

CICN Community Information-Centric Networking



FD.io: The Universal Dataplane



- Project at Linux Foundation

- Multi-party
- Multi-project

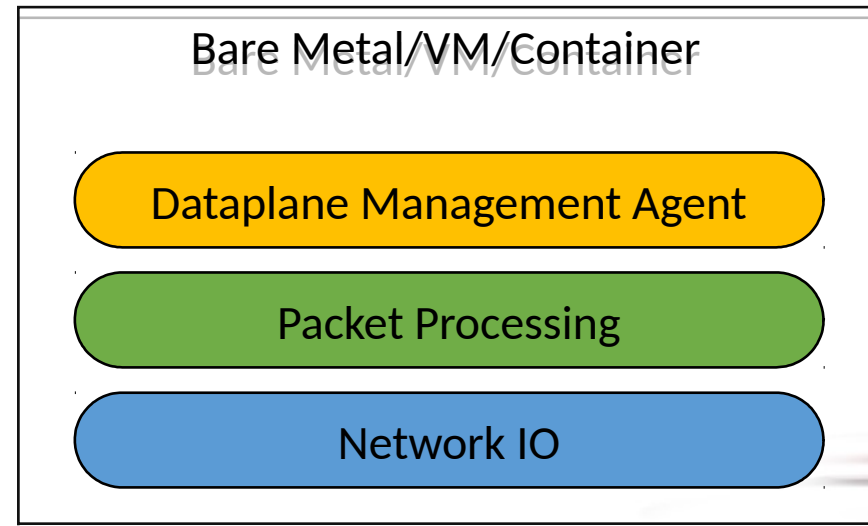
- Software Dataplane

- High throughput
- Low Latency
- Feature Rich
- Resource Efficient
- Bare Metal/VM/Container
- Multiplatform

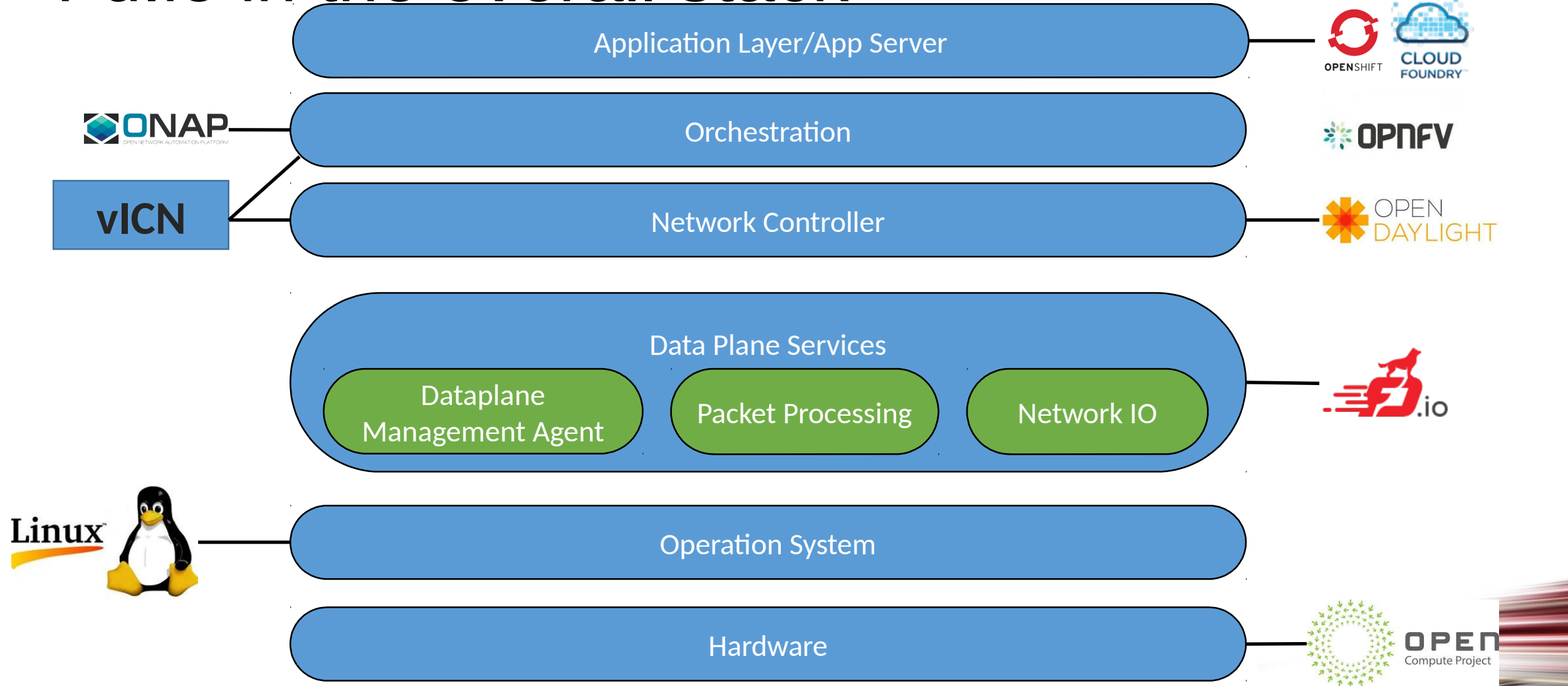


- Fd.io Scope:

- **Network IO** - NIC/vNIC <-> cores/threads
- **Packet Processing** -
Classify/Transform/Prioritize/Forward/Terminate
- **Dataplane Management Agents** - ControlPlane



Fd.io in the overall stack



Multiparty: Broad Membership

Service Providers



Network Vendors



Chip Vendors



Integrators



Multiparty: Broad Contribution



Qiniu

HUAWEI



KALRAY



ZTE



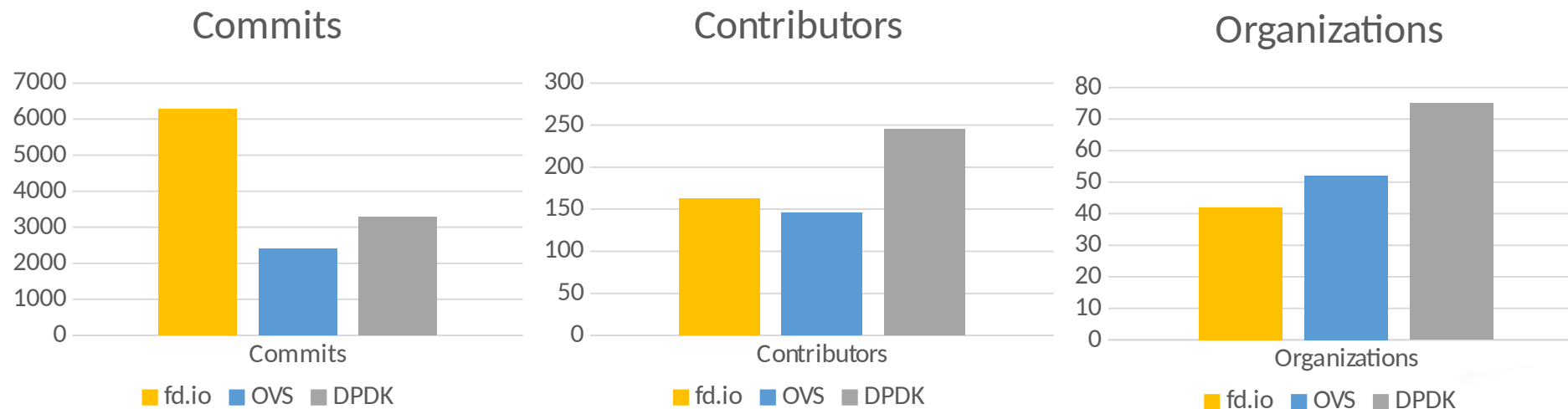
Universitat Politècnica de Catalunya (UPC)



Code Activity

- In the period since its inception, fd.io has more commits than OVS and DPDK combined, and more contributors than OVS

2016-02-11 to 2017-04-03	Fd.io	OVS	DPDK
Commits	6283	2395	3289
Contributors	163	146	245
Organizations	42	52	78



Multiproject: Fd.io Projects

Dataplane Management Agent

vICN

hc2vpp

Honeycomb

Testing/Support

CSIT

puppet-fdio

trex

Packet Processing

ICNET

ONE

TLDK

CICN

odp4vpp

VPP Sandbox

VPP

Network IO

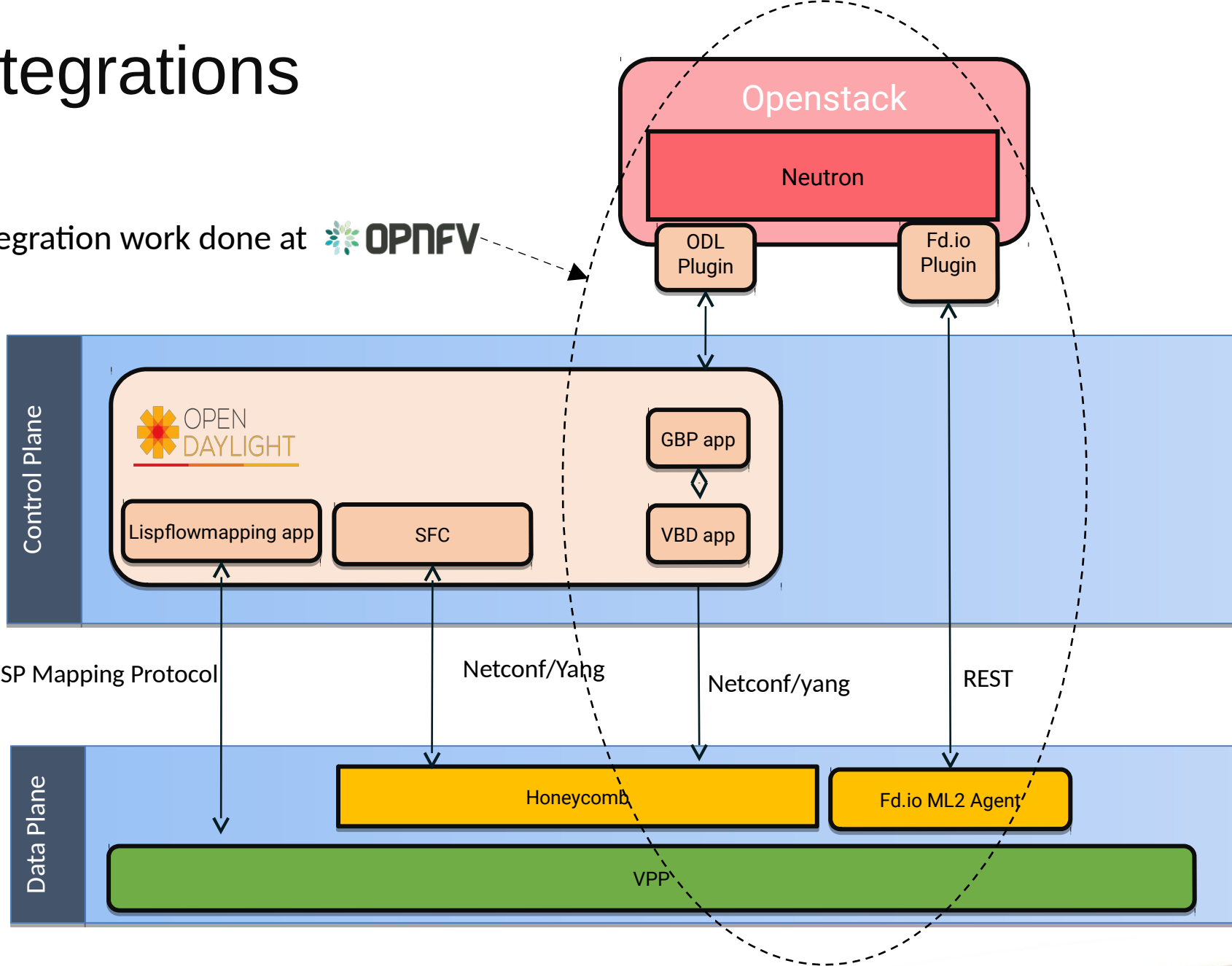
deb_dpdk

rpm_dpdk

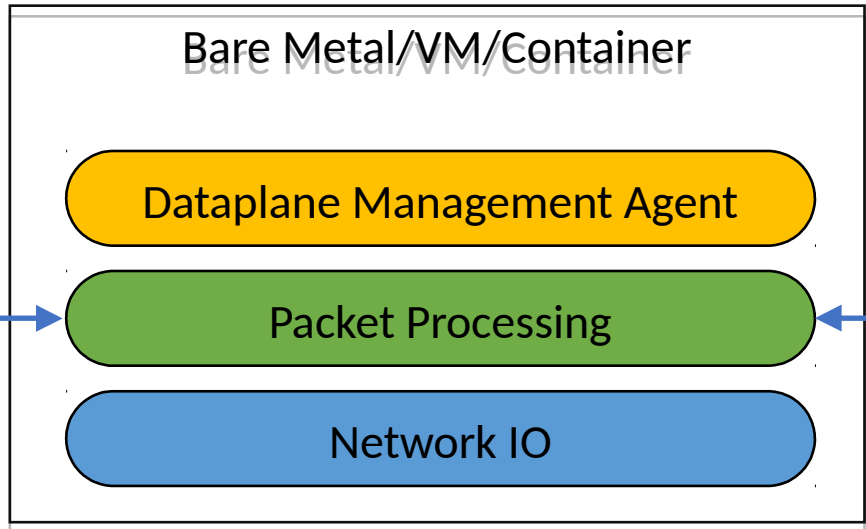
Fd.io Integrations



Integration work done at  **OPNFV**



Vector Packet Processor - VPP



- Packet Processing Platform:

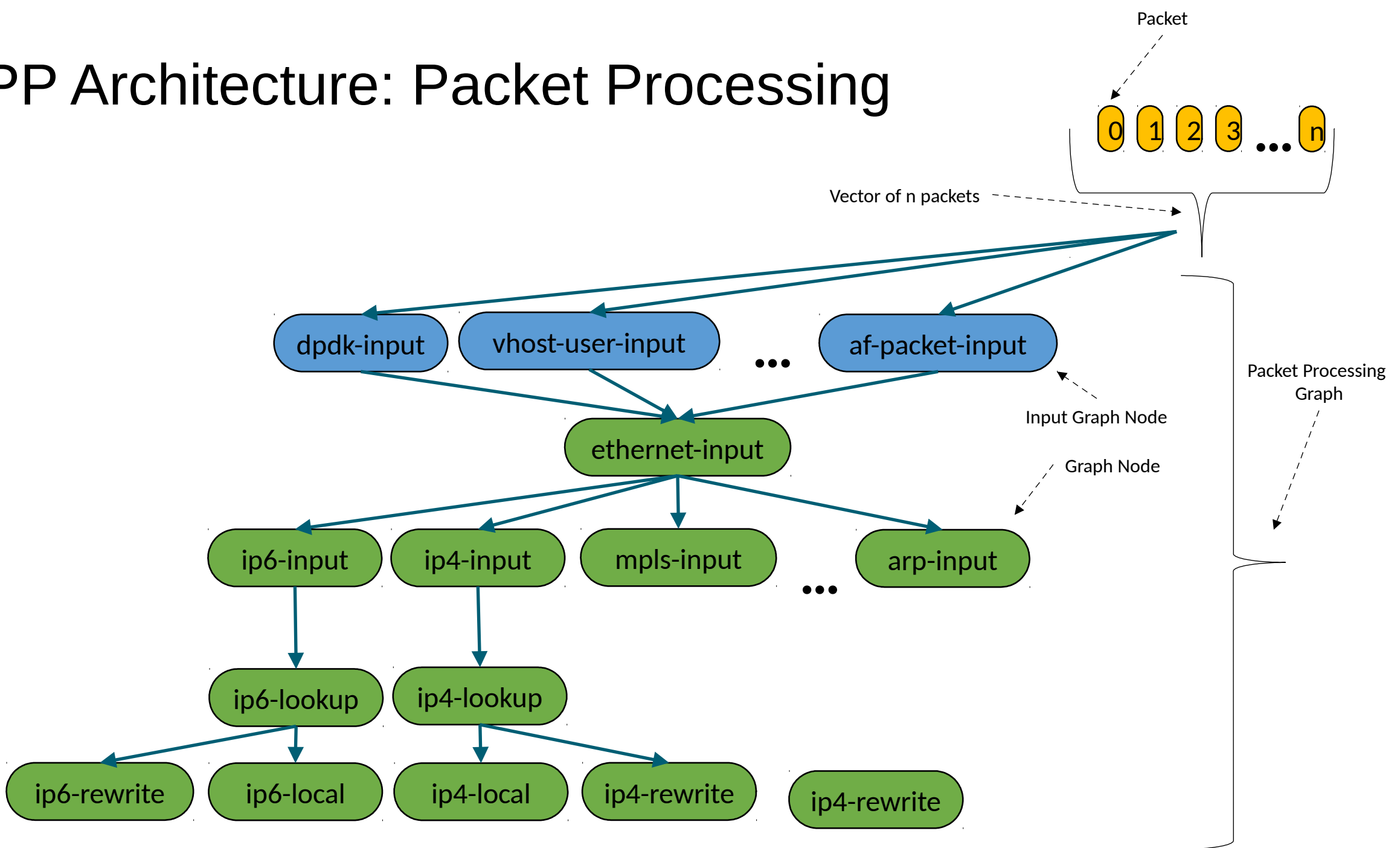
- High performance
- Linux User space

- Run's on commodity CPUs:

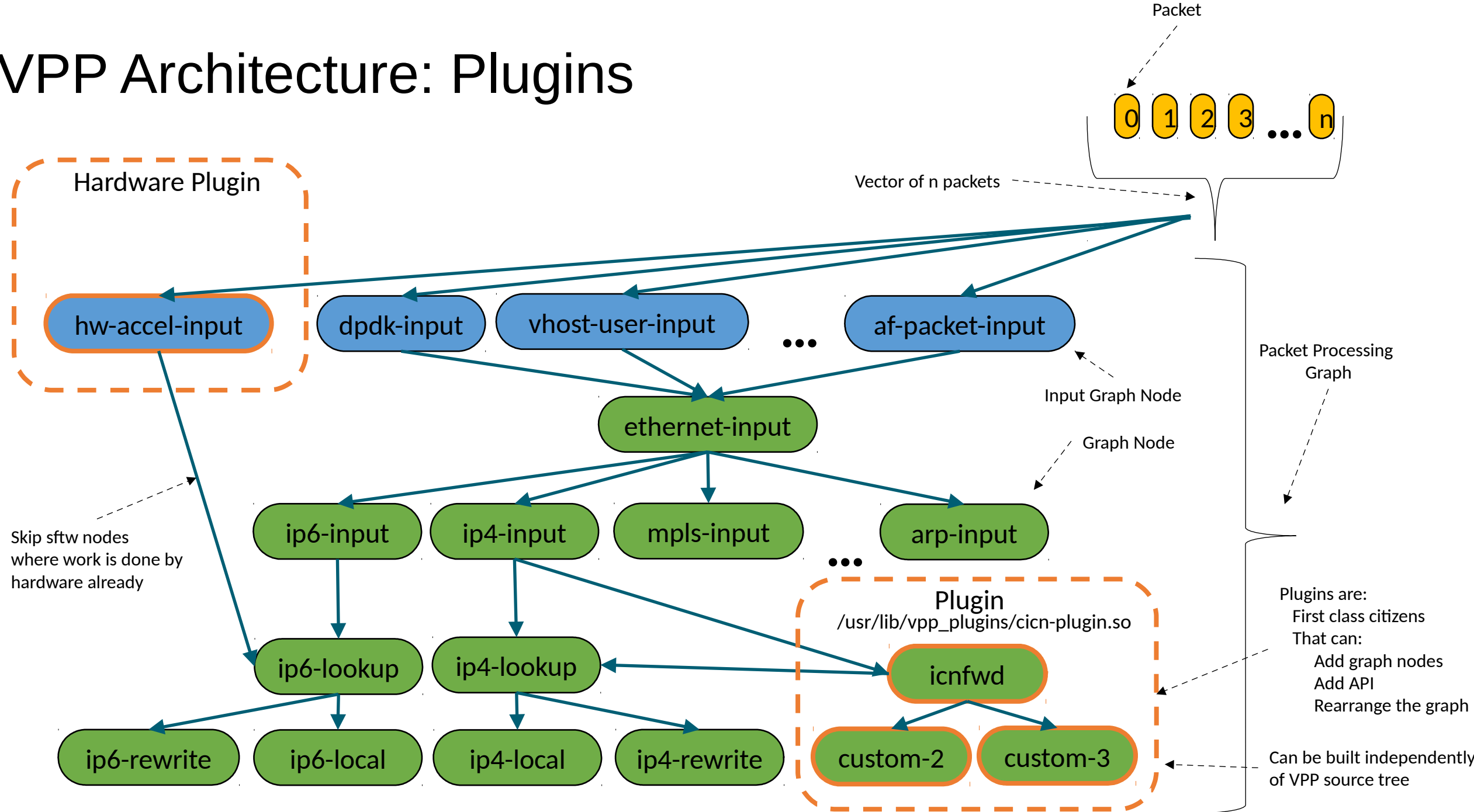


- Shipping at volume in server & embedded products since 2004.

VPP Architecture: Packet Processing

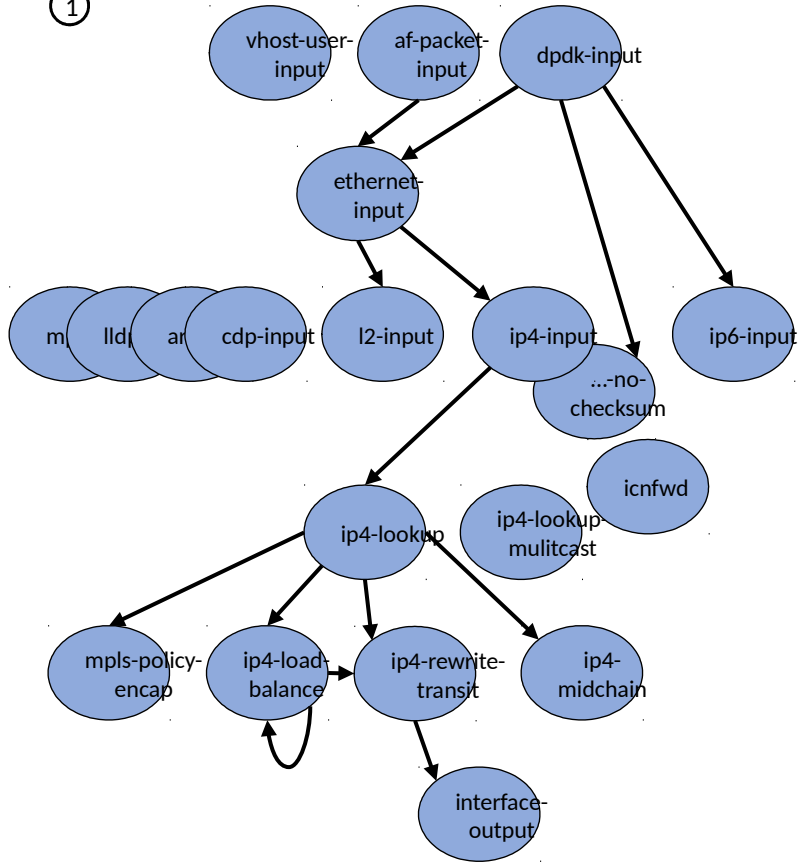


VPP Architecture: Plugins

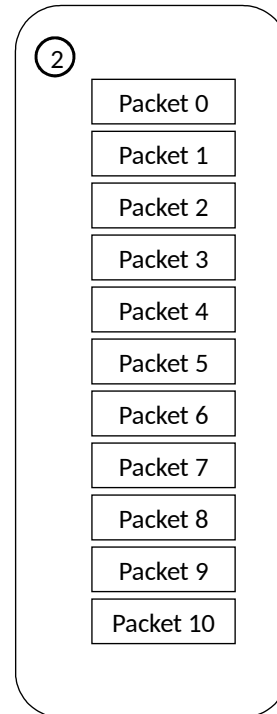


VPP: How does it work?

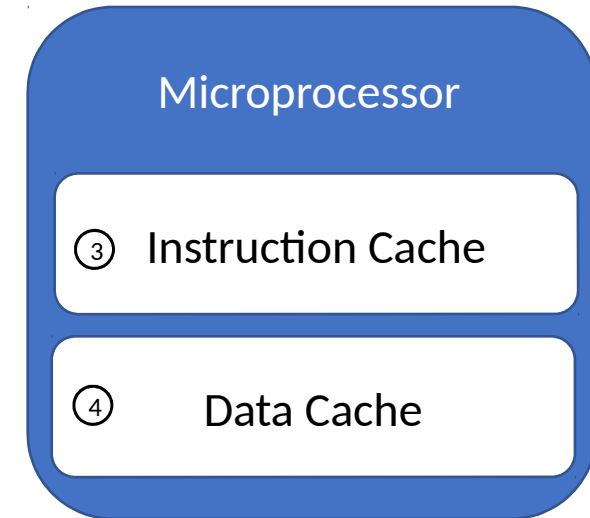
①



②



... graph nodes are optimized to fit inside the instruction cache ...



... packets moved through graph nodes in vector ...

... packets are pre-fetched, into the data cache ...

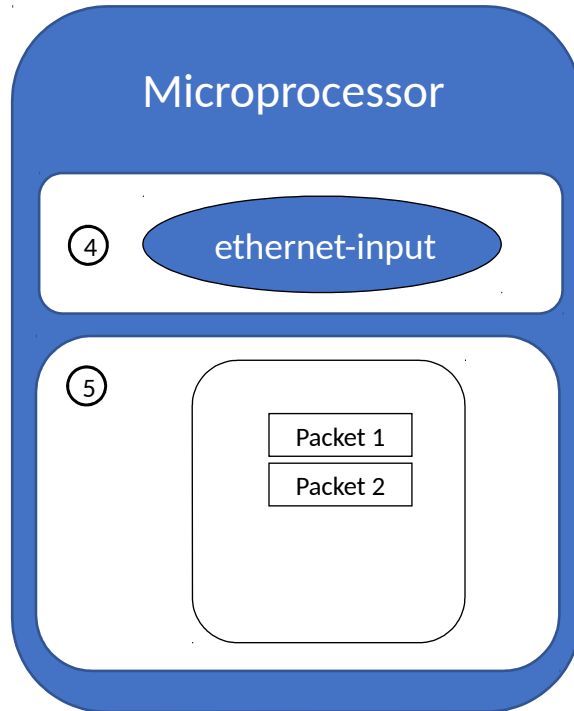
Packet processing is decomposed into a directed graph node ...

* approx. 173 nodes in default deployment

VPP: How does it work? ^⑥

dispatch fn()

... instruction cache is warm with the instructions from a single graph node ...



... data cache is warm with a small number of packets ..

while packets in vector

Get pointer to vector

while 4 or more packets

PREFETCH #3 and #4

PROCESS #1 and #2

ASSUME next_node same as last packet

Update counters, advance buffers

Enqueue the packet to next_node

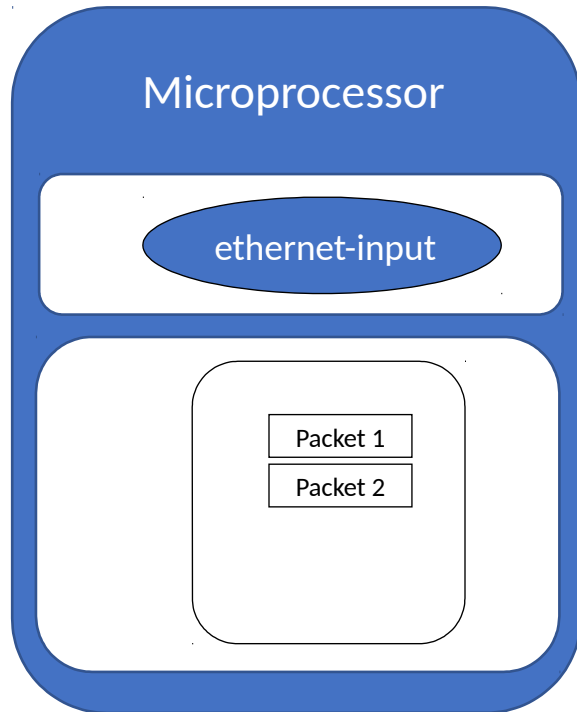
while any packets

<as above but single packet>

... packets are processed in groups of four, any remaining packets are processed on by one ...

VPP: How does it work?

⑦



dispatch fn()

while packets in vector

Get pointer to vector

while 4 or more packets

PREFETCH #1 and #2

PROCESS #1 and #2

ASSUME next_node same as last packet

Update counters, advance buffers

Enqueue the packet to next_node

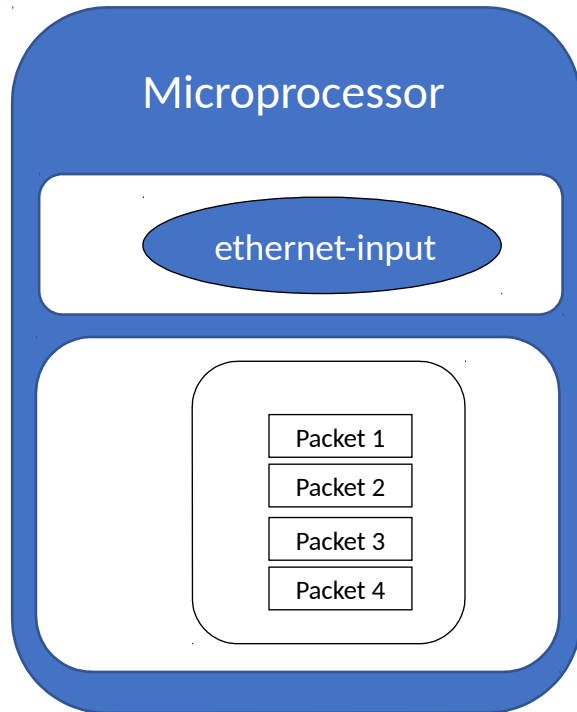
while any packets

<as above but single packet>

... prefetch packets #1 and #2 ...

VPP: How does it work?

⑧



dispatch fn()

while packets in vector

Get pointer to vector

while 4 or more packets

PREFETCH #3 and #4

PROCESS #1 and #2

ASSUME next_node same as last packet

Update counters, advance buffers

Enqueue the packet to next_node

while any packets

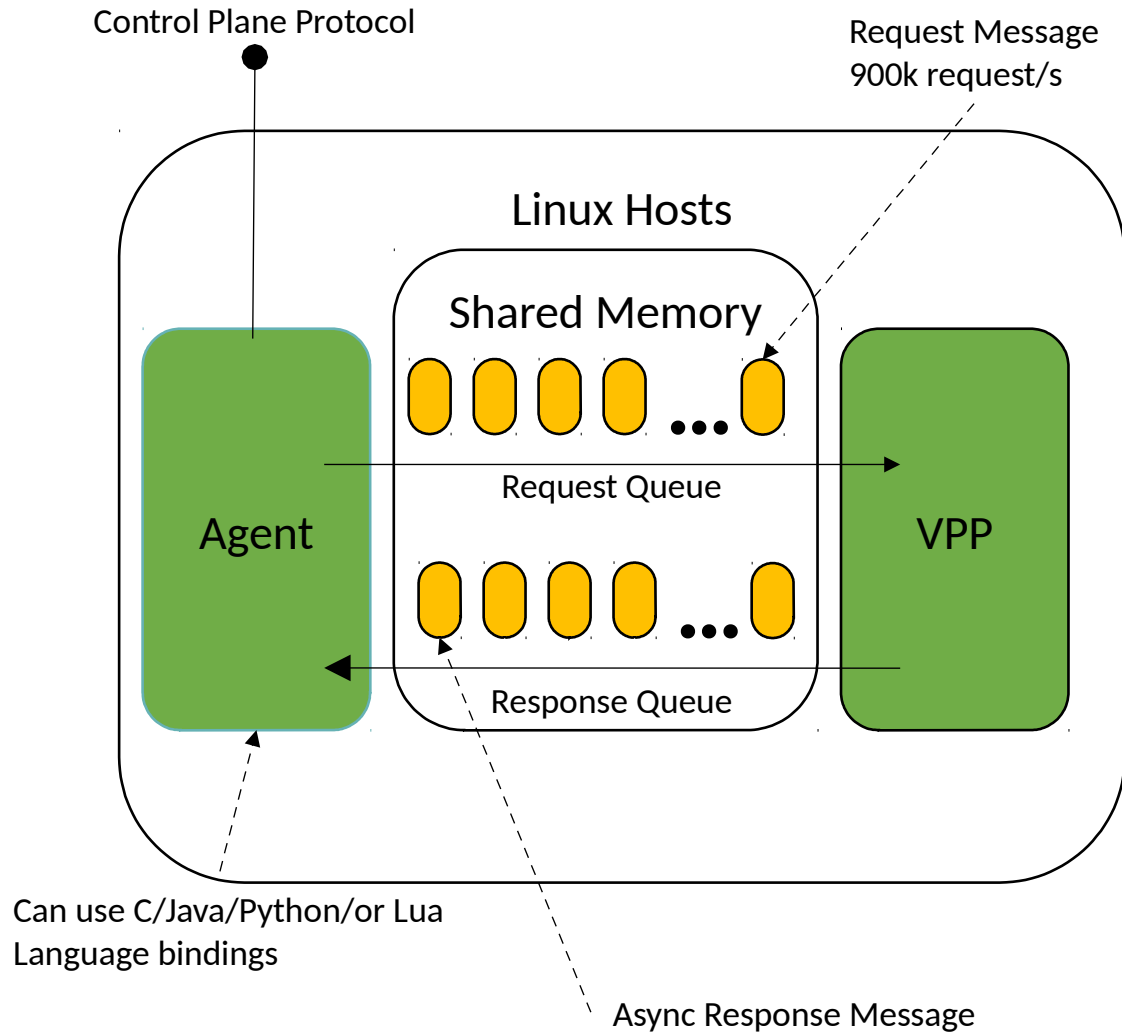
<as above but single packet>

... process packet #3 and #4 ...

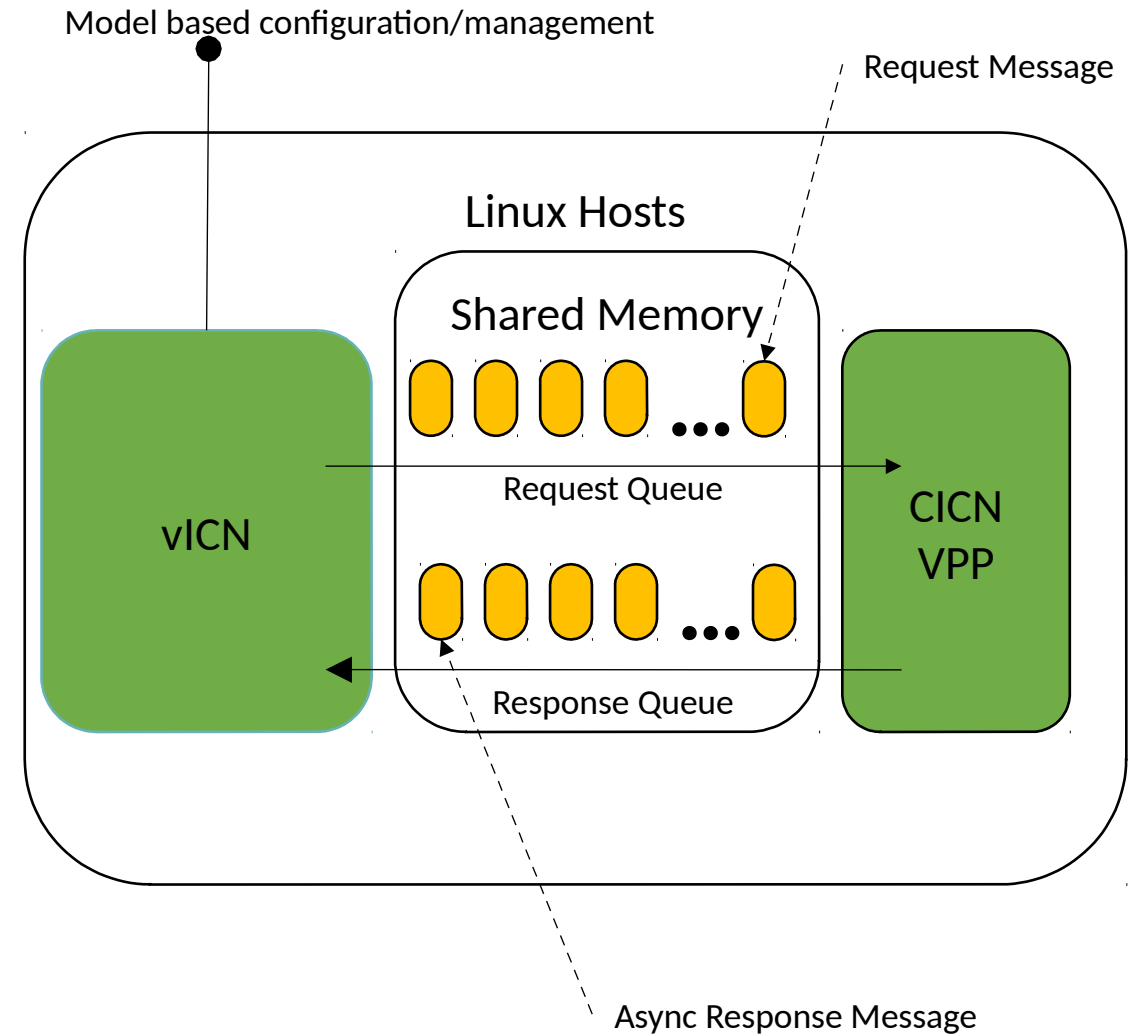
... update counters, enqueue packets to the next node ...

VPP Architecture: Programmability

Architecture



Example: v1CN



Universal Dataplane: Features



Hardware Platforms

Pure Userspace - X86,ARM 32/64,Power
Raspberry Pi

Interfaces

DPDK/Netmap/AF_Packet/TunTap
Vhost-user - multi-queue, reconnect,
Jumbo Frame Support

Language Bindings

C/Java/Python/Lua

Tunnels/Encaps

GRE/VXLAN/VXLAN-GPE/LISP-GPE/NSH
IPSEC
Including HW offload when available

MPLS

MPLS over Ethernet/GRE
Deep label stacks supported

Routing

IPv4/IPv6
14+ MPPS, single core
Hierarchical FIBs
Multimillion FIB entries
Source RPF
Thousands of VRFs
Controlled cross-VRF lookups
Multipath - ECMP and Unequal Cost

ICN

PIT/CS/FIB
Strategy layer

Segment Routing

SR MPLS/IPv6
Including Multicast

LISP

LISP xTR/RTR
L2 Overlays over LISP and GRE encaps
Multitenancy
Multihome
Map/Resolver Failover
Source/Dest control plane support
Map-Register/Map-Notify/RLOC-probing

Switching

VLAN Support
Single/ Double tag
L2 forwd w/EFP/BridgeDomain concepts
VTR - push/pop/Translate (1:1,1:2, 2:1,2:2)
Mac Learning - default limit of 50k addr
Bridging
Split-horizon group support/EFP Filtering
Proxy Arp
Arp termination
IRB - BVI Support with RouterMac assignmt
Flooding
Input ACLs
Interface cross-connect
L2 GRE over IPsec tunnels

Security

Mandatory Input Checks:
TTL expiration
header checksum
L2 length < IP length
ARP resolution/snooping
ARP proxy
SNAT
Ingress Port Range Filtering
Per interface whitelists
Policy/Security Groups/GBP (Classifier)

Network Services

DHCPv4 client/proxy
DHCPv6 Proxy
MAP/LW46 - IPv4aaS
MagLev-like Load
Identifier Locator Addressing
NSH SFC SFF's & NSH Proxy
LLDP
BFD
Policer
Multiple million Classifiers -
Arbitrary N-tuple

Inband iOAM

Telemetry export infra (raw IPFIX)
iOAM for VXLAN-GPE (NGENA)
SRv6 and iOAM co-existence
iOAM proxy mode / caching
iOAM probe and responder

Monitoring

Simple Port Analyzer (SPAN)
IP Flow Export (IPFIX)
Counters for everything
Lawful Intercept

Continuous Quality, Performance, Usability

Built into the development process – patch by patch



Build/Unit Testing
120 Tests/Patch

Build binary packaging for
Ubuntu 14.04
Ubuntu 16.04
Centos 7

Automated Style Checking

Unit test :

IPFIX	IPv6
BFD	IP Multicast
Classifier	L2 FIB
DHCP	L2 Bridge Domain
FIB	MPLS
GRE	SNAT
IPv4	SPAN
IPv4 IRB	VXLAN
IPv4 multi-VRF	

System Functional Testing
252 Tests/Patch

DHCP - Client and Proxy
GRE Overlay Tunnels
L2BD Ethernet Switching
L2 Cross Connect Ethernet Switching
LISP Overlay Tunnels
IPv4-in-IPv6 Software Tunnels
Cop Address Security
IPSec
IPv6 Routing - NS/ND, RA, ICMPv6
uRPF Security
Tap Interface
Telemetry - IPFIX and Span
VRF Routed Forwarding
iACL Security - Ingress - IPv6/IPv6/Mac
IPv4 Routing
QoS Policer Metering
VLAN Tag Translation
VXLAN Overlay Tunnels

Performance Testing
144 Tests/Patch, 841 Tests

L2 Cross Connect
L2 Bridging
IPv4 Routing
IPv6 Routing
IPv4 Scale - 20k,200k,2M FIB Entries
IPv4 Scale - 20k,200k,2M FIB Entries
VM with vhost-user
PHYS-VPP-VM-VPP-PHYS
L2 Cross Connect/Bridge
VXLAN w/L2 Bridge Domain
IPv4 Routing
COP - IPv4/IPv6 whiteless
iACL - ingress IPv4/IPv6 ACLs
LISP - IPv4-o-IPv6/IPv6-o-IPv4
VXLAN
QoS Policer
L2 Cross over
L2 Bridging

Usability

Merge-by-merge:
apt installable deb packaging
yum installable rpm packaging
autogenerated code documentation
autogenerated cli documentation

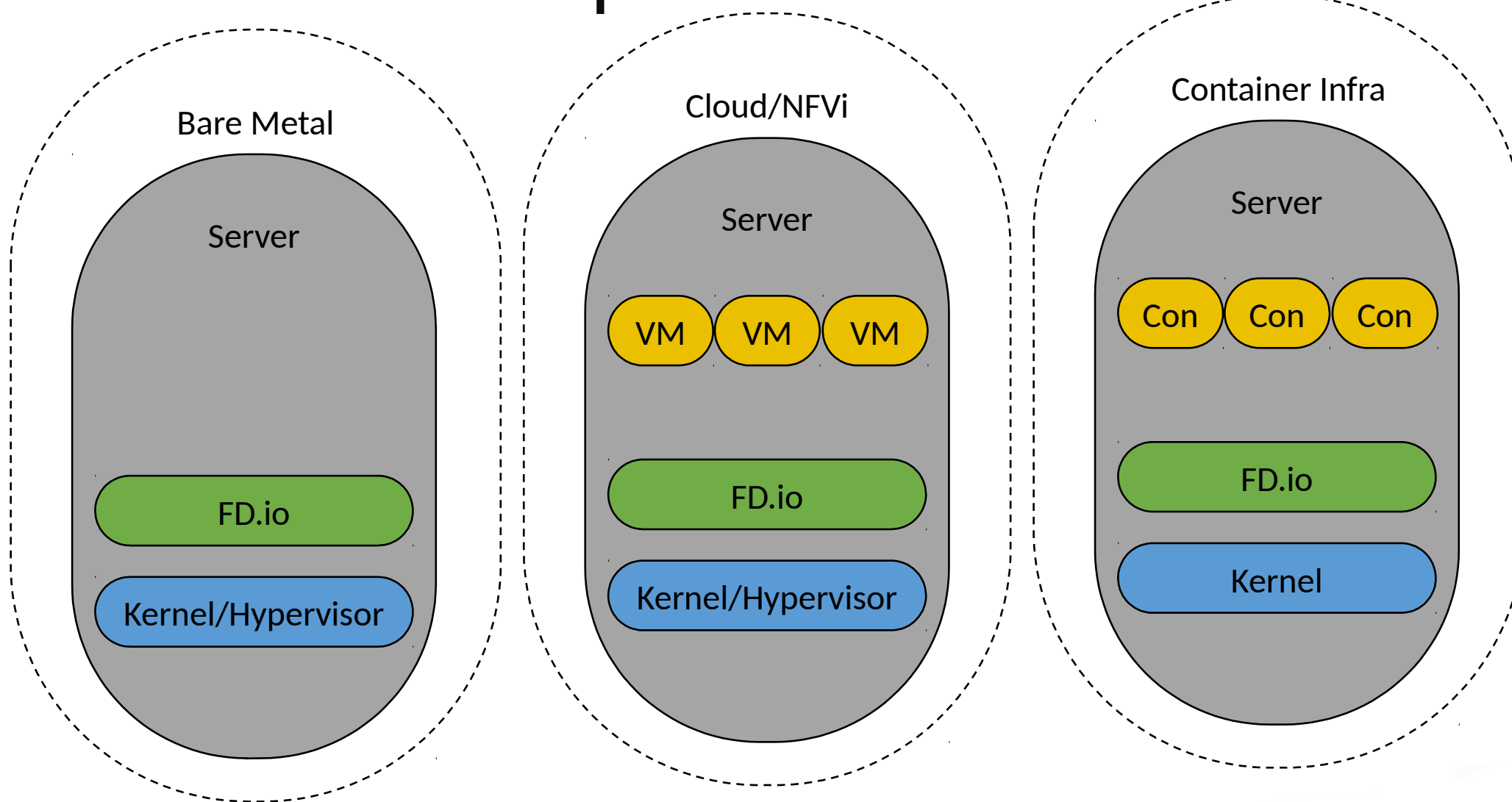
Per release:
autogenerated testing reports
report perf improvements

Puppet modules
Training/Tutorial videos
Hands-on-usecase documentation

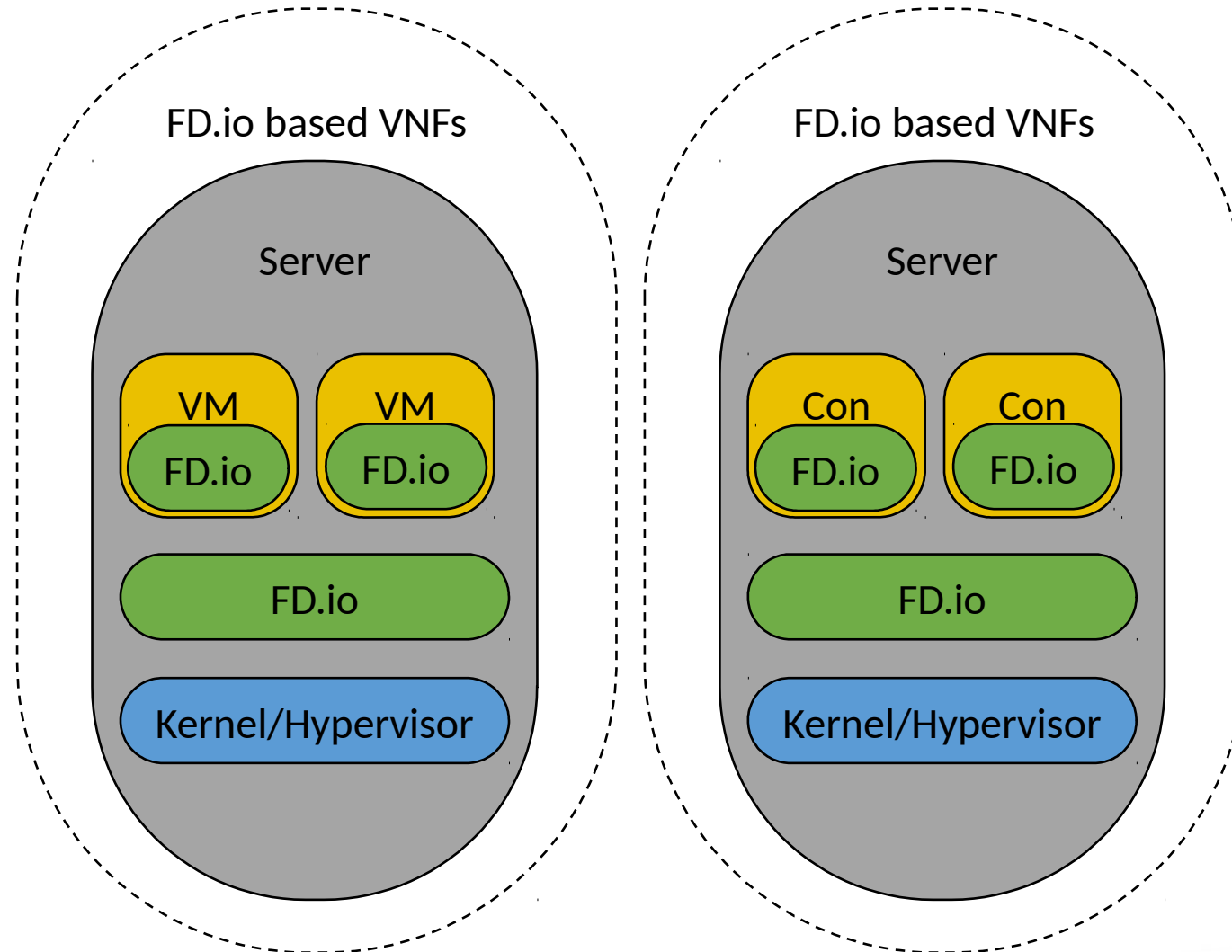
Merge-by-merge packaging feeds
Downstream consumer CI pipelines

Run on real hardware in fd.io Performance Lab

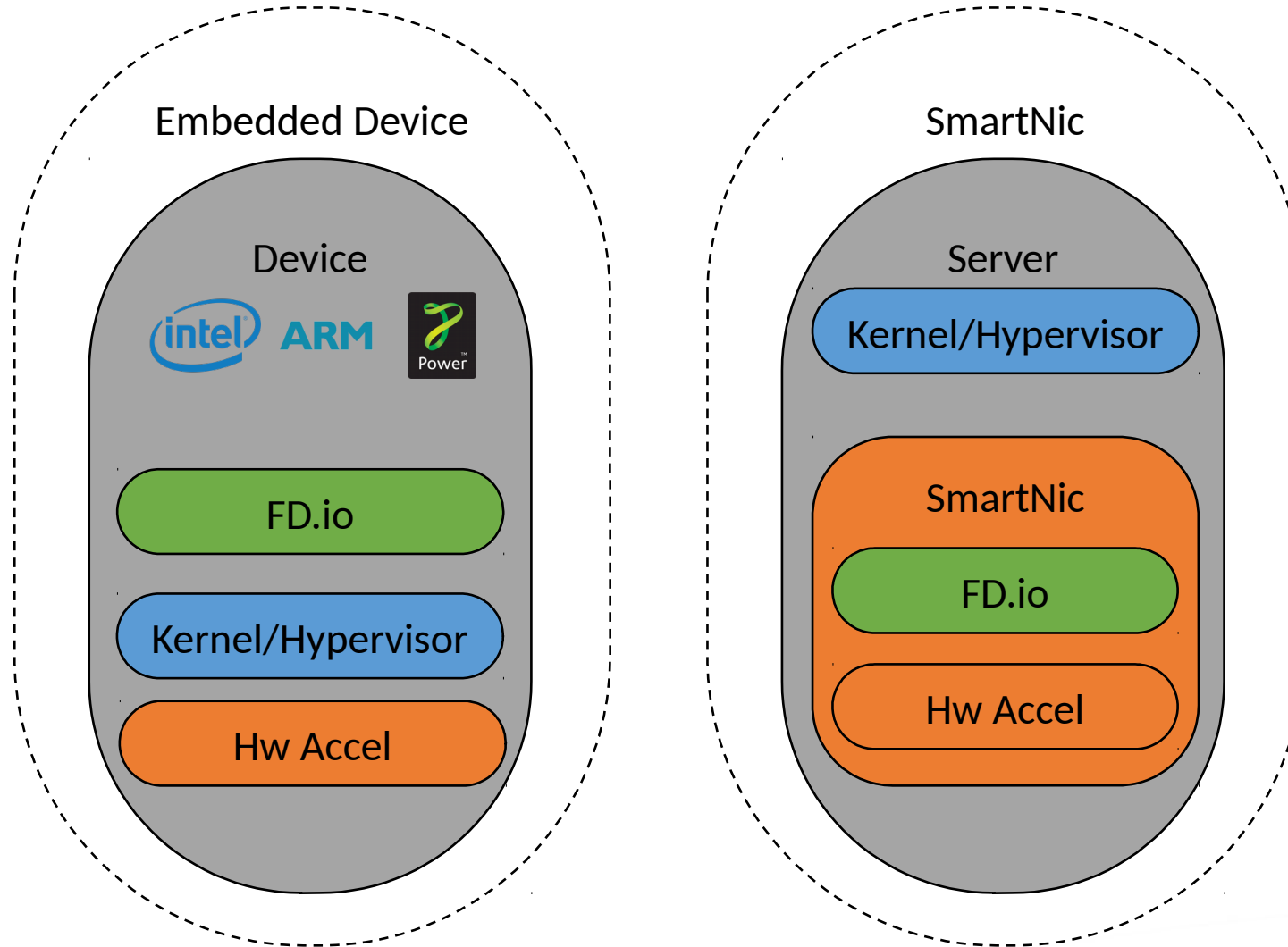
Universal Dataplane: Infrastructure



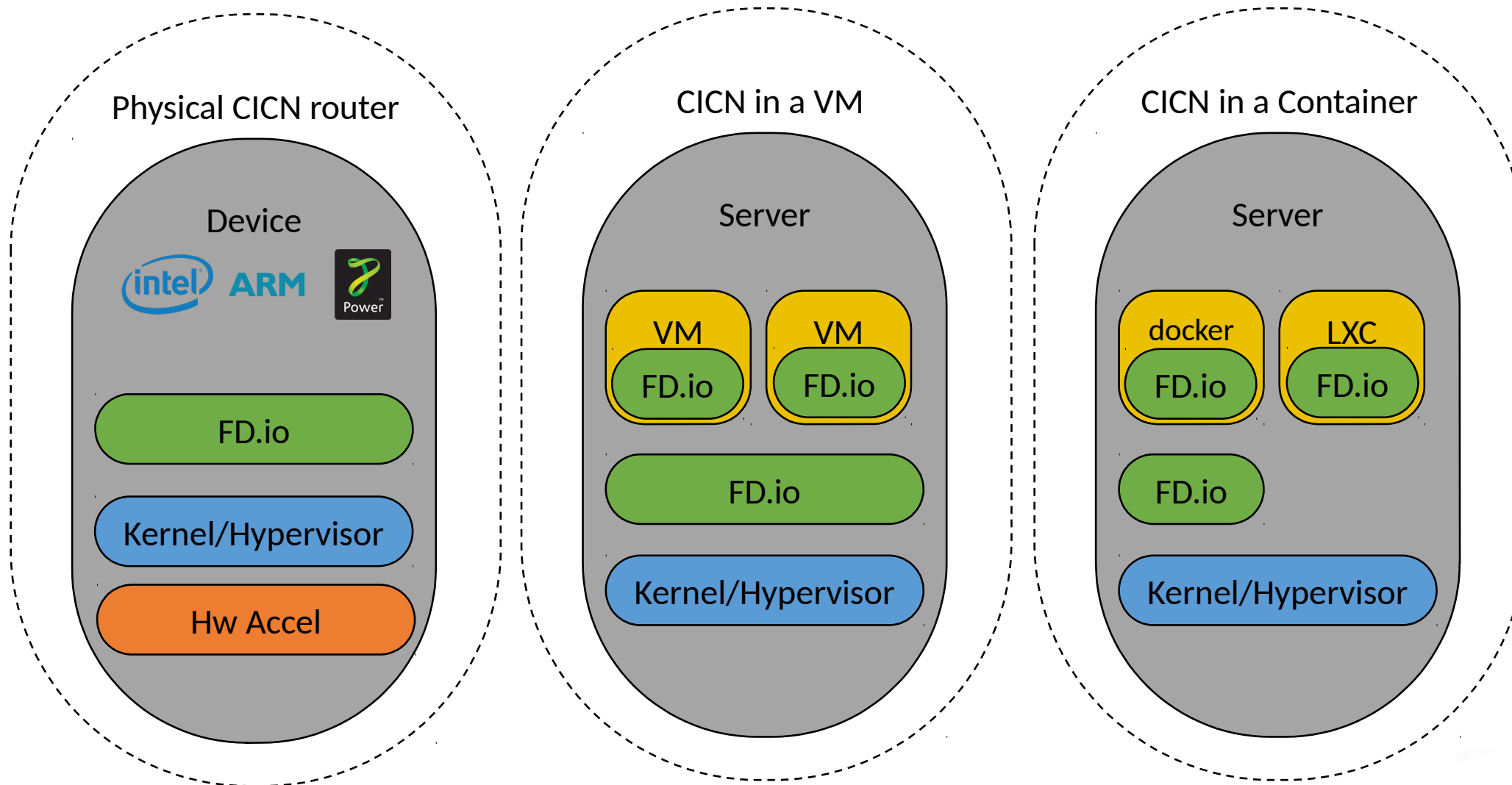
Universal Dataplane: VNFs



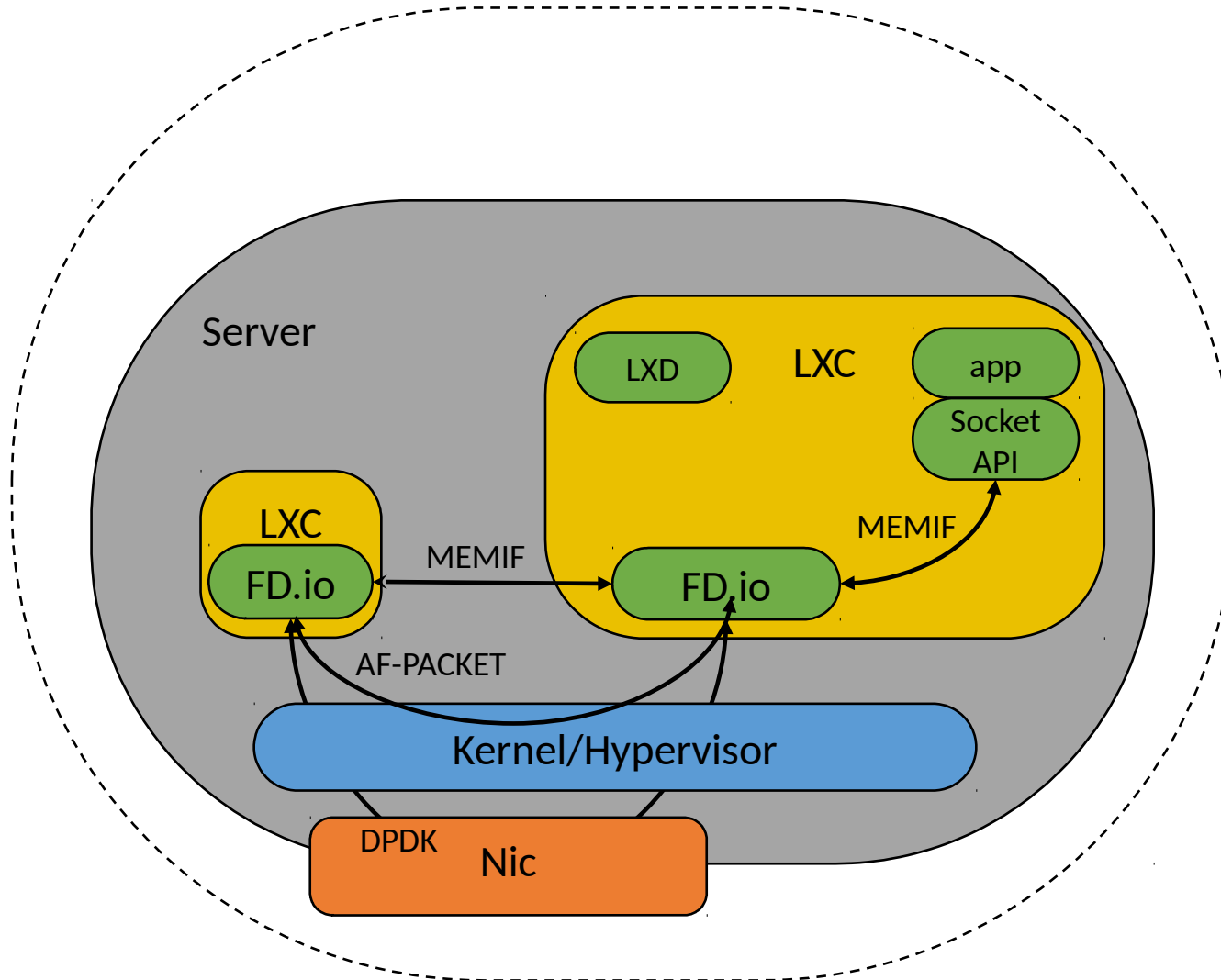
Universal Dataplane: Embedded



Universal Dataplane: CICN Example



Universal Dataplane: communication/API



Consumer/Producer Socket API

- Segmentation/Naming
- Manifest management
- Reassembly
- Flow and Congestion Control

Existing drivers for links

- DPDK
- AF-PACKET
- MEMIF (SHARED MEMORY)

CICN distribution

- Core libraries
 - Consumer/Producer Socket API, CCNx libs, PARC C libraries
- Server and Router
 - VPP cicn plugin for Ubuntu 16, CentOS 7
 - HTTP video server
- Client
 - Metis Forwarder
 - VIPER MPEG-DASH video player
 - Android 7, MacOS X 10.12, iOS 10, Ubuntu 16, CentOS 7
 - Soon Apple Store and Google Play
- vICN
 - intent-based networking
 - model driven programmable framework
 - monitoring and streaming for BigData support

Opportunities to Contribute



- Forwarding strategies
- Mobility management
- Hardware Accelerators
- vICN, configuration/management/control
- Consumer/Producer Socket API
- Reliable Transport
- Instrumentation tools
- HTTP integration

We invite you to Participate in fd.io

- [Get the Code, Build the Code, Run the Code, install from binaries](#)
- [from binary packages](#)
- [Read/Watch the Tutorials](#)
- [Join the Mailing Lists](#)
- [Join the IRC Channels](#)
- [Explore the wiki](#)
- [Join fd.io as a member](#)
- <https://wiki.fd.io/view/cicn>
- <https://wiki.fd.io/view/vicn>
- <https://fd.io/>

