

# Open FastPath

An open source user space fast path TCP/IP stack

# ➤ What is Our Intention with OpenFastPath (OFP)?

## **To enable efficient IP communication**

- Essential in practically all networking use-cases, including NFV

## **To enable efficient system design**

- The IP stack runs close to the underlying hardware and can leverage hardware features

## **To lower the barriers to build network applications in a multivendor environment**

- ODP and DPDK support offers application portability across hardware architectures

## **To reduce development costs through Open Source business model**

- Enables companies to focus investments on differentiating features

# ➤ What is OpenFastPath?

## IP fast path incubation project between Nokia, Enea and ARM

- Developers and supporting staff active from all three companies

## Organized as an Open source project

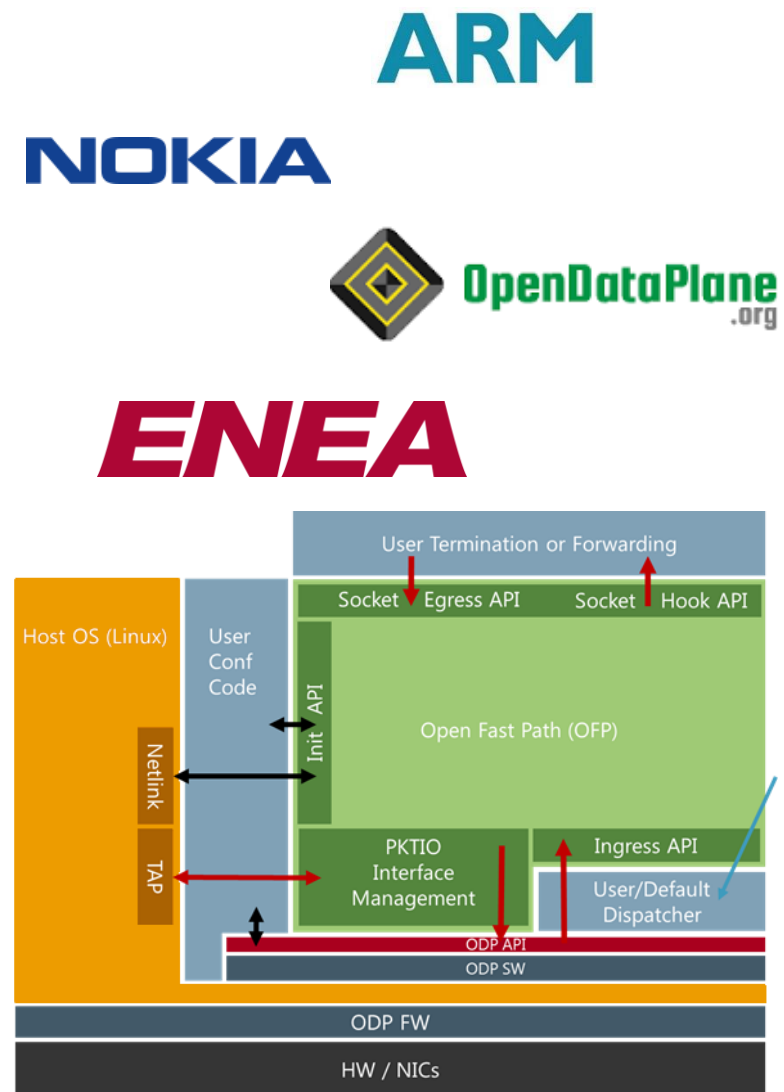
- Democratic governance model
- Using open source software
- BSD license
- Source code on GitHub
- Mailing list, shared Google docs, weekly calls, Freenode chat etc

## Leverages ODP and DPDK to access hardware from user space

- Maximizes portability and software reuse
- Native DPDK is evaluated

## Linux user space IP fast path implementation

- Maximizes throughput and scalability by minimizing Linux overhead
- Leverages Linux for slowpath and route/ARP table updates



# Features implemented

## **Fast path protocols processing:**

- Layer 4: UDP termination, TCP termination, ICMP protocol
- Layer 3
  - ARP/NDP
  - IPv4 and IPv6 forwarding and routing
  - IPv4 fragmentation and reassembly
  - VRF for IPv4
  - IGMP and multicast
- Layer 2: Ethernet, VLAN, VxLAN
- GRE Tunneling

## **Routes and MACs are in sync with Linux through Netlink**

## **Integration with Linux Slow path IP stack through TAP interface**

## **Command line interface**

- Packet dumping and other debugging
- Statistics, ARP, routes, and interface printing
- Configuration of routes and interfaces with VRF support

## **OFPP IP and ICMP implementations passing Ixia conformance tests**

## **IP and UDP implementations has been optimized for performance, TCP implementation is functional but not performance optimized**

## **Project roadmap candidates**

- IPsec, Packet filtering, OFPP application IPC, OFPP on RTOS, SCTP, GTP-U

# ➤ OpenFastPath Source code

## **New open-source code**

- Developed by partners during the incubation stage

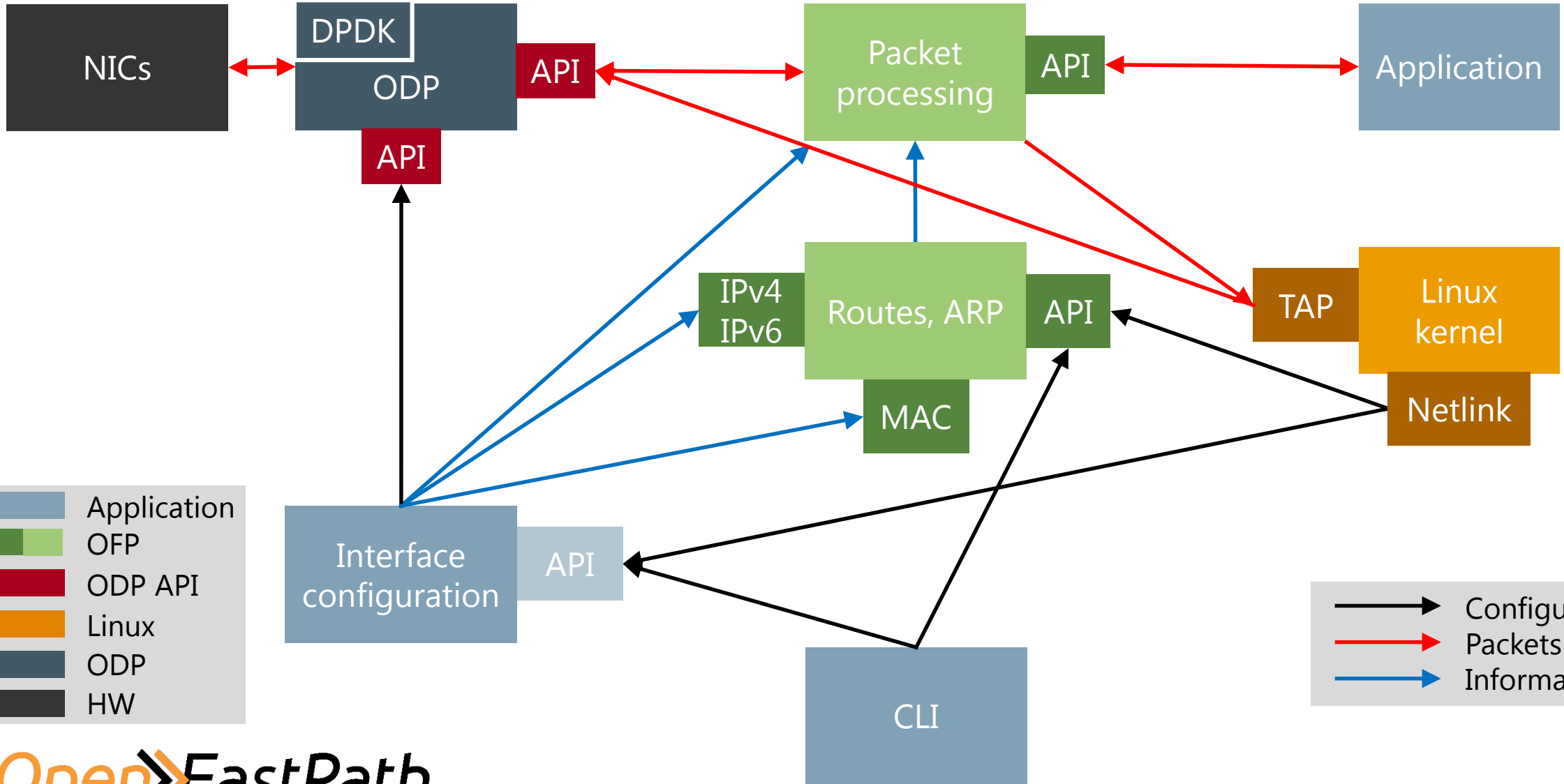
## **UDP, TCP, ICMP code was ported from libuinet (User space FreeBSD port)**

- Non-blocking event based socket API
- Modular multithreaded design focused on performance and scalability
- Tightly coupled to application, linked in as a library
- Maintainability – Tracks evolution of FreeBSD

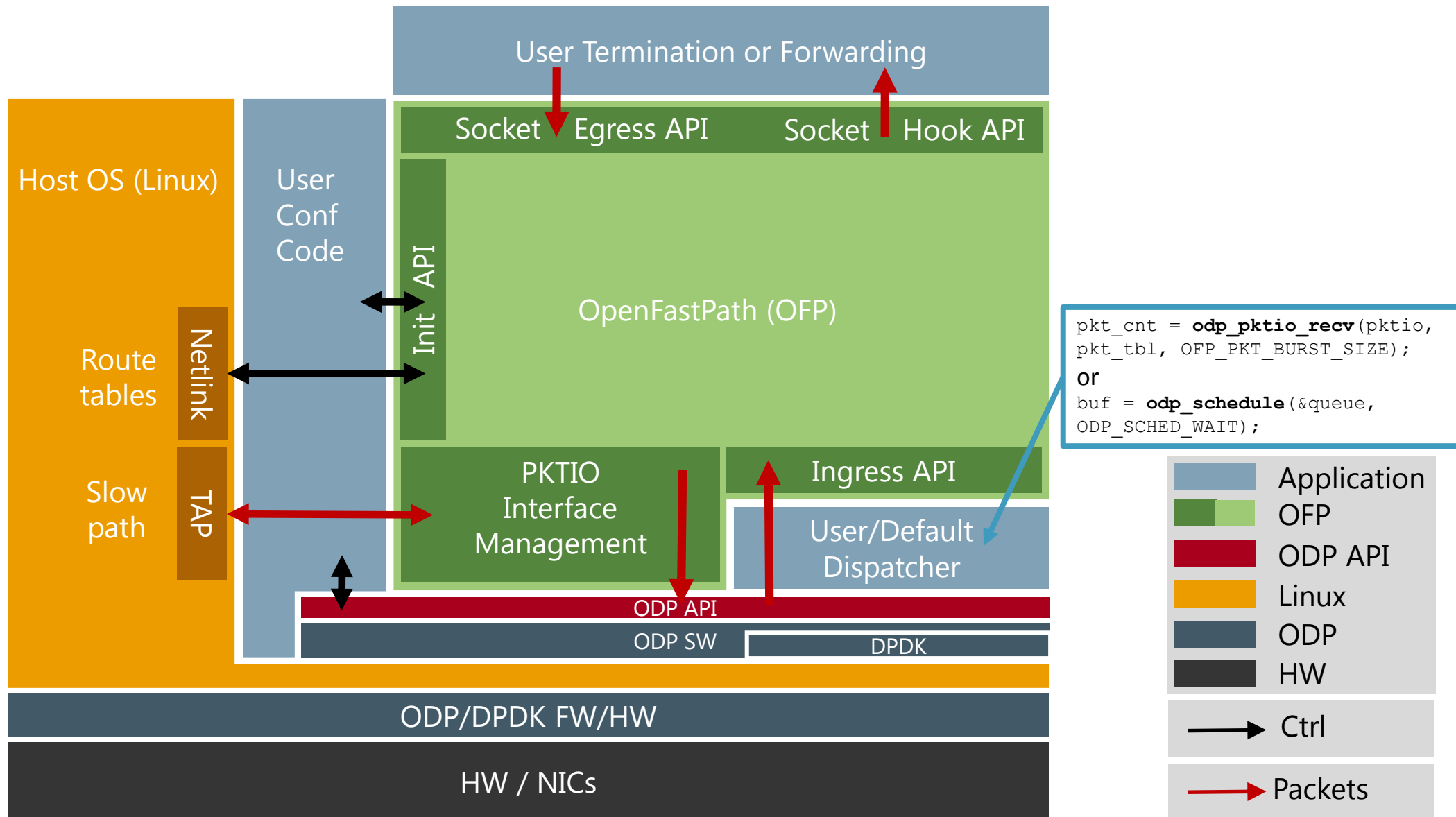
## **High performance and scalable implementation for MAC and Route tables**

- Lockless synchronization based on epoch reclamation

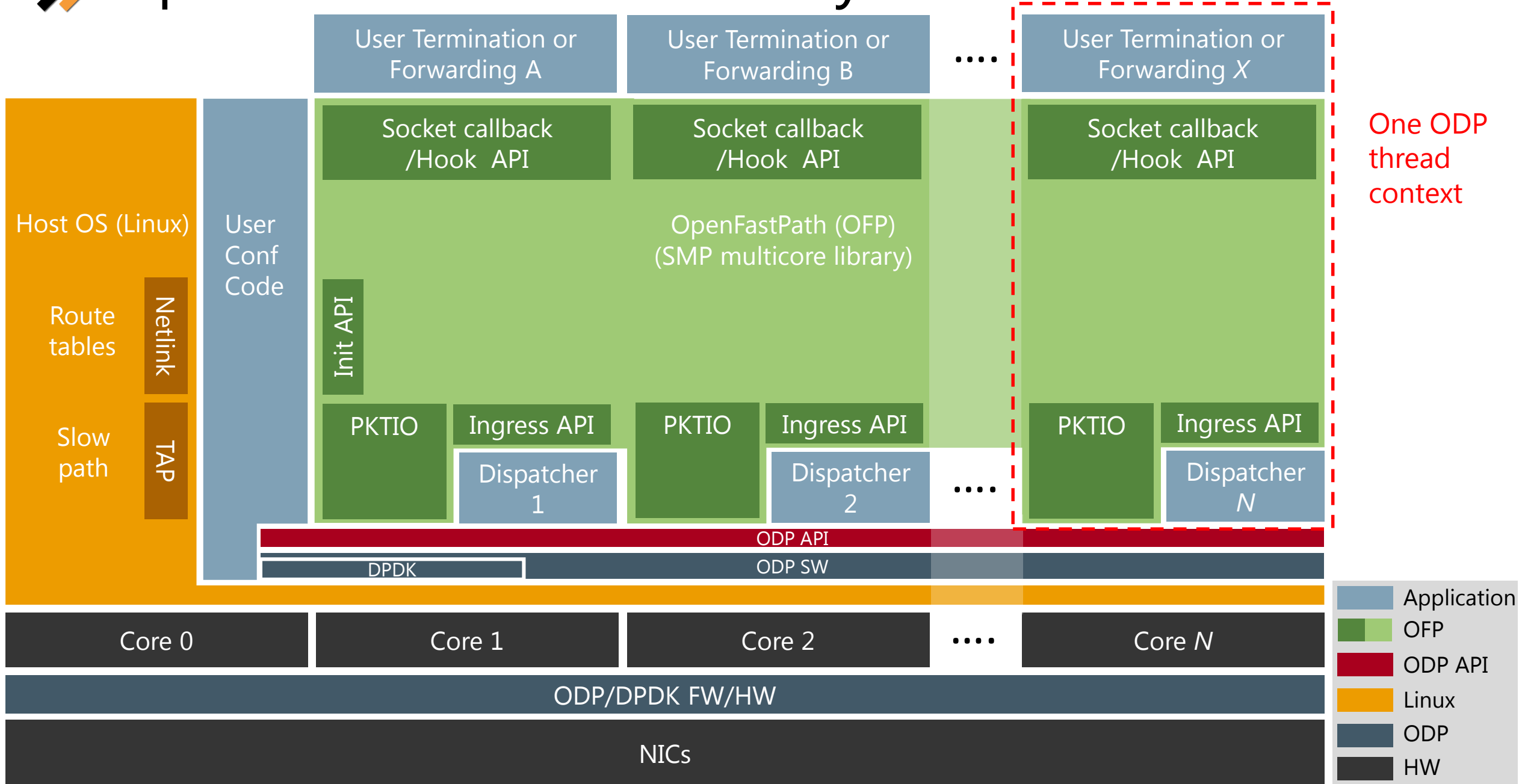
# OpenFastPath system components



# OpenFastPath System View

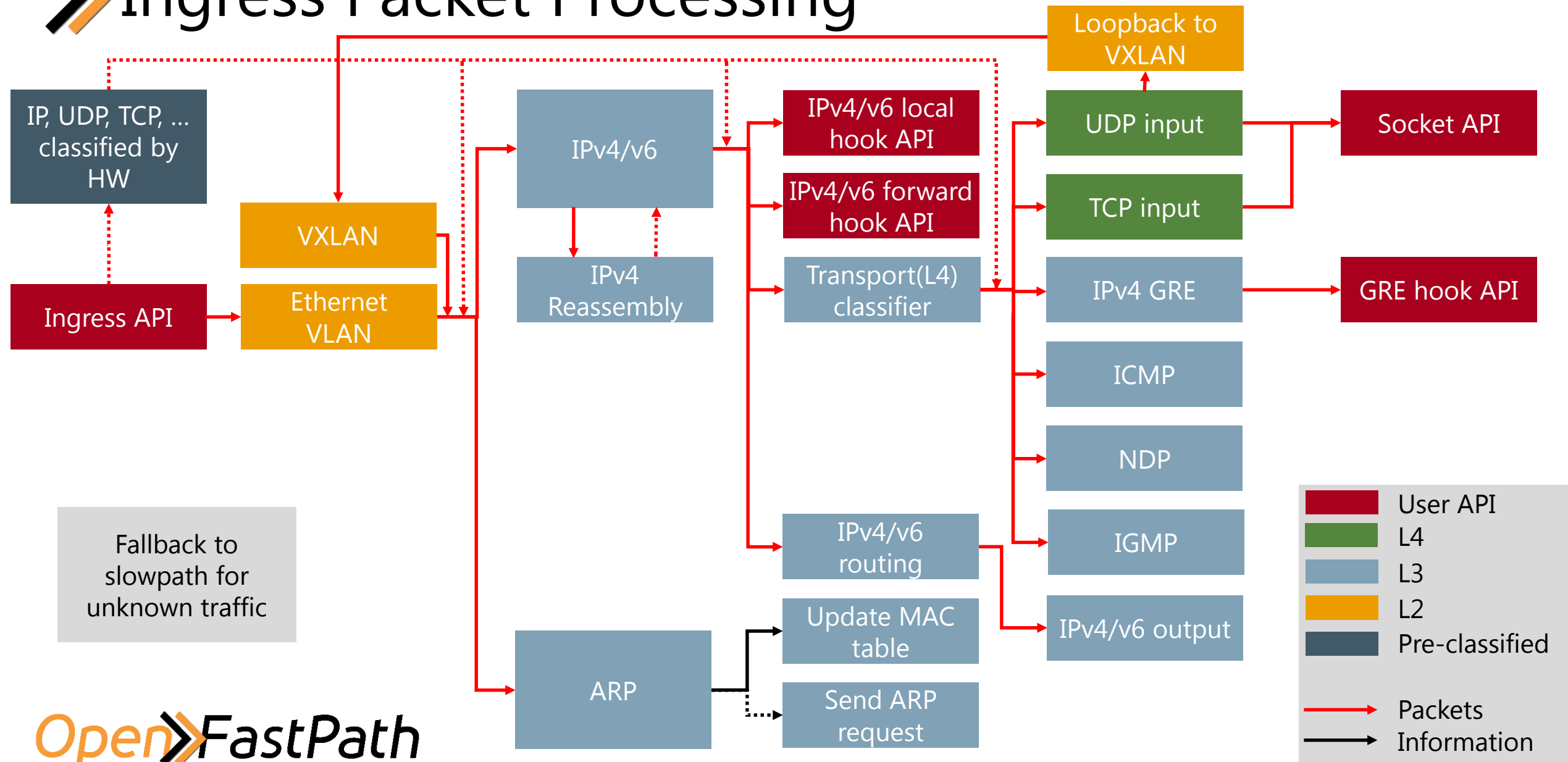


# OpenFastPath multicore System View

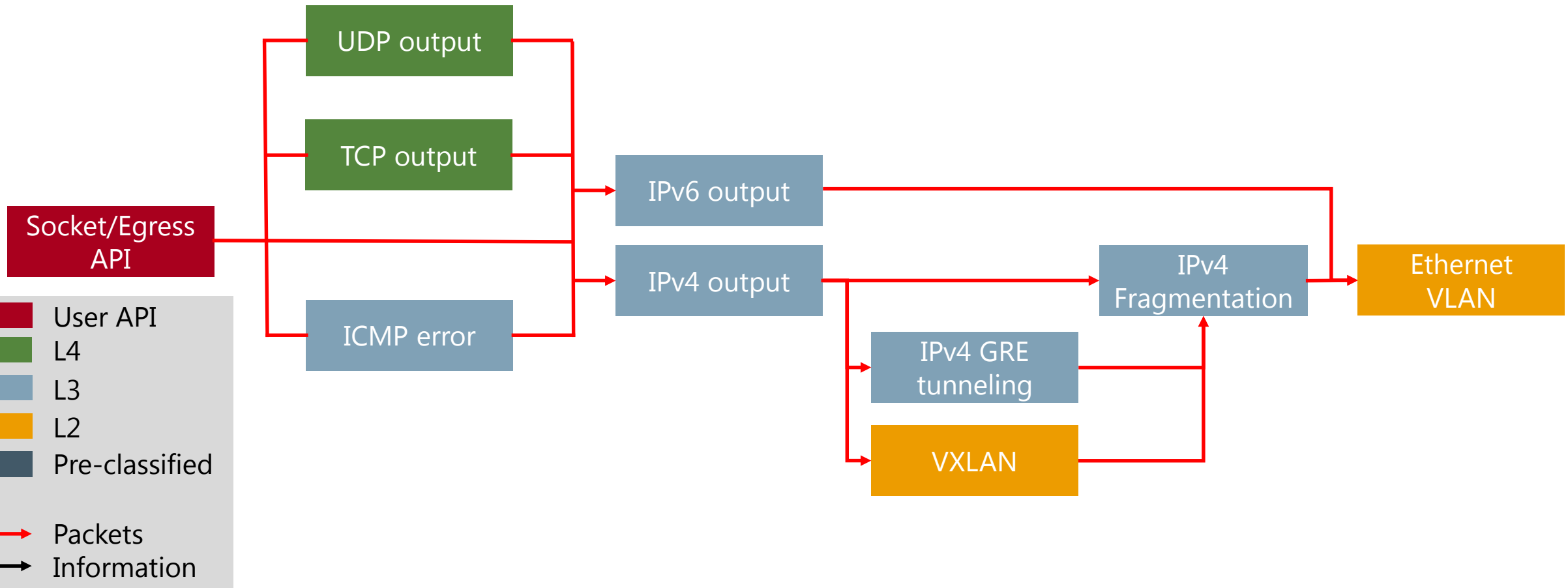




# ➤ Ingress Packet Processing



# >> Egress Packet Processing



# ➤ Optimized OpenFastPath socket APIs

## **New zero-copy APIs optimized for single thread run-to-completion environments**

- UDP
  - Send: Optimized send function with ODP packet container (`odp_packet_t`)
  - Receive: A function callback can be registered to read on a socket. Receives ODP packet container and socket handle
- TCP
  - Accept event: A function callback can be registered for TCP accept event. Receives socket handle.
  - Receive: A function callback can be registered to read on socket. Receives ODP packet container and a socket handle

## **Standard BSD Socket interface**

- For compatibility with legacy applications

# ➤ Other OpenFastPath user application APIs

- Initiation of Open Fast Path
- Interface configuration
- Route and MAC table access
- Packet Ingress and Egress processing
- Hooks for IP local, IP forwarding and GRE
- Timer callbacks
- Statistics
- Packet capture

# ➤ Why should someone use OpenFastPath?

## **Open source project**

- Enables companies to focus investments on differentiating features

## **Highly optimized and scalable solution**

- Non-blocking event based API focused on performance and scalability

## **Portable high performance solution supporting multiple HW and SW platforms**

- Functionality verified on ARM, MIPS and x86 HW with native ODP, ODP-DPDK, ODP-NETMAP support

## **Reliable**

- OFP IP and ICMP implementations passing Ixia conformance test

## **User space implementation**

- Simplifies maintenance and maximizes throughput and scalability by minimizing Linux kernel dependency

# » Thank You

For additional information, please visit  
[www.openfastpath.org](http://www.openfastpath.org)