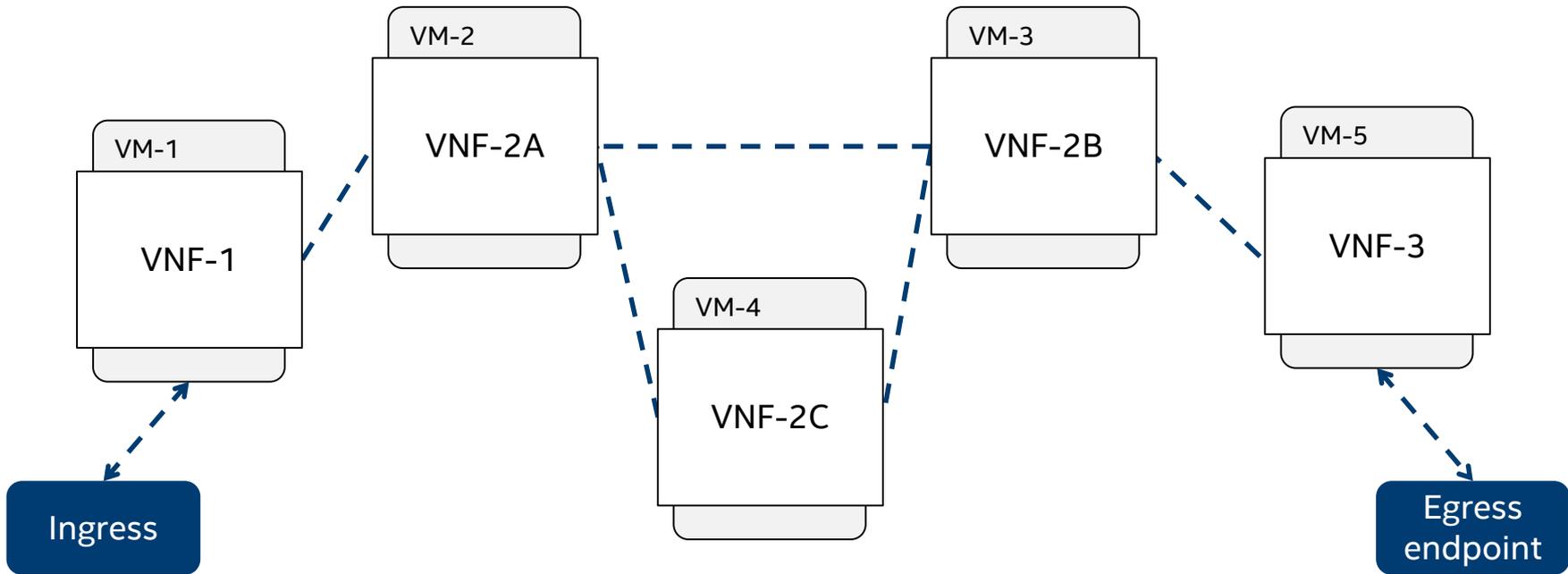
# SRIO-V

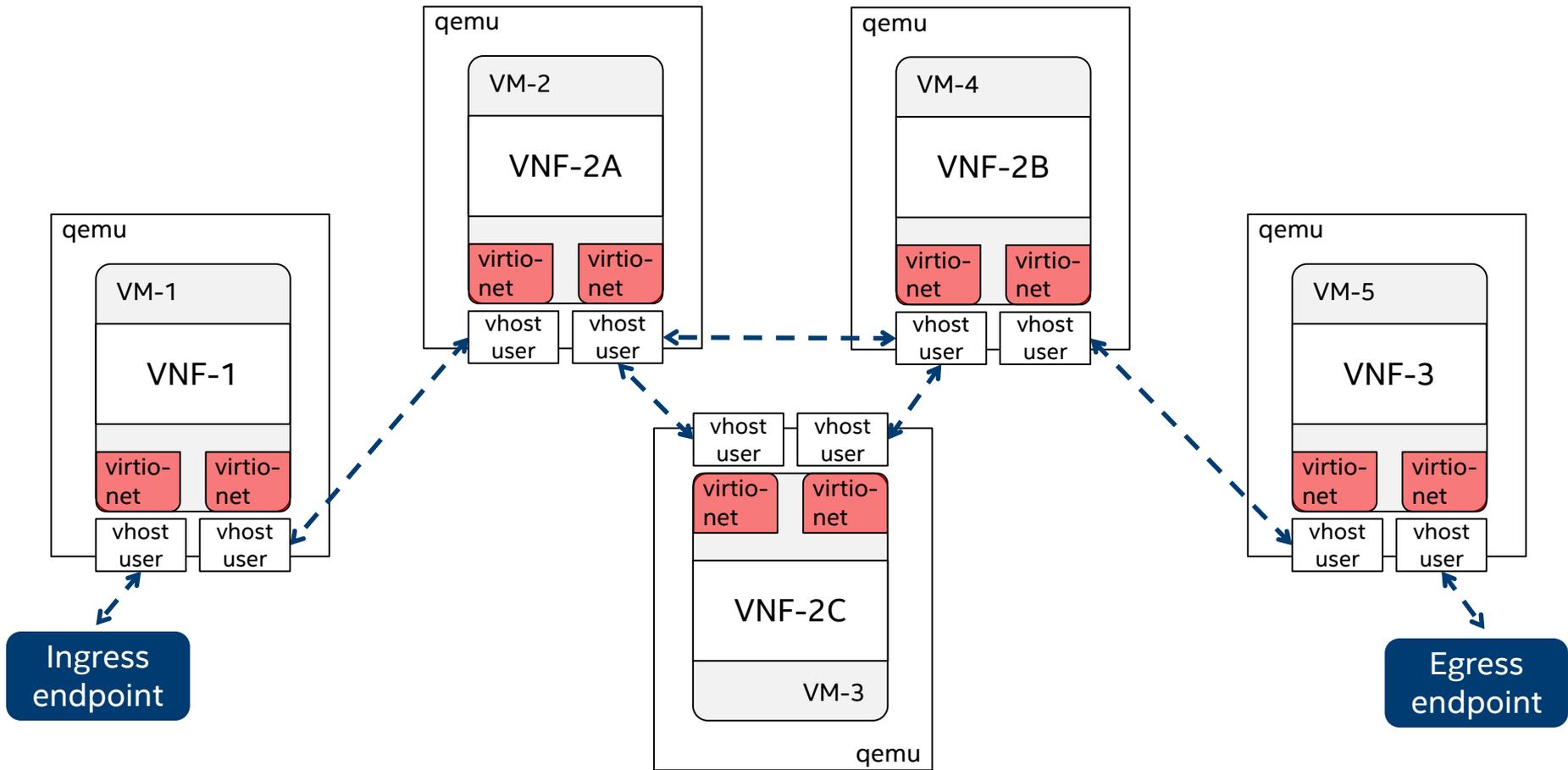
- Extension of PCI specification.
- Allows a device to separate its resources among various PCIe hardware functions.
  - Physical function: This is the primary function of the device, which also exposes the SRIO-V capability to the platform.
  - Virtual function: Once enabled, this shows up on the host with a BDF, and depending on the devices hardware design, can operate in its own IOMMU group.
- With SRIO-V, you can pass a physical connection directly to a VM, allowing native performance (no virtualization overhead).

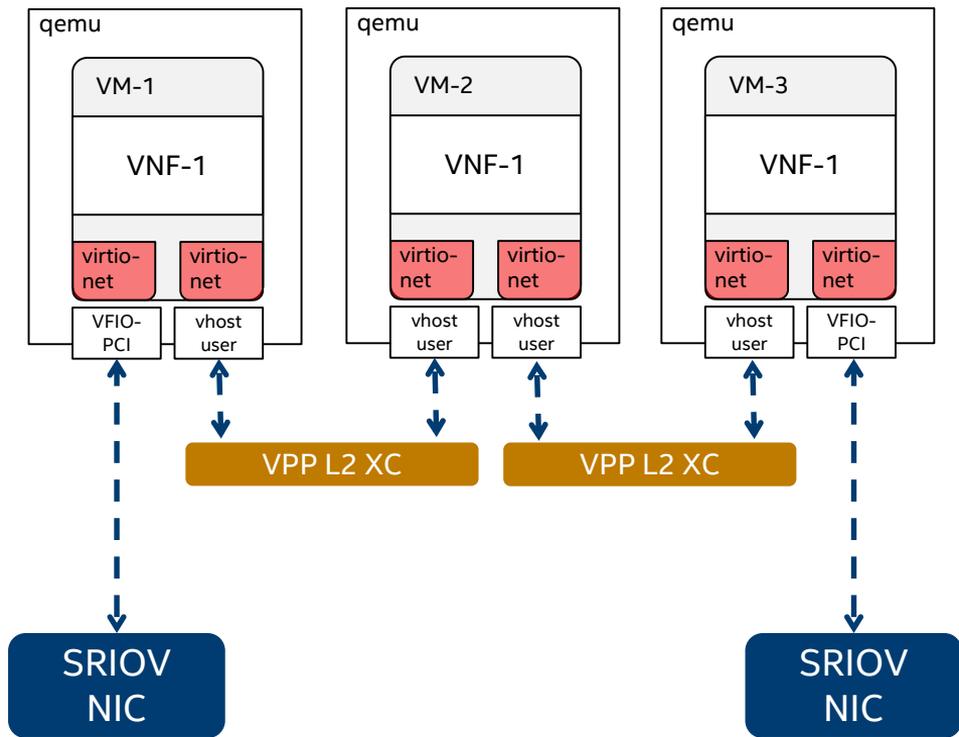
# SRIO-V: Learn more

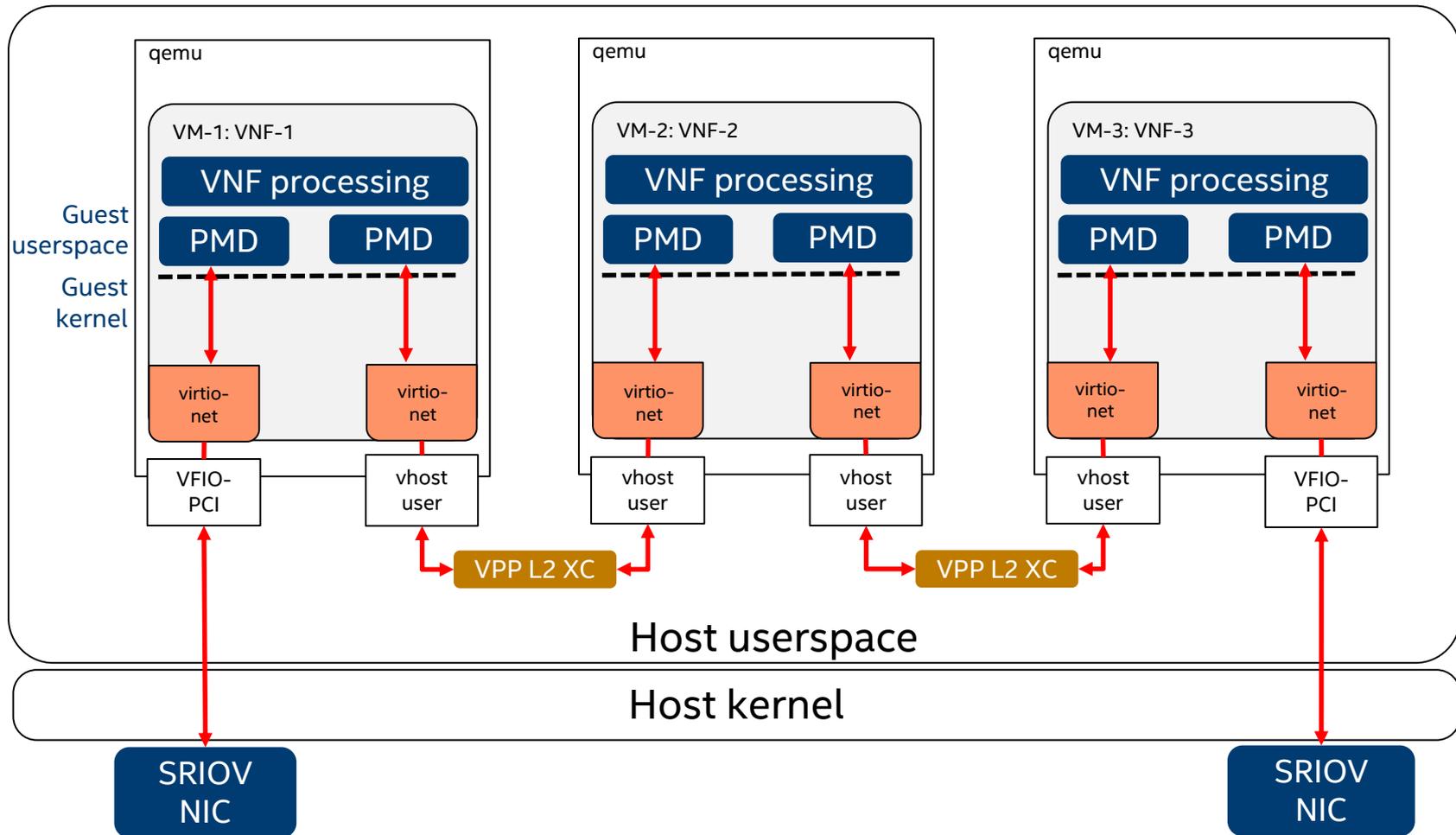
- Overview of SRIO-V:
  - <https://www.intel.sg/content/dam/doc/application-note/pci-sig-sr-iov-primer-sr-iov-technology-paper.pdf>
- CNM plugin for Intel Clear Containers:
  - <https://github.com/clearcontainers/sriov>
- Example of using SRIO-V with Intel Clear Containers:
  - <https://gist.github.com/egernst/b0fd8ec4d9bd5cc323d2d00a0ad1f81c>

# PUTTING IT ALL TOGETHER









# SUPPORT IN INTEL CLEAR CONTAINERS

# CNM based plugins

- **OVS-DPDK:** `$ go get github.com/clearcontainers/ovsdpdk`
  - The plugin creates an OVS bridge on the host system per Docker\* network creation.
  - Each container added to this network is provided a vhost-user interface. Intel Clear Containers search for this vhost-user socket to identify if special network setup is required.
- **VPP\*:** `$ go get github.com/clearcontainers/vpp`
  - Plugin will add endpoints for the same VPP-based network to the same L2 bridge domain. Intel Clear Containers search for this vhost-user socket to identify if special network setup is required.
- **SRIO-V:** `$ go get github.com/clearcontainers/sriov`
  - Plugin assumes that VFs are already created on the host and places the VF into the container namespace (this works with *runc* out of the box).

**DEMONSTRATION**



# Disclaimers

Intel and the Intel logo are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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

© Intel Corporation