i.MX 8 Fast Packet Routing Software

Yannick Vignon

Senior Software Engineer

October 2018 | AMF-AUT-T3369

SECURE CONNECTIONS FOR A SMARTER WORLD

Agenda

- Overview of Fast Packet Processing Solutions
- Implementation Details
- Experimental Results
- Roadmap

Fast Path Routing Solutions on i.MX

Fast Path Routing

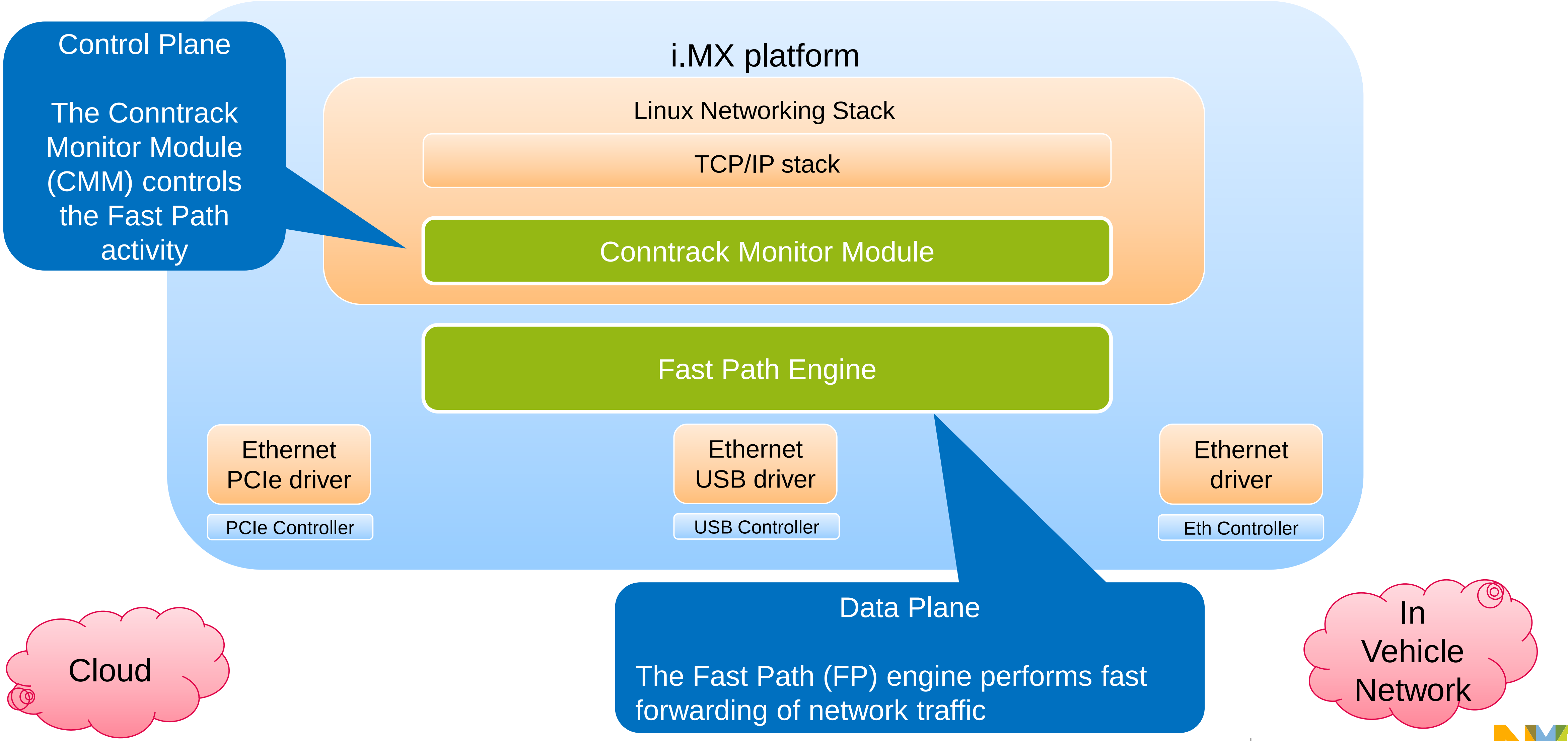
Definition

- A Fast Path (FP) to route/bridge data between 2 network interfaces.
  - Internal FP: same OS as the host networking stack (Slow Path - SP), same or separate core.
  - External FP: separate OS and Core.

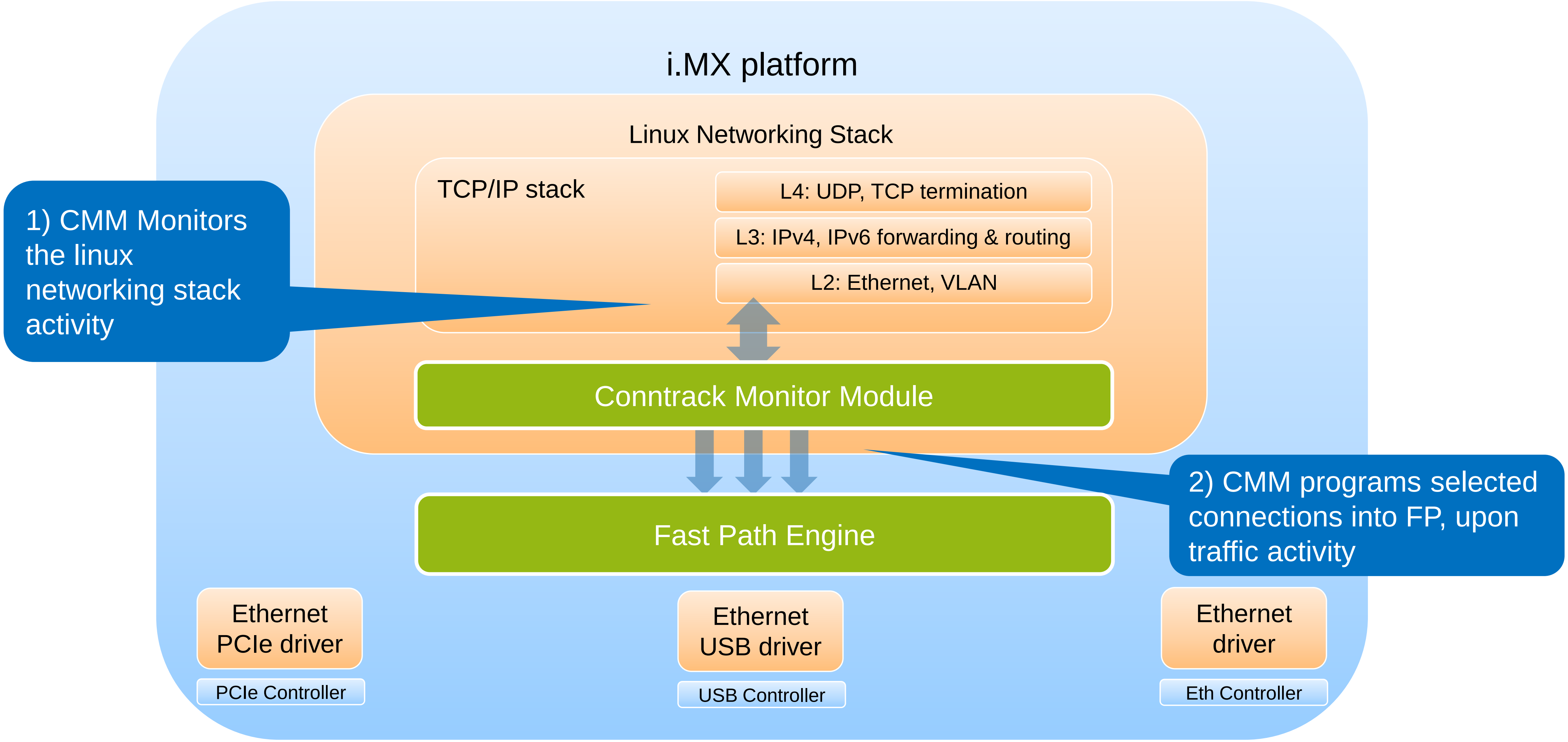
Purpose

- Maximize data routing performance with minimum CPU load requirements to the main processor:
  - Reason: the native data routing performance of the host OS is not optimized:
    - Generic, feature-rich networking stack
    - Does not use the SoC HW efficiently (incomplete utilization of multi-core architecture, local memories…)

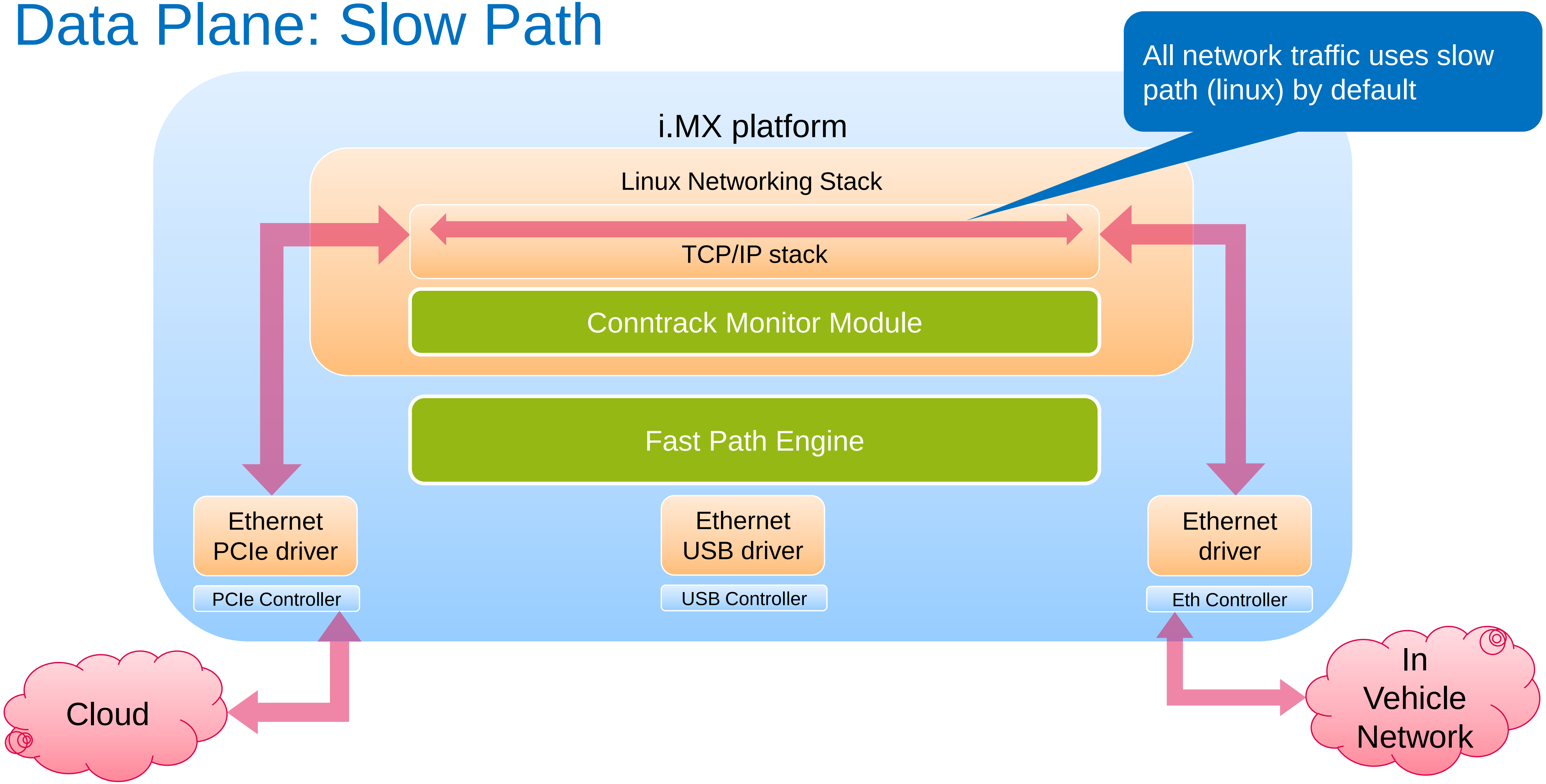
i.MX Linux Fast Path Routing Components



Control Plane: CMM

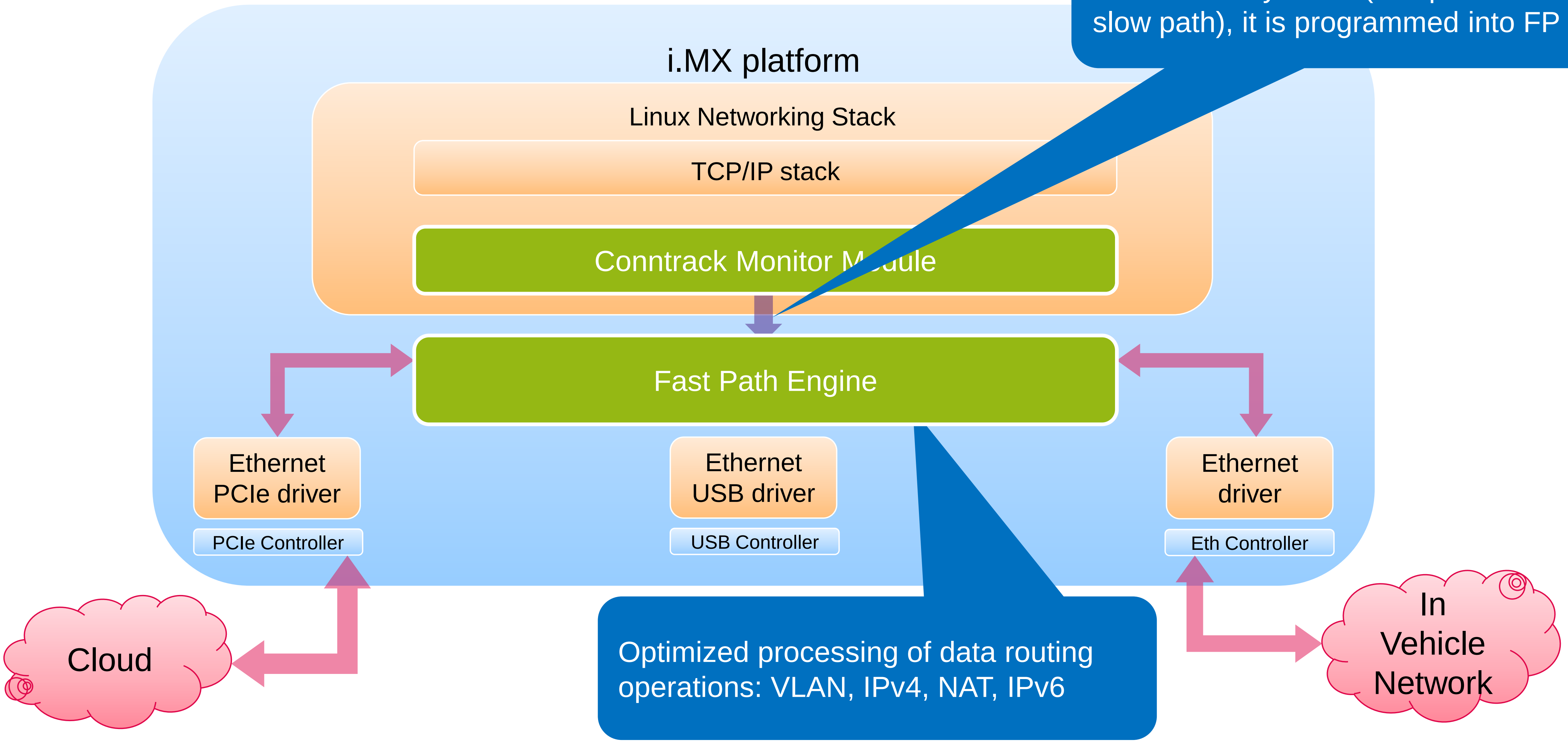


Data Plane: Slow Path



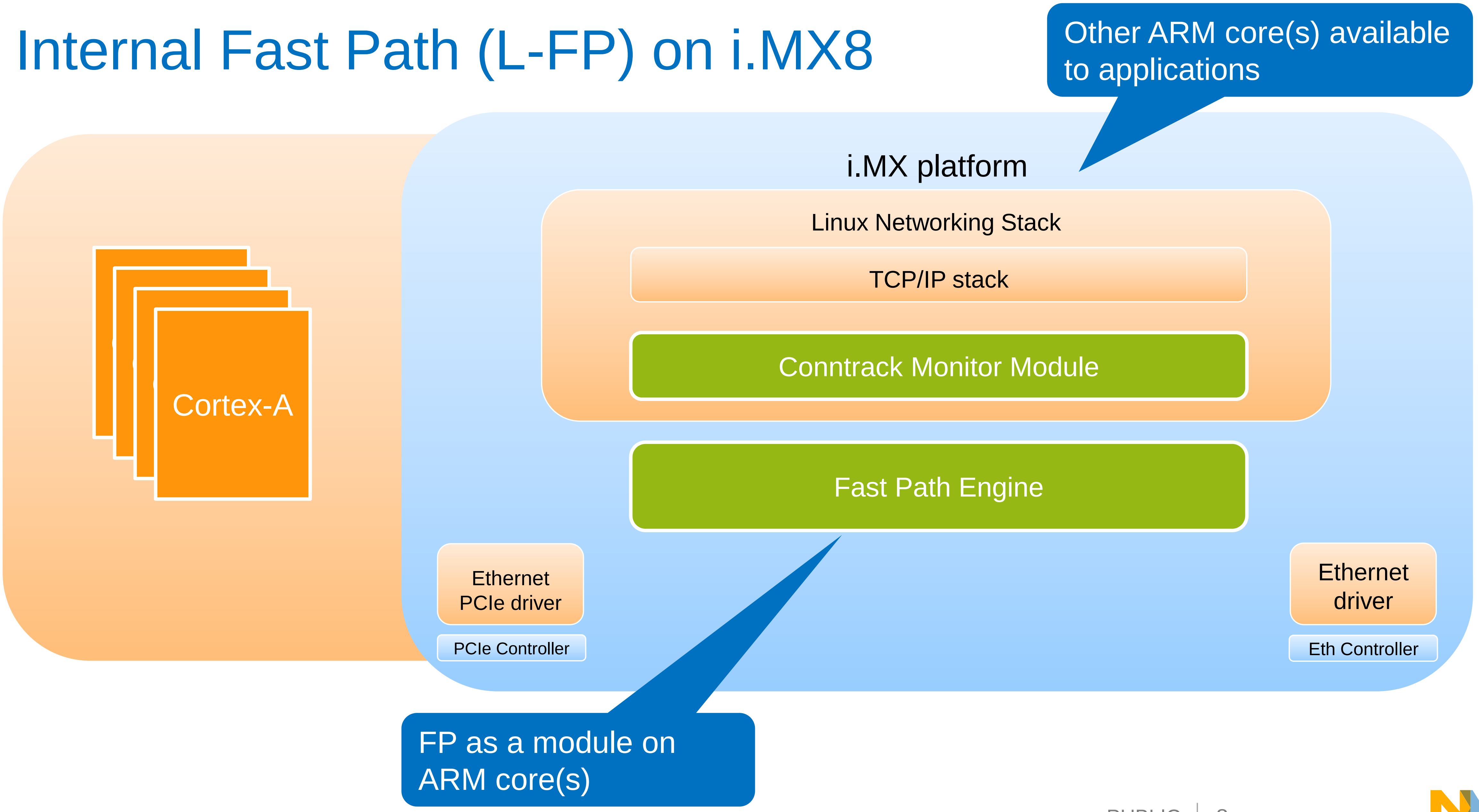
Data Plane: Fast Path (FP)

Once a connection to be fast forwarded is identified by CMM (first packet in slow path), it is programmed into FP

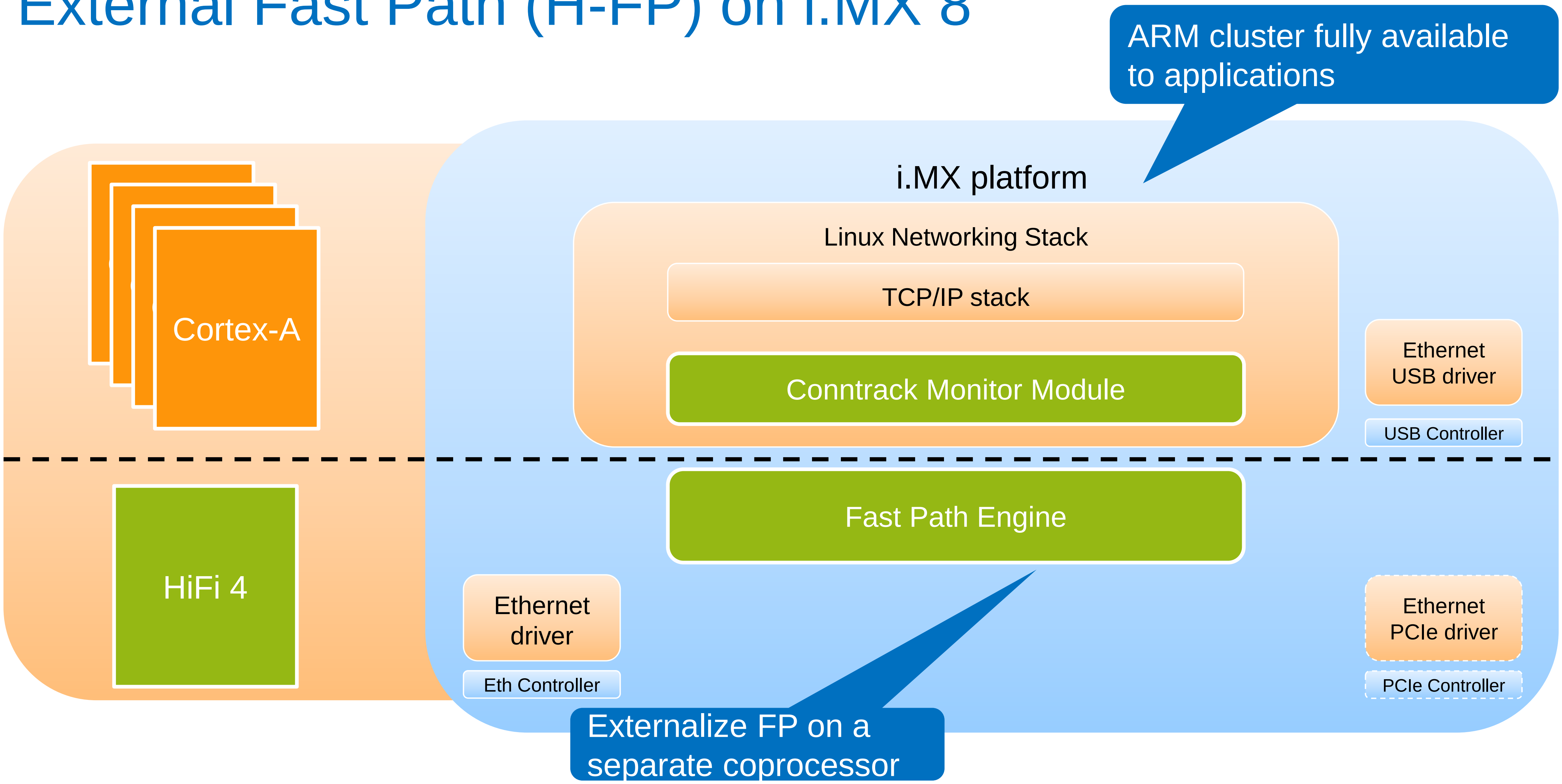


Optimized processing of data routing operations: VLAN, IPv4, NAT, IPv6

Internal Fast Path (L-FP) on i.MX8



External Fast Path (H-FP) on i.MX 8



i.MX Fast Path Routing Highlights

Fast Path Routing

- Fast path processing is performed between 2 network interfaces (WAN/LAN both directions). More than 2 interfaces may be involved in the fast path.
- Smooth integration into the host OS (Linux) networking subsystem: programmed routing rules are propagated by CMM to FP.

Firewall

- By default all incoming traffic is going to slow path (host OS). At that stage firewall rules programmed in the host OS (Linux) are applied to all incoming traffic. CMM considers fast path for a connection which successfully passes filtering rules. Connections rejected by the firewall never reach fast path.

QoS

- L-FP: relies on existing host OS (linux) capability: FP interfaces on egress before the Linux transmit scheduler. Hence it inherits the TC (Traffic Control) and qdisc scheduler features available in Linux.
- H-FP: QoS and scheduler features implemented in FP.

Implementation Details

Linux Fast Path Engine (L-FP)

- Use of ingress Netfilter hook for Rx and device transmit queue for Tx:

→ Support for any Linux network interface

- Implements routing operations for TCP/UDP connections:
  - IPv4 with and without NAT
  - IPv6
  - VLAN tag insertion and removal
- Fixed/pre-configured flow and route entries (no CMM)
- Driver optimizations
  - Memory buffers address < 32bit (avoid remap/memory copy)
  - Reduce data cache maintenance range (CPU only touches 1 to 2 cache lines)
  - Recycle transmitted buffers for receive
  - Burst read of non-cacheable descriptors
- Target platform: i.MX8QuadXPlus (quad core Cortex-A35 @1.2GHz)
- Planned: automated maintenance of flow and route tables through CMM

i.MX 8 HiFi 4 Feature Summary

- Optimized for audio processing, but turns out to be a good fit as a co-processor for a fast-path engine as well

- Relevant features for a fast-path implementation:
  - >600MHz core, capable of running generic software
  - 128 local AXI bus, running at the core frequency
  - Has access to most/all SoC blocks (including ENET)
  - 32kB L1 Instruction cache, 48 kB L1 Data cache
  - Fast local memories:
    - 64kB internal memory (same latency as cache)
    - 448kB On Chip RAM (through AXI bus)
  - Designed for efficient data transfers (prefetch instructions, large cache lines)

HiFi 4 Fast Path Engine (Current)

- Proof-of-concept only
- Bare-metal implementation, supporting only the ENET (i.MX8 internal Ethernet MAC controller)
- Co-exists with Linux using the ENET Frame Parser and multiple queues:
  - On Rx, packets with VLAN 255 are sent to the HiFi 4 queue, other packets go to Linux
  - On Tx, round-robin is used between the queues, with one queue for the HiFi 4 and one for Linux
- A tight loop:
  - Polls each interface
  - Takes any incoming packets one by one, and copies data from Rx to Tx buffers, without any processing (no flow matching, no NAT/VLAN modifications)

→ Approximate model of performance when using average packet sizes

HiFi 4 Fast Path Engine (Planned)

- Direct control of both ENET interfaces, possible support of other interface types (PCIe, USB) through professional support services
- Routing operations for TCP/UDP connections:
  - IPv4 with and without NAT
  - IPv6
  - VLAN tag insertion and removal
- Automated maintenance of flow and route tables through CMM
- Bare metal environment for full control over the CPU
- High efficiency firmware:
  - Packet batching to improve I-cache usage
  - Maximize D-cache usage and internal memory usage
  - Zero copy
  - Interrupt mitigation

Experimental Results

i.MX 8 Performance Measurements

- 2 test "campaigns" performed:
  - Initial one based on iperf, to showcase performance difference between Linux standard stack and Linux Fast-path
  - Recent one based on T-Rex traffic generator: better automation potential and flexibility (varying packet sizes, etc), higher packet generation performance. Useful to evaluate HiFi 4 Fast Path performance

- Test results of the 2 campaigns are consistent: iperf setup likely to be phased out

- All measurements were made on an i.MX8QuadXPlus MEK board, using 1000BaseT connections

Iperf Test Results

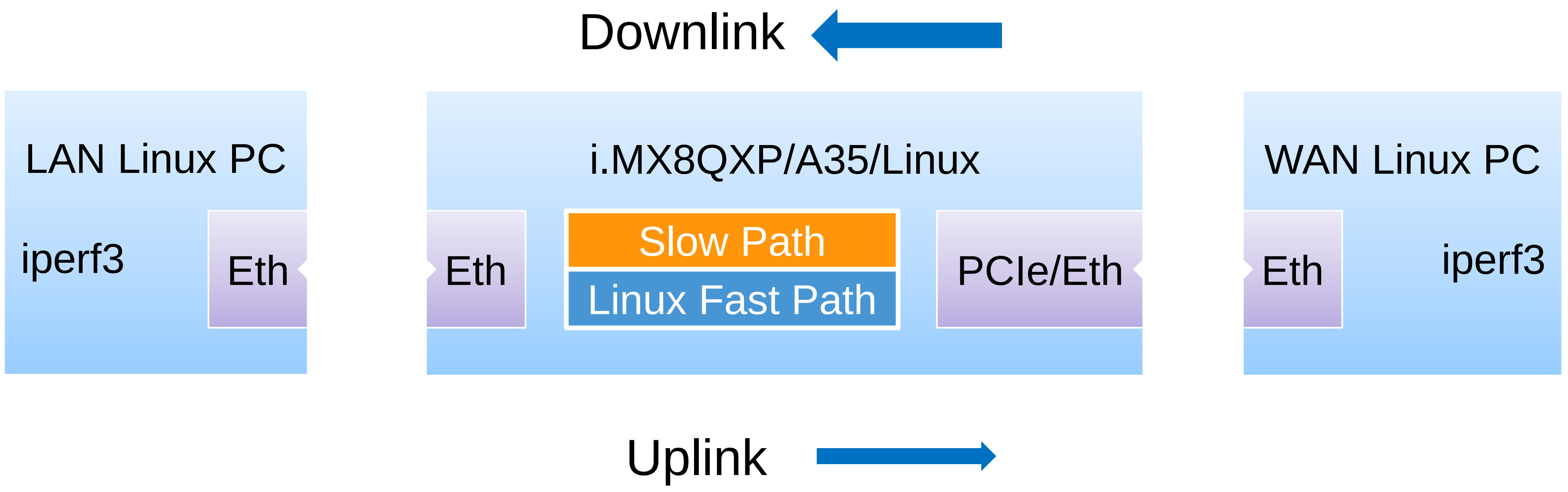
i.MX8 Performance Measurements With iperf

- Captured metrics:
  - CPU load, Network bandwidth

- IP routing between Ethernet and PCIe Ethernet (Intel Pro1000):
  - With and without VLAN tag insertion/removal
  - With and without NAT
  - For IPv4/IPv6, TCP/UDP protocols

- Note: iperf only reports the layer 4 bandwidth (see next slides), and uses large packets (typically 1500 bytes for IPv4)

Iperf Test Setup for Performance Analysis



- "Idle" tool used for CPU load measurements
  - CPU load determined from variation in time taken to execute fixed count busy loop
  - CPU load given in percentage of one core

Theoretical Bandwidth

- Ethernet frame
  - –IFG = 12, SFD = 8
  - –ETH = 14, VLAN = 4
  - –MTU = 1500
  - –FCS = 4

- Protocol headers:
  - –IPv4 (no options) = 20
  - –IPv6 (no ext. headers) = 40
  - –UDP = 8
  - –TCP (no options) = 20

Encapsulation	Total Bytes per Frame	Max Packet Rate (pps)	Max L4 Payload (bytes)	L4 Bandwidth (Mbps)
IPv4/UDP	1538	81274.4	1472	957.1
IPv4/TCP	1538	81274.4	1460	949.3
IPv6/UDP	1538	81274.4	1452	944.1
IPV6/TCP	1538	81274.4	1440	936.3
VLAN/IPv4/UDP	1542	81063.6	1472	954.6
VLAN/IPv4/TCP	1542	81063.6	1460	946.8
VLAN/IPv6/UDP	1542	81063.6	1452	941.6
VLAN/IPV6/TCP	1542	81063.6	1440	933.9

IP Routing Measurements (i.MX 8QXP Beta Linux BSP)

- No Fast Path / optimizations
- VLAN on LAN interface
- NAT on WAN interface

Receive	Transmit	L4 Bandwidth (Mbps)	CPU Load A35 (%)
VLAN/IPv4/UDP	PCIe/IPv4/UDP	439	100
VLAN/IPv4/TCP	PCIe/IPv4/TCP	295	100
VLAN/IPv6/UDP	PCIe/IPv6/UDP	565	100
VLAN/IPv6/TCP	PCIe/IPv6/TCP	374	100
PCIe/IPv4/UDP	VLAN/IPv4/UDP	475	100
PCIe/IPv4/TCP	VLAN/IPv4/TCP	399	100
PCIe/IPv6/UDP	VLAN/IPv6/UDP	587	100
PCIe/IPv6/TCP	VLAN/IPv6/TCP	486	100

Fast Path Enabled IP Routing Measurements

- Fast Path / optimizations
- No VLAN / No NAT
- CPU load variations as expected
- Line rate attained for all cases

Receive	Transmit	L4 Bandwidth (Mbps)	CPU Load A35 (%)
IPv4/UDP	PCIe/IPv4/UDP	949	48.8
IPv4/TCP	PCIe/IPv4/TCP	948	52.7
IPv6/UDP	PCIe/IPv6/UDP	936	49.7
IPv6/TCP	PCIe/IPv6/TCP	935	55.3
PCIe/IPv4/UDP	IPv4/UDP	952	48.9
PCIe/IPv4/TCP	IPv4/TCP	947	50.6
PCIe/IPv6/UDP	IPv6/UDP	939	49.7
PCIe/IPv6/TCP	IPv6/TCP	934	52.7

Fast Path Enabled IP Routing Measurements

- Fast Path / optimizations
- VLAN on LAN interface
- NAT on WAN interface
- CPU load variations as expected
- Line rate attained for all cases

Receive	Transmit	L4 Bandwidth (Mbps)	CPU Load A35 (%)
VLAN/IPv4/UDP	PCIe/IPv4/UDP	949	50.5
VLAN/IPv4/TCP	PCIe/IPv4/TCP	945	56.6
VLAN/IPv6/UDP	PCIe/IPv6/UDP	936	51.2
VLAN/IPv6/TCP	PCIe/IPv6/TCP	932	58.1
PCIe/IPv4/UDP	VLAN/IPv4/UDP	951	46.5
PCIe/IPv4/TCP	VLAN/IPv4/TCP	946	50.2
PCIe/IPv6/UDP	VLAN/IPv6/UDP	938	48.6
PCIe/IPv6/TCP	VLAN/IPv6/TCP	933	51.8

Fast Path Enabled Measurements (Full Duplex Traffic)

- Fast Path/optimizations
- Vlan on LAN interface
- NAT on WAN interface
- Uplink 1Gbps (Eth -> PCIe)
- Downlink 1Gbps (PCIe -> Eth)
- CPU load variations as expected
- Line rate for UDP traffic

Traffic type	Uplink L4 Bandwidth (Mbps)	Downlink L4 Bandwidth (Mbps)	CPU Load A35 (%)
IPv4/UDP	951	954	100
IPv4/TCP	939	820	100
IPv6/UDP	938	936	100
IPv6/TCP	843	804	100

T-Rex Test Results

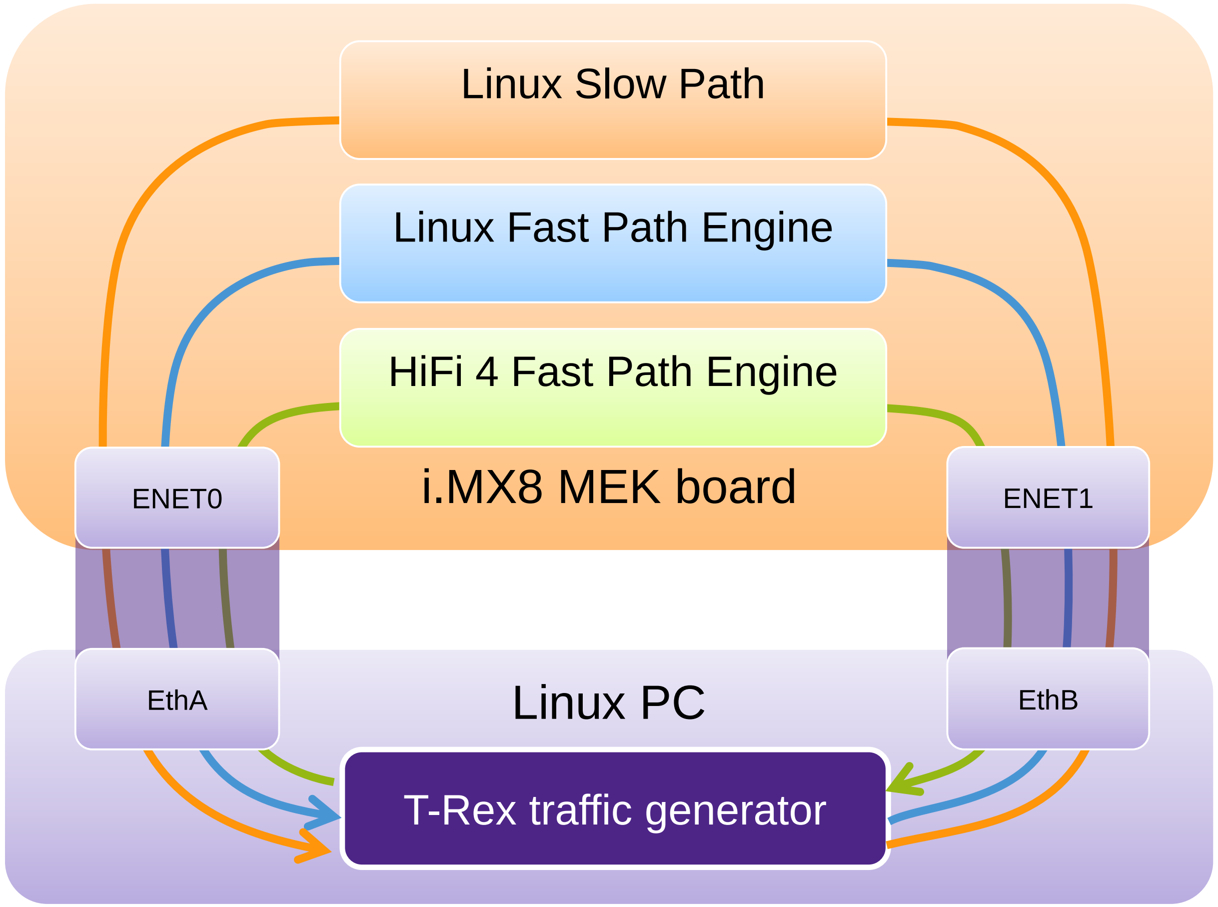
i.MX8 Performance Measurements With T-Rex

- Captured metrics:
  - CPU load, Network bandwidth (at Layer 2)
  - Varying packet sizes

- Measure packet forwarding performance between 2 ENET interfaces (no PCIe)

- Note: Linux slow-path performance benefits from Linux fast-path generic driver optimizations

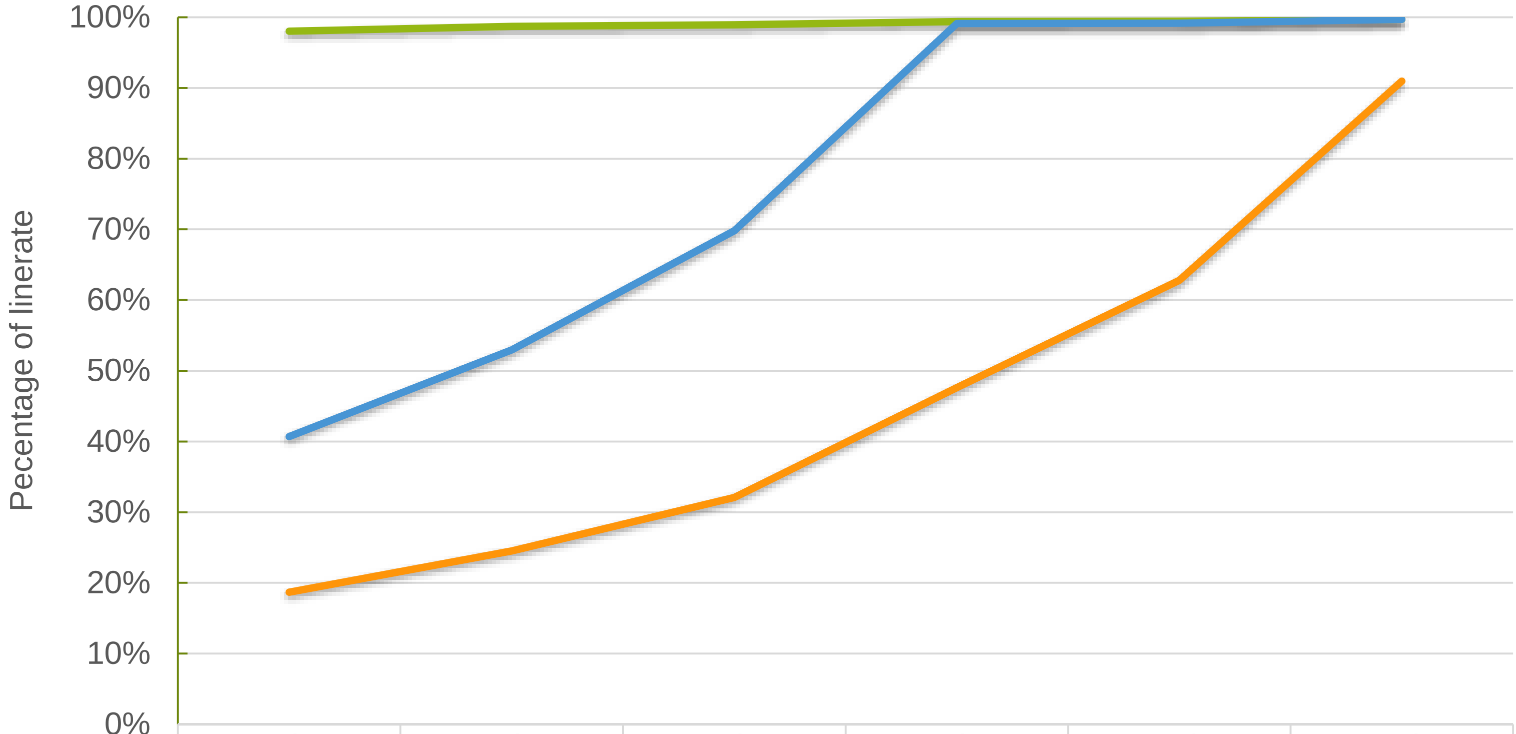
T-Rex Test Setup for Performance Measurements



- Linux slow-path flow:
  - IPv4/UDP routing with NAT, VLAN tag addition
- Linux fast-path flow:
  - IPv4/UDP routing with NAT, VLAN tag addition
- HiFi 4 fast-path flow:
  - Straight buffer copy

Performance Results

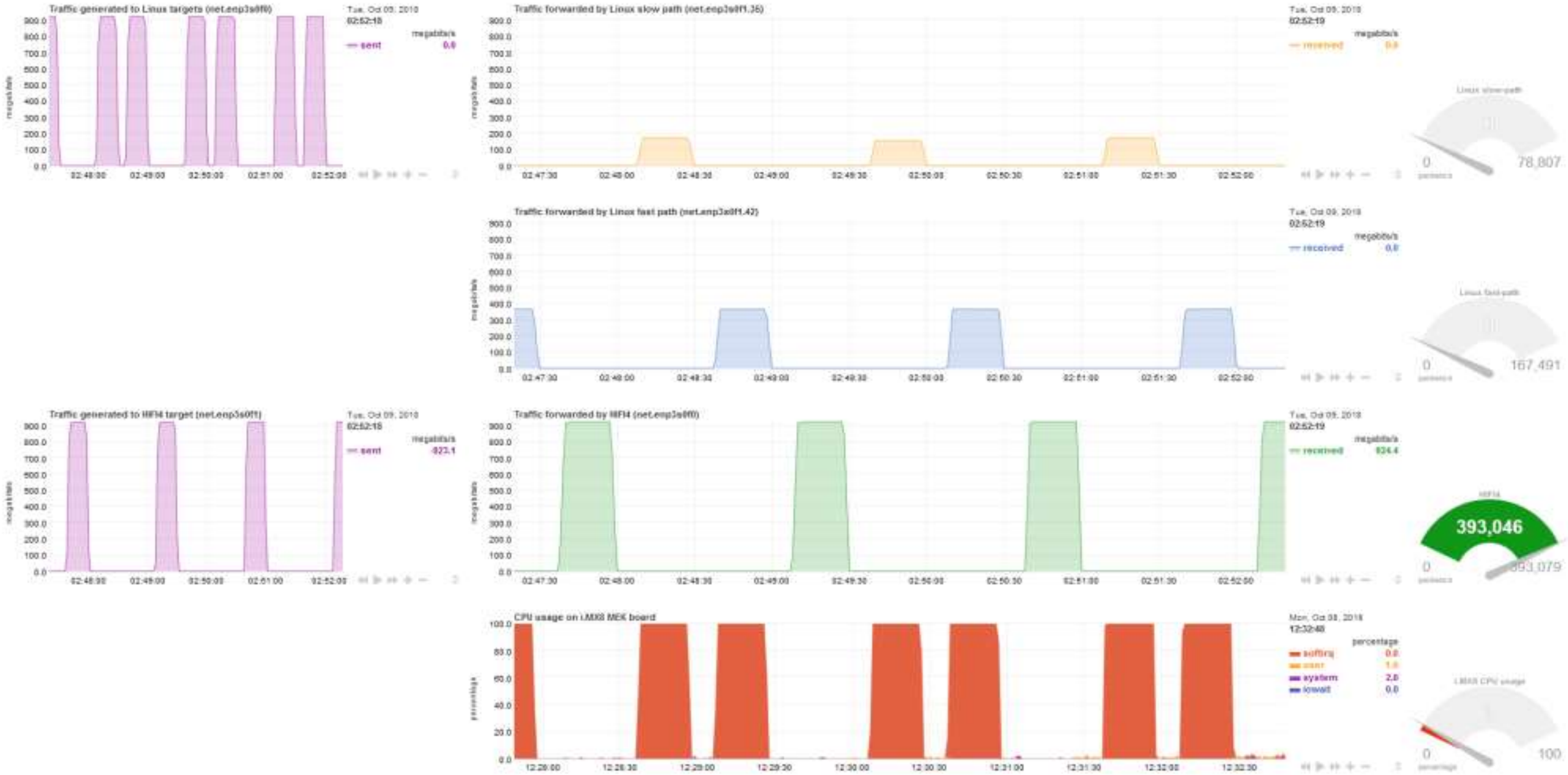
Throughput for varying packet sizes



	288-byte	384-byte	512-byte	768-byte	1024-byte	1500-byte
HiFi 4 Fast path	98.05%	98.71%	98.94%	99.41%	99.42%	99.69%
Linux Fast path	40.68%	52.97%	69.80%	99.11%	99.15%	99.72%
Slow path	18.67%	24.51%	32.09%	47.61%	62.80%	90.97%

Maximum packet rates	
HiFi 4 Fast path	393kpps
Linux Fast path	166kpps
Linux Slow path	78kpps

Live Demo Screenshot

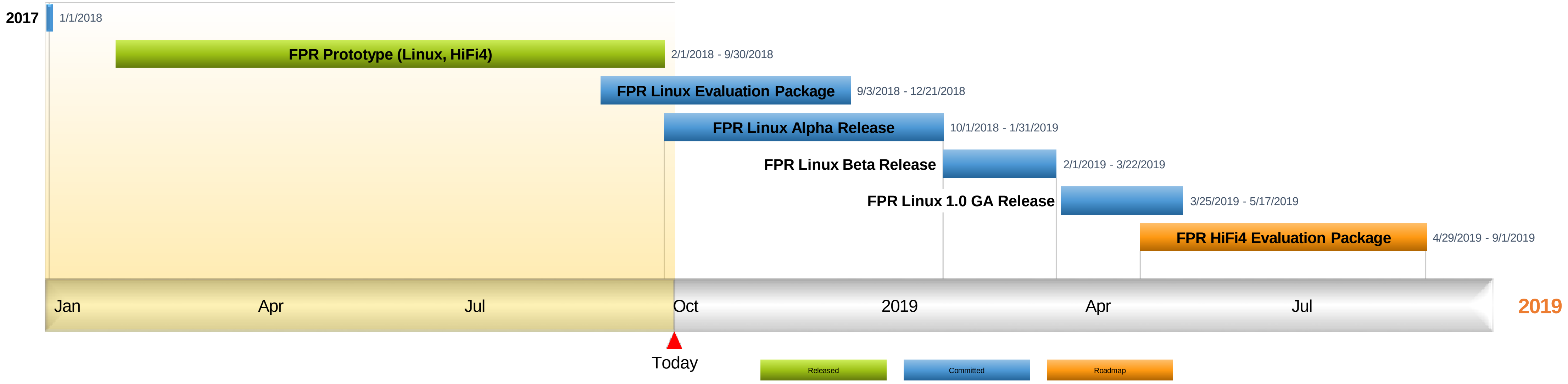


Roadmap

Fast Path Routing Roadmap

FPR

- High Performance Data Routing for UDP/TCP, IPv4/NAT, IPv6, VLAN traffic
- Telematics Box applications
- Linux FPR on i.MX families
  - Network interfaces Eth MAC, PCIe Wifi/5G, Eth USB
- HiFi4 FPR on i.MX8, i.MX8X families



Session Summary

- A Fast path engine can yield significant performance improvements for network packet processing on i.MX8:
  - Linux Fast Path: 2x or more, using any Linux network interface.
  - HiFi 4 Fast Path: 5x expected, while keeping the Cortex-A cluster completely free. (increase will be less if the HiFi 4 does not control both interfaces)

- Maximum packet rate is a (very) good indicator of a system's packet forwarding performance: packet size/bandwidth has very little impact on packet rate.

SECURE CONNECTIONS
FOR A SMARTER WORLD

www.nxp.com