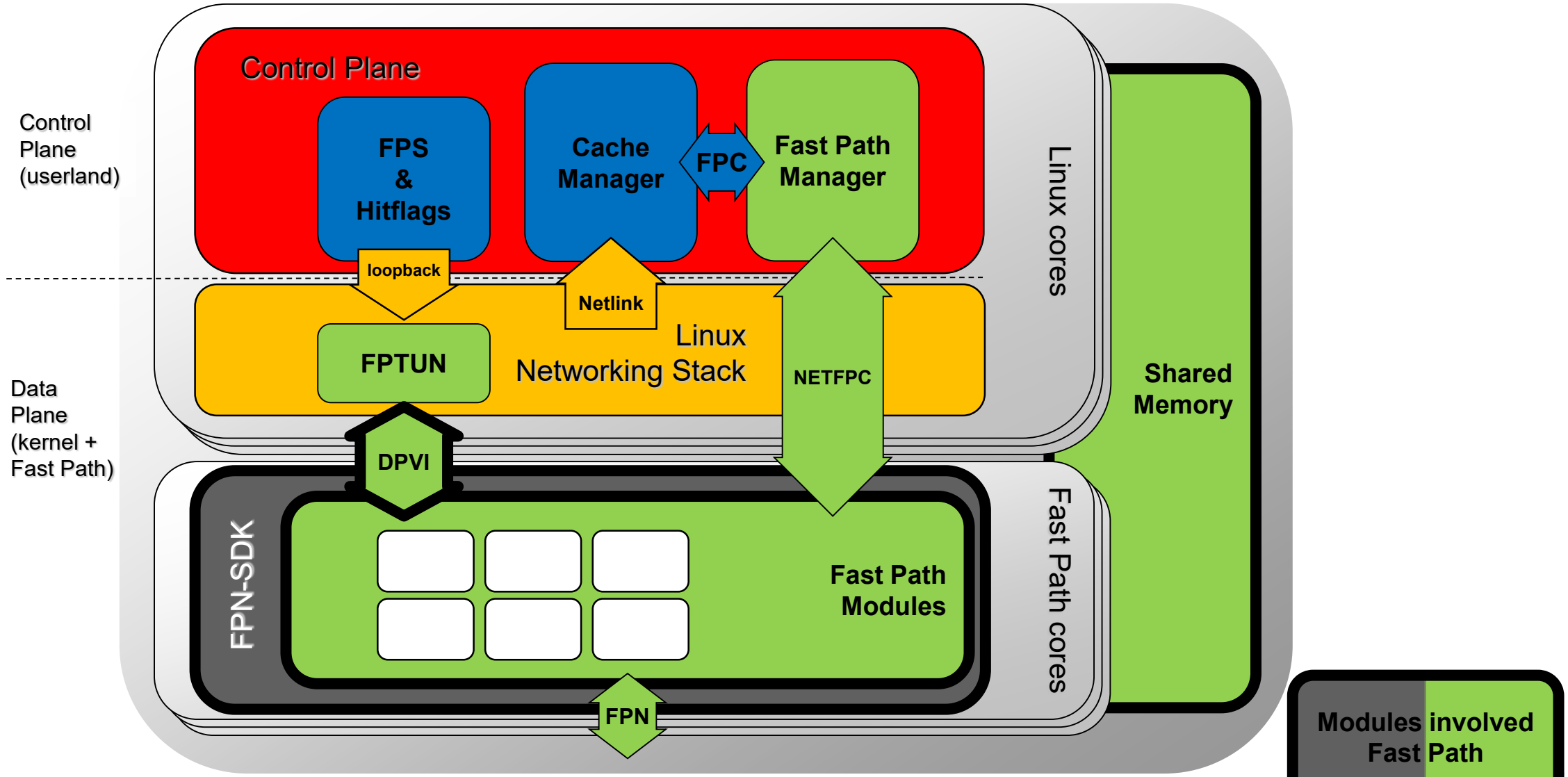
6WINDGate Fast Path Implementation

6WIND

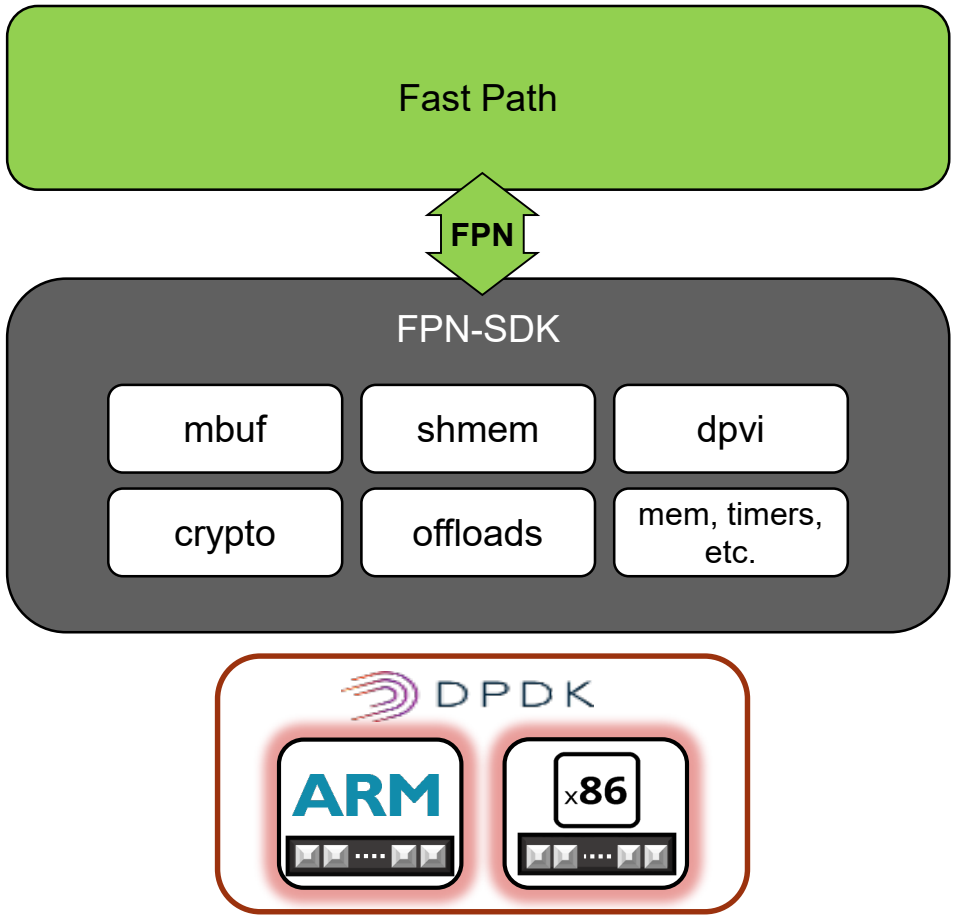
#SPEEDMATTERS For Serious Networks

Fast Path: Main Components



FPN-SDK Packet Processing Library

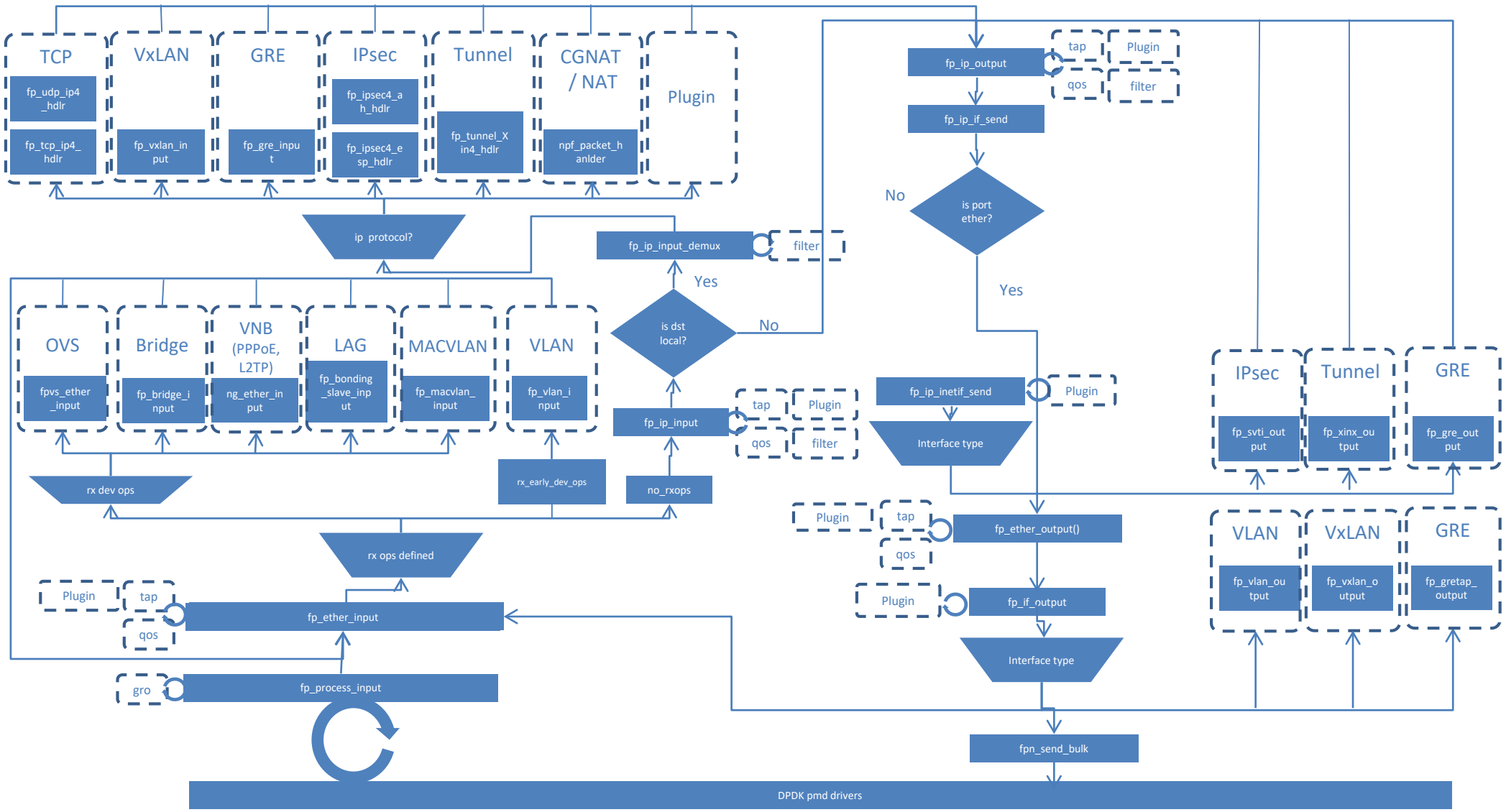
- **Hardware abstraction layer**
- **Northbound: FPN API for Fast Path modules**
  - Packet buffer (mbuf)
  - Shared Memory (userland / kernel / fast path)
  - DPVI
  - Crypto with HW support and SW fallback
  - Offloads: checksum, TCP (LRO, TSO)
  - And more: Control Plane protection, fast and scalable timers, memory pool and ring, lock and synchronization, atomic operations, CPU usage monitoring, function calls tracking for debugging, inter-core packet distribution
- **Southbound: hardware-specific SDK**



Protocol Implementation: Fast Path Modules

- **Generic software using the FPN-SDK generic API**
  - Same code used on supported hardware platforms
- **High performance architecture**
  - Run to completion model with pipeline capabilities when required (QoS…)
  - Implement only simple features to process 99 % of the traffic with maximum efficiency
  - Complex processing for the rest of the traffic is delegated to the Linux Networking Stack
- **Optimized code**
  - Straightforward case optimized (if xxx_likely())
  - Lock-free, prefetch, cache usage
  - Statistics are implemented per core to minimize performance impact
- **Fast algorithms**
  - Route lookup with 8/8/8/8 trie
  - Security Policy Database lookup with automatic linear / trie switching
  - Load balancing of packets can be done (pipeline-hash plugin) if not provided by hardware
- **Use of hardware offloads abstracted by the FPN-SDK API**
  - Packet forwarding / sanity checks
  - Crypto, QoS….

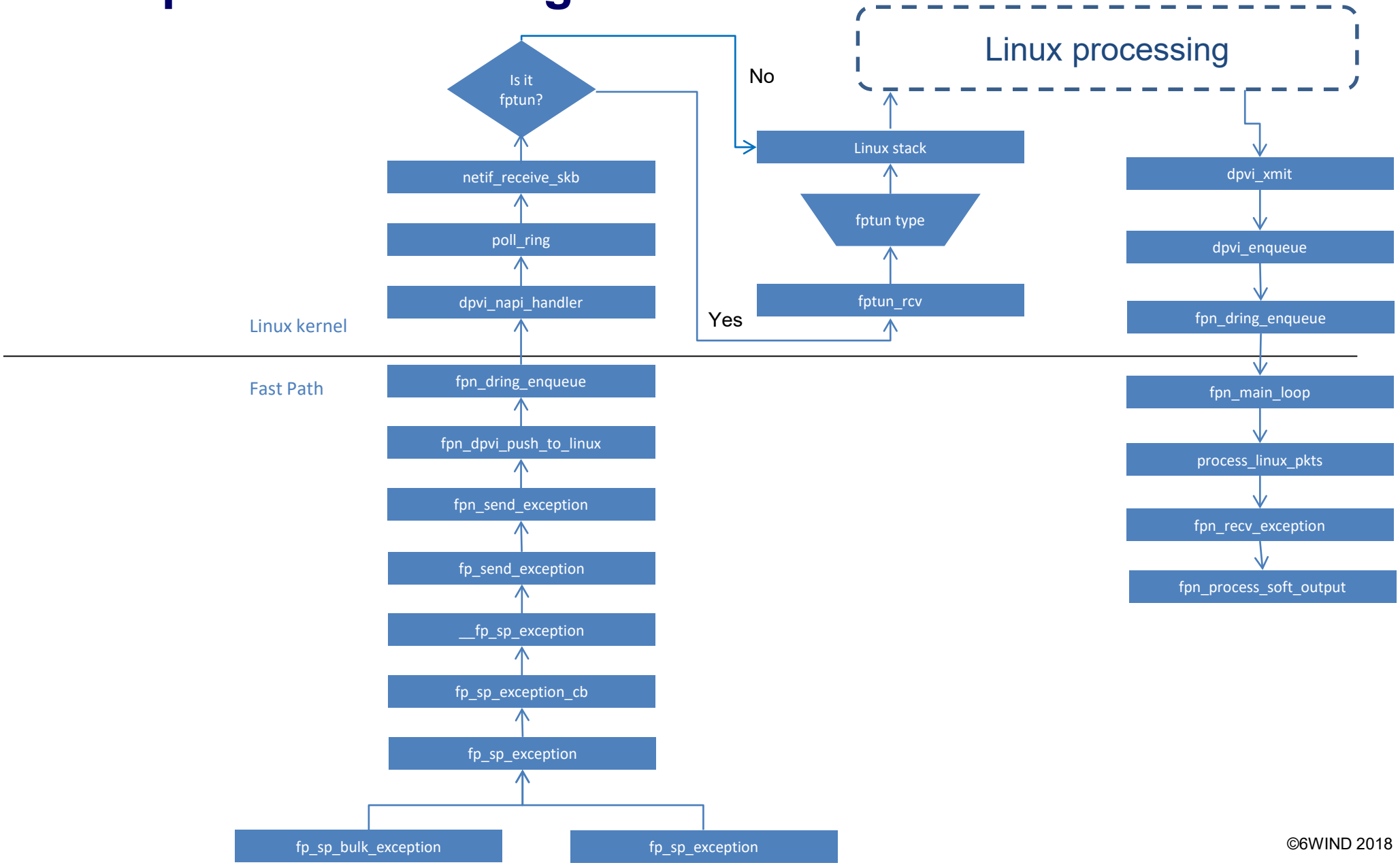
Packet Handling Example



Interface Between Fast Path And Linux Networking Stack: DPVI

- **Data Path Virtual Interface**
  - Makes Fast Path ports appear as netdevices into the Linux Networking Stack
  - Implemented using FPN-SDK
- **Provides NIC representor in Linux for standard control (ethtool and other Linux tools)**
- **Implement exception strategy**
  - Basic exception: standard processing through netif_rx
  - Extended exception for packets that have been preprocessed by the Fast Path: thanks to FPTUN header, packets are injected at the right place into the Linux Networking Stack
- **Used by Linux to send packets through the Fast Path**

Packet Exception Processing



Fast Path Commands: fp-cli

- **For debugging and testing purposes, fp-cli is used to**
  - Set / get parameters of the Fast Path
  - Manage interfaces
  - Display statistics
  - Display debugging information
- **Command output is formatted in json**
- **List of Fast Path commands**
  - Cf. « Command Reference » in Fast Path Baseline documentation
  - Cf. Fast Path Filtering IPv4 and Fast Path Filtering IPv6 documentation
  - Each Fast Path module brings its own commands
  - Run `fp-cli help` for the list of commands

```
root@localhost:~# fp-cli iface
1:lo [VR-0] ifid=1 (virtual) <UP|RUNNING|FWD4|FWD6|MPLS> (0x5b)
        type=loop mac=00:00:00:00:00:00 mtu=0 no numa tcp4mss=0 tcp6mss=0
        blade=1
        IPv4 routes=0  IPv6 routes=0
        if_ops: rx_dev=none rx_early=none tx_dev=none ip_output=none
2:ens3 [VR-0] ifid=2 (virtual) <FWD4|FWD6|MPLS> (0x58)
        type=ether mac=52:54:00:16:c3:ea mtu=1500 no numa tcp4mss=0 tcp6mss=0
        blade=1
        IPv4 routes=0  IPv6 routes=0
        if_ops: rx_dev=none rx_early=none tx_dev=none ip_output=none
5:dp0 [VR-0] ifid=5 (port 0) <UP|RUNNING|FWD4|FWD6|MPLS> (0x5b)
        type=ether mac=52:54:00:d6:4f:25 mtu=1500 numa=0 tcp4mss=0 tcp6mss=0
        blade=1
        IPv4 routes=1  IPv6 routes=0
        if_ops: rx_dev=none rx_early=none tx_dev=none ip_output=none

root@localhost:~# fp-cli iface-json
[
  {
    "ifaces": [
      {
        "port": "virtual",
        "name": "lo",
        "ipv6_routes": 0,
        "ifid": 1,
        "blade": 1,
        "mac": "00:00:00:00:00:00",
        "mtu": 0,
        "type": "loop",
        "vrfid": 0,
        "tcp6_mss": 0,
        "numa": "no numa",
        "ipv4_routes": 0,
(…)
```

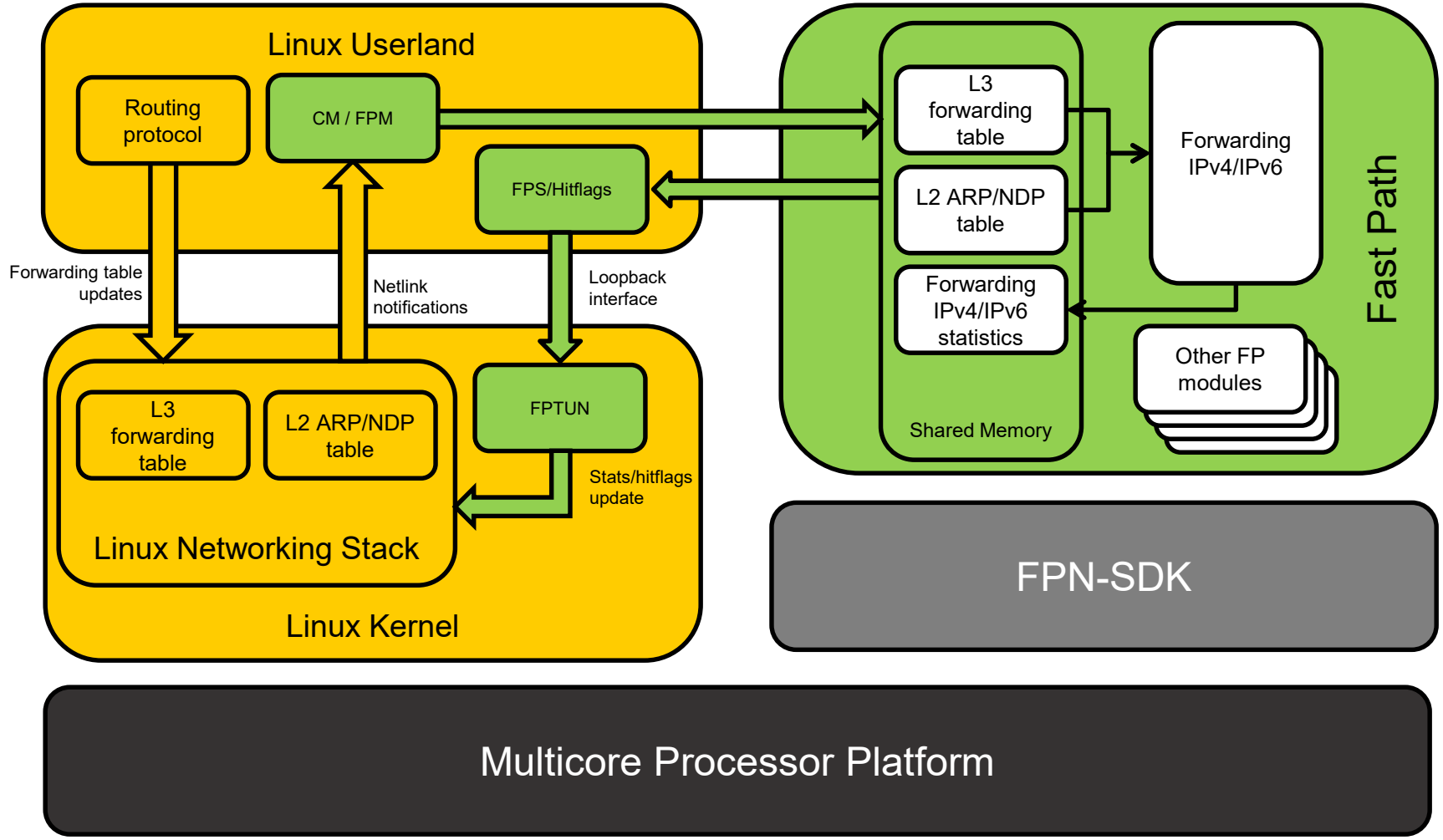
6WINDGate Fast Path Modules

- **FPN-SDK**
  - Baseline
  - Add-on for DPDK (Intel, ARM)
  - Add-on for Cavium

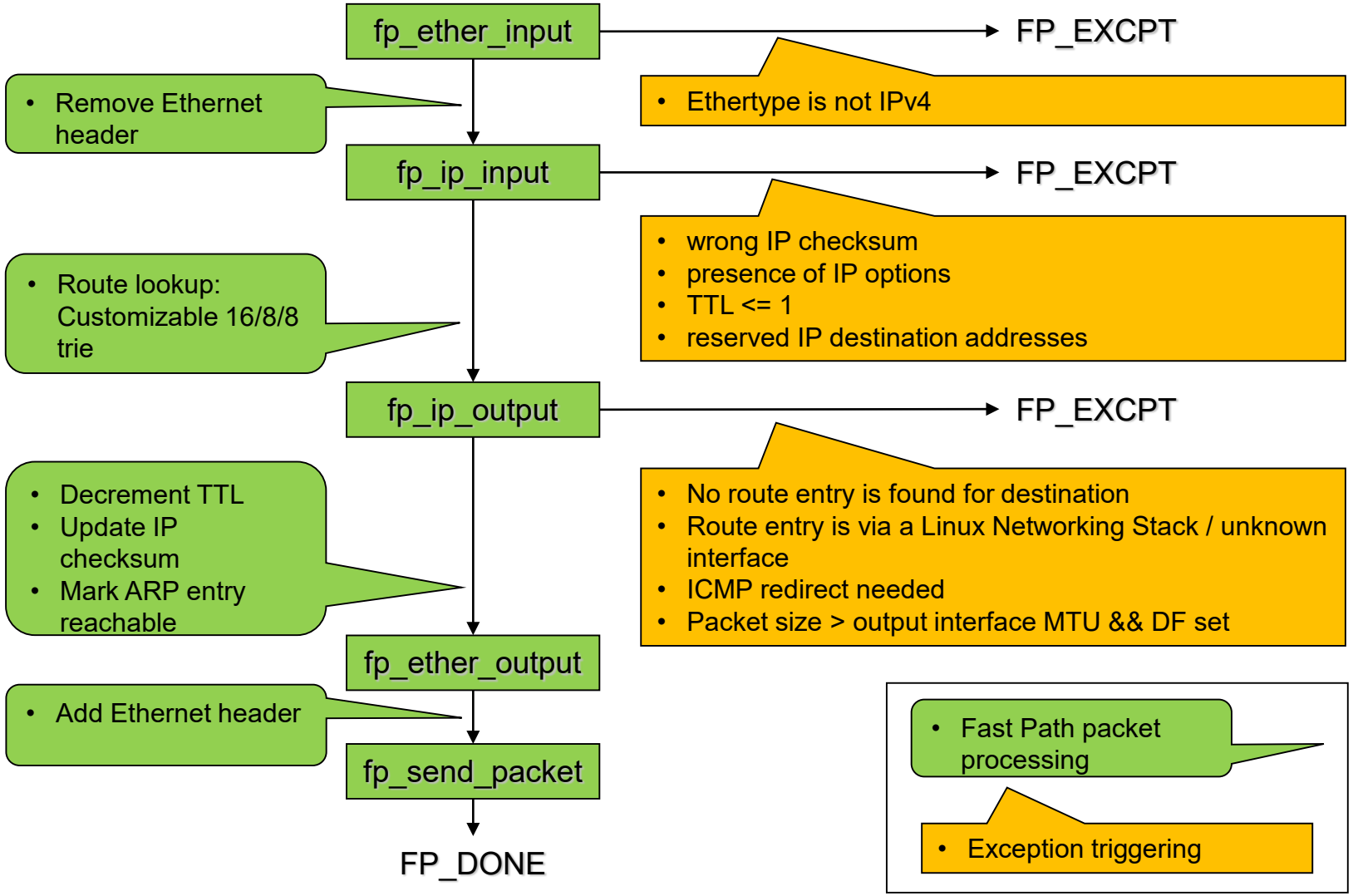
- **Fast Path …**
  - Fast Path Baseline
  - Ethernet Bridge
  - VLAN
  - VXLAN
  - LAG
  - GRE
  - QoS Basic (ex-QoS)
  - QoS Advanced
  - Forwarding IPv4
  - Forwarding IPv6
  - Reassembly IPv4
  - Reassembly IPv6
  - Tunneling (IPinIP)

- **…Fast Path**
  - IPsec IPv4
  - IPsec IPv6
  - IPsec SVTI
  - Filtering Ethernet Bridge
  - Filtering IPv4
  - Filtering IPv6
  - NAT
  - CG-NAT
  - Policy-Based Routing
  - Flow Inspection / Packet Capture
  - Multicast IPv4
  - Multicast IPv6
  - MPLS
  - L2TP client/server
  - PPPoE client/server
  - GTP-U
  - OVS Acceleration

6WINDGate IP Forwarding Example: Architecture



6WINDGate IP Forwarding Example: Call Flow



6WINDGate Logical Interfaces Case: Summary

- Some modules register rx_dev_ops / rx_early_dev_ops / tx_dev_ops / ip_output_ops
- When a logical interface is created, the FPM associates it to a module (in order to link it with rx/tx specific operations)
- It is possible to override fp_ip_input / fp_if_output / fp_ether_input / fp_ether_output / fp_ip_if_send / fp_ip_inetif_send / … functions using hook feature (load balancer case for example, refer plugins section)

Thank You

6WIND.com

6WIND

#SPEEDMATTERS For Serious Networks