



Leveraging IO Visor for NFV Accelerations

Yunsong.Lu@Huawei.com

Networking & Virtualization

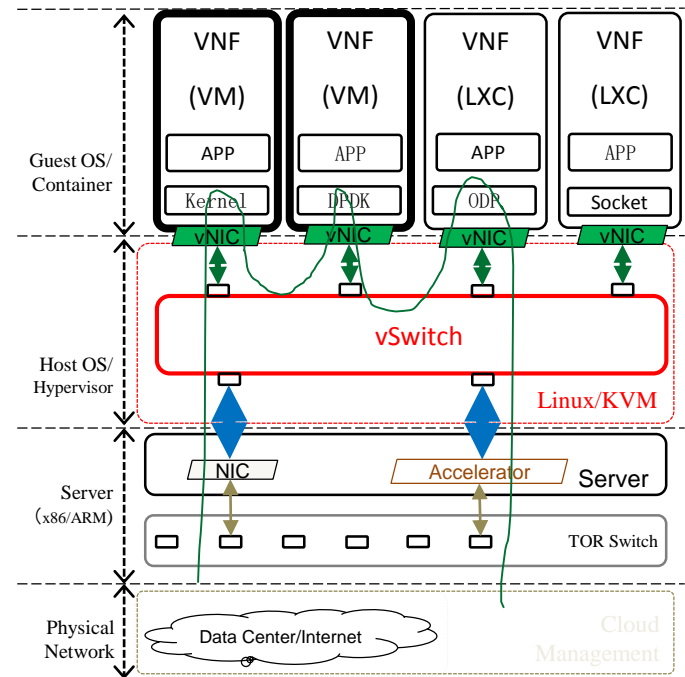
Early Stage of NFV: Porting existing implementations to VMs

NFV ARCHITECTURE 1.0 (0.9?)

VNFs Connected to vSwitch

Functionality

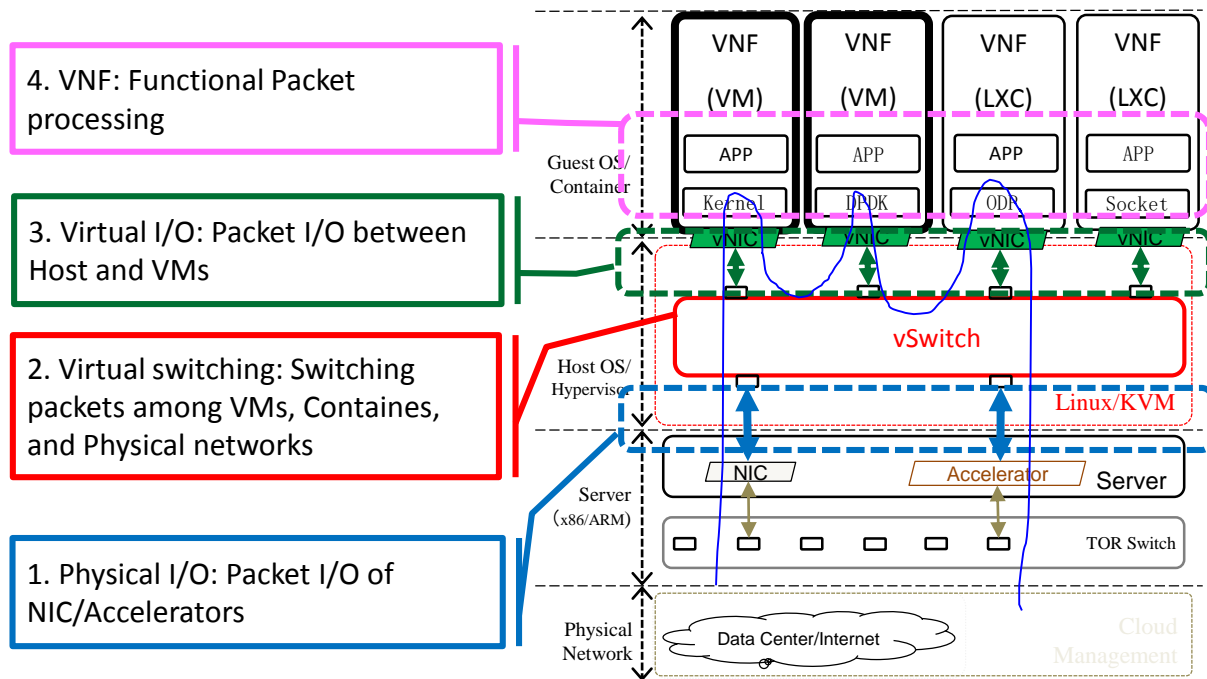
1. Virtual Switch is fundamental for virtual networking
2. SR-IOV is not ready for NFV without hardware acceleration
3. Many functions are written in VM/LXC as application today
4. Functions can be moved to Host Kernel with IO Visor



Performance

1. Multiple I/O operations for forwarding each packet
2. Can't simply bypass layers to compromise functionality and flexibility
3. Lightweight Virtualization technologies can't solve all performance issues
4. Acceleration can be achieved only by "bypass", real hardware acceleration is needed

Performance Pain Points



- Making use of Virtualization, IT technology, is with a price
- Performance can be improved with “acceleration” technologies
- NFV is to eliminate hardware specific implementation in VNF
- Or, Rethink about NFV Data Plane Design

NFV Data Plane Acceleration

Software Acceleration (bypass)

- Kind of “bypass” technologies
- Optimizations to existing components or modules
- Compromises between performance and openness(compatibility, eco system, and)

Rethink about bypass technologies:

- Linux Kernel: Driver, Scheduling, Functionalities

or

- Userland SDK: DPDK vs. ODP

- ✓ IO Visor may be the sharable “IO Engine” as agnostic “VNF Data Plane Container”

Hardware Acceleration

- Leveraging mature hardware designs: Offload workload to hardware
- May lead to vendor lock-in if without a community-driven framework: DPACC! (and IO Visor 😊)
- NFV-Oriented Software-Hardware Co-Design

- IHV’s SDK: xDK, yDK, zDK...

or

- Cross-Platform Open Framework and APIs

- ✓ IO Visor can be an option of platform-independent “VNF Data Plane Container”

Rethink about Network Acceleration

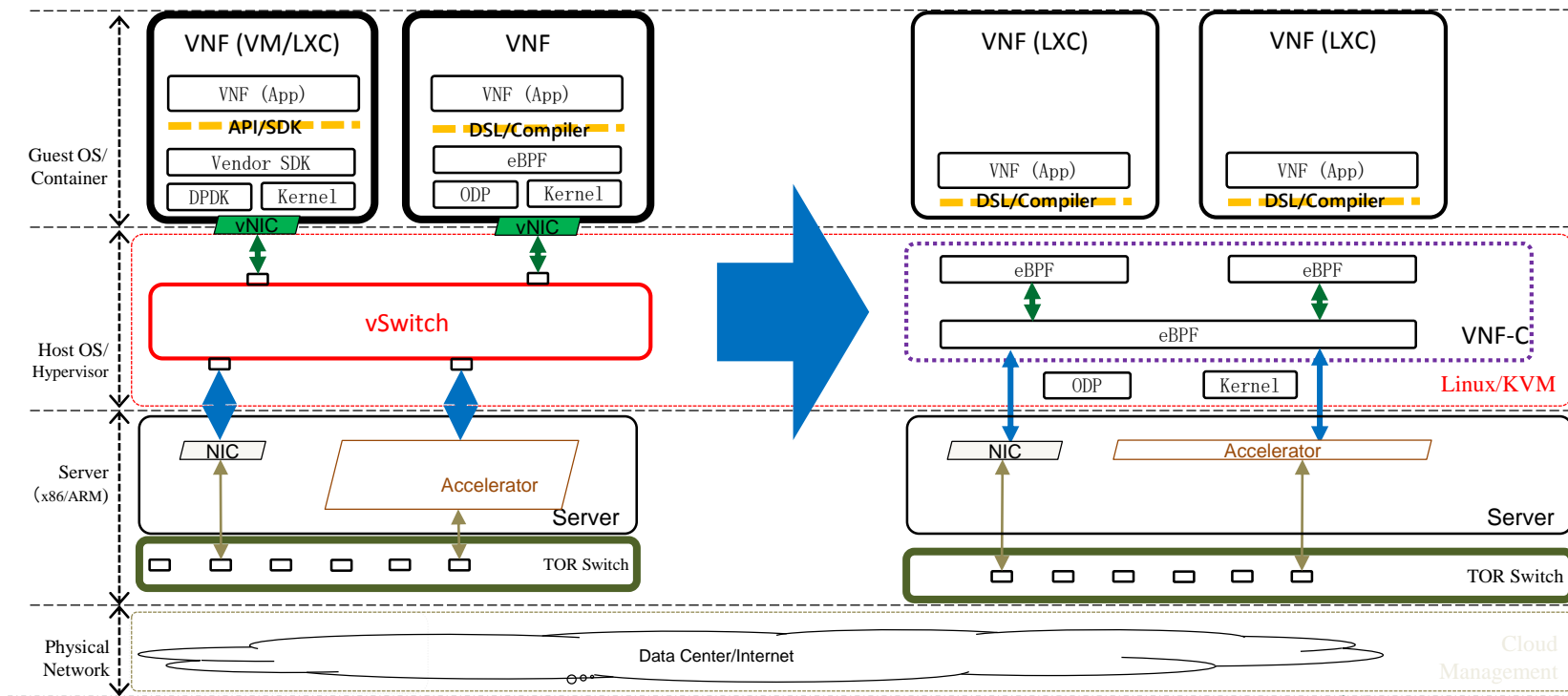
1. To consolidate data planes to reduce “layers” instead of bypassing functionalities
2. Enable Acceleration with a full programmable IO Engine in a multi-vendor eco-system
3. Open all possibilities with open API of acceleration engines

Indeed 3 steps to experiment>

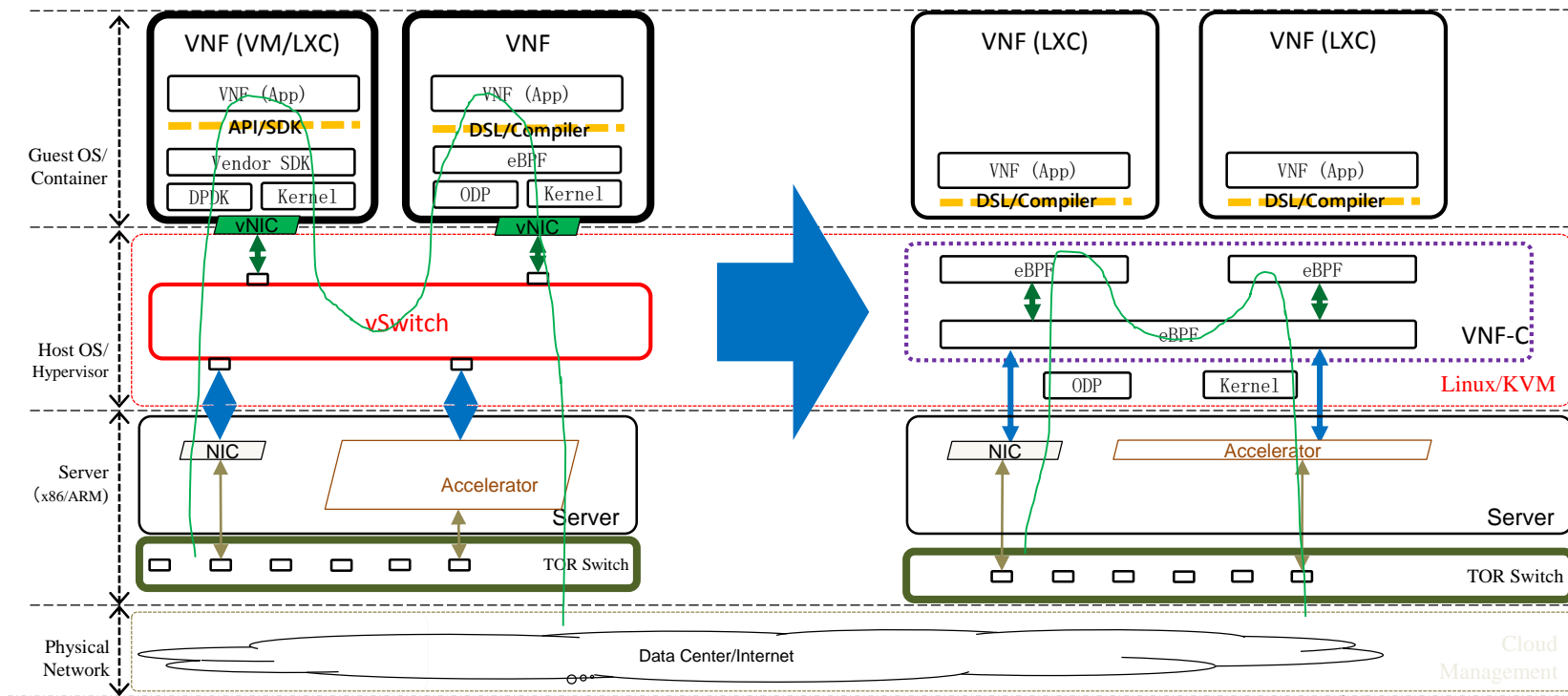
Step 1: Merge vSwitch and VNFs into one data plane

DATA PLANE EVOLUTION (W/O ACCELERATOR)

Data Plane Evolution (w/o Accelerator)



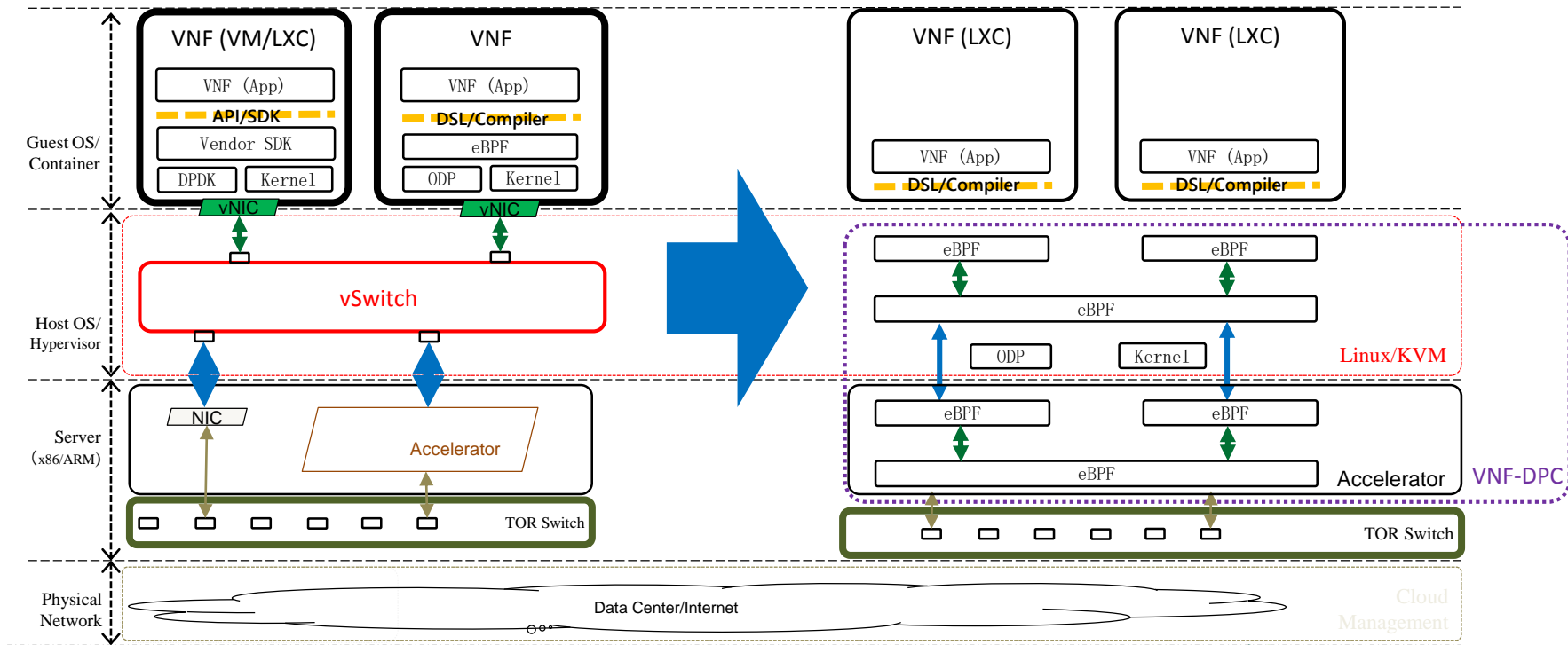
Data Plane Evolution (w/o Accelerator)



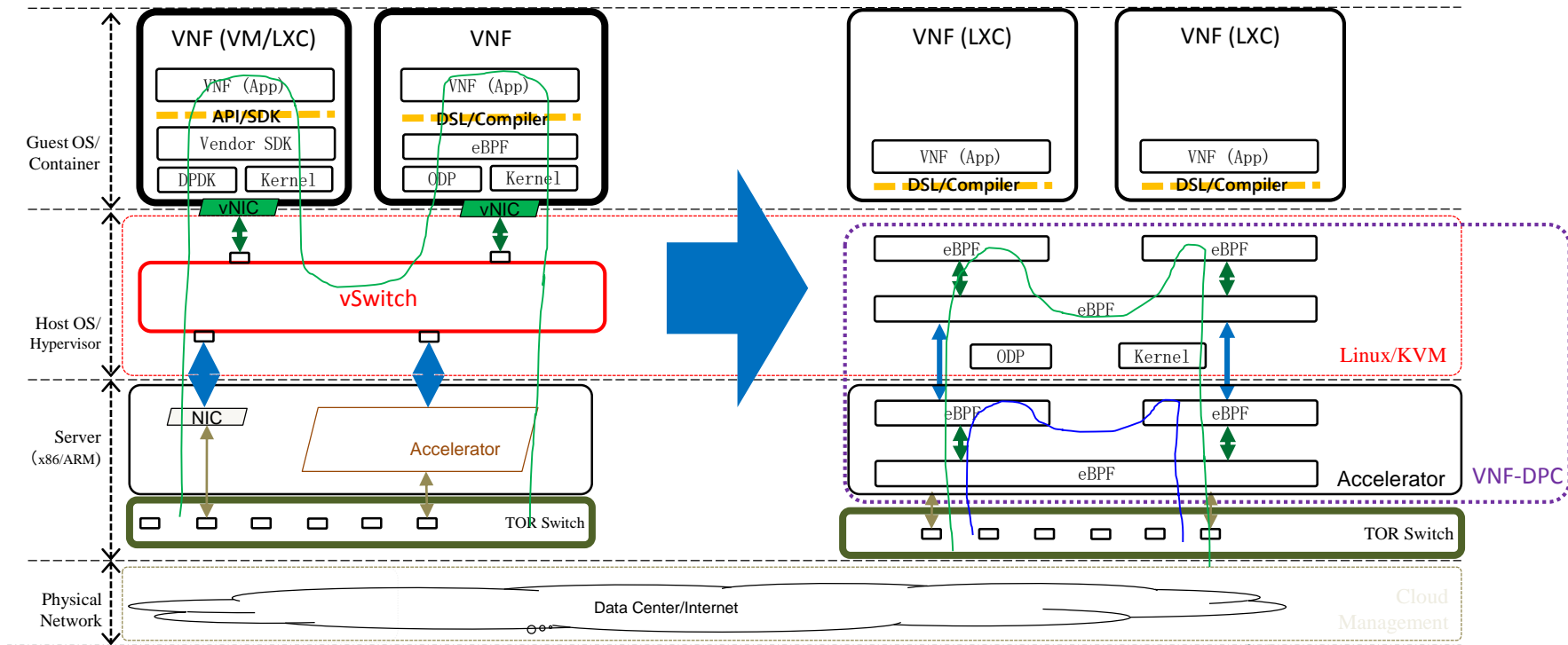
Step 2: Introduce IO Visor to Hardware Accelerators

DATA PLANE EVOLUTION (W/ ACCELERATOR)

Data Plane Evolution (w/ Acceleration)



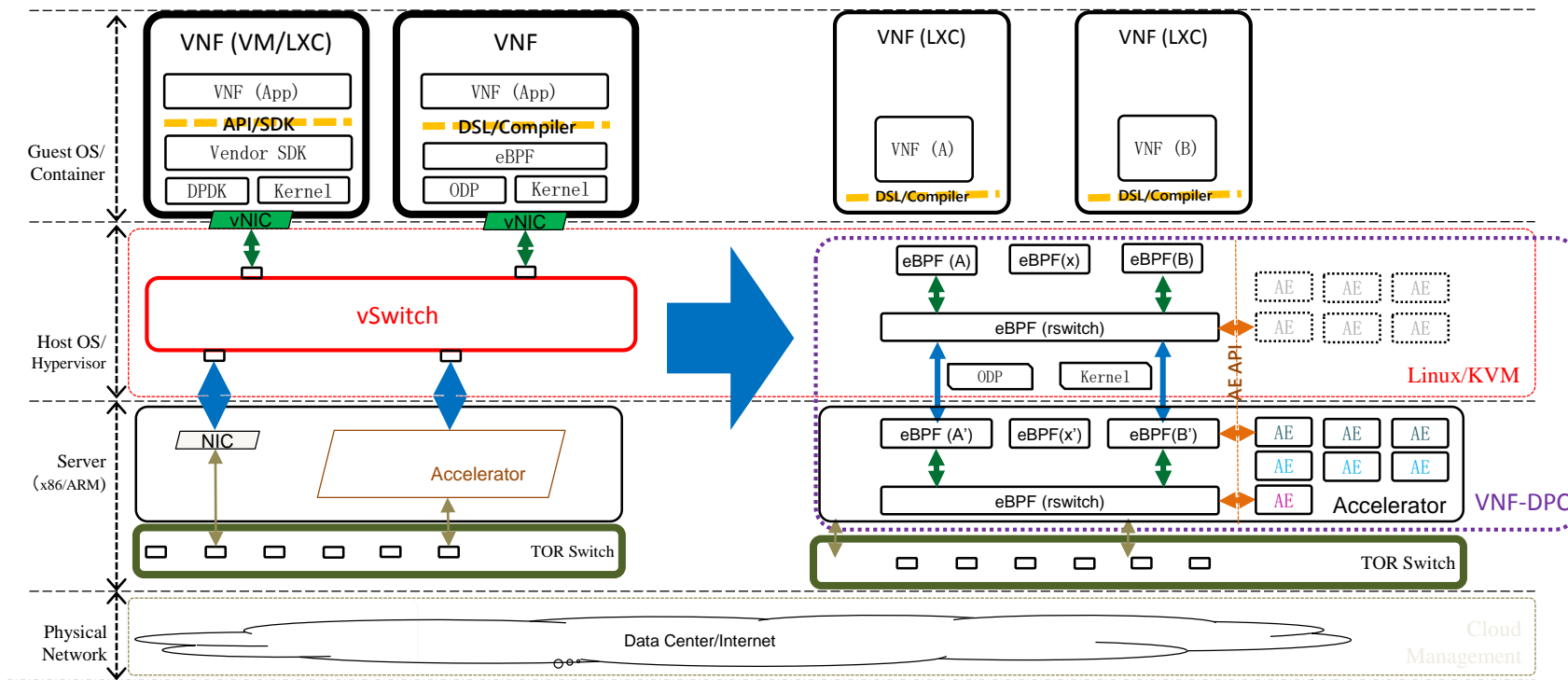
Data Plane Evolution (w/ Acceleration)



Step 3: Leverage Accelerators Engines for better performance

DATA PLANE EVOLUTION (W/ ACCELERATOR)

Data Plane Evolution (w/ Acceleration Engines)



Thank you!