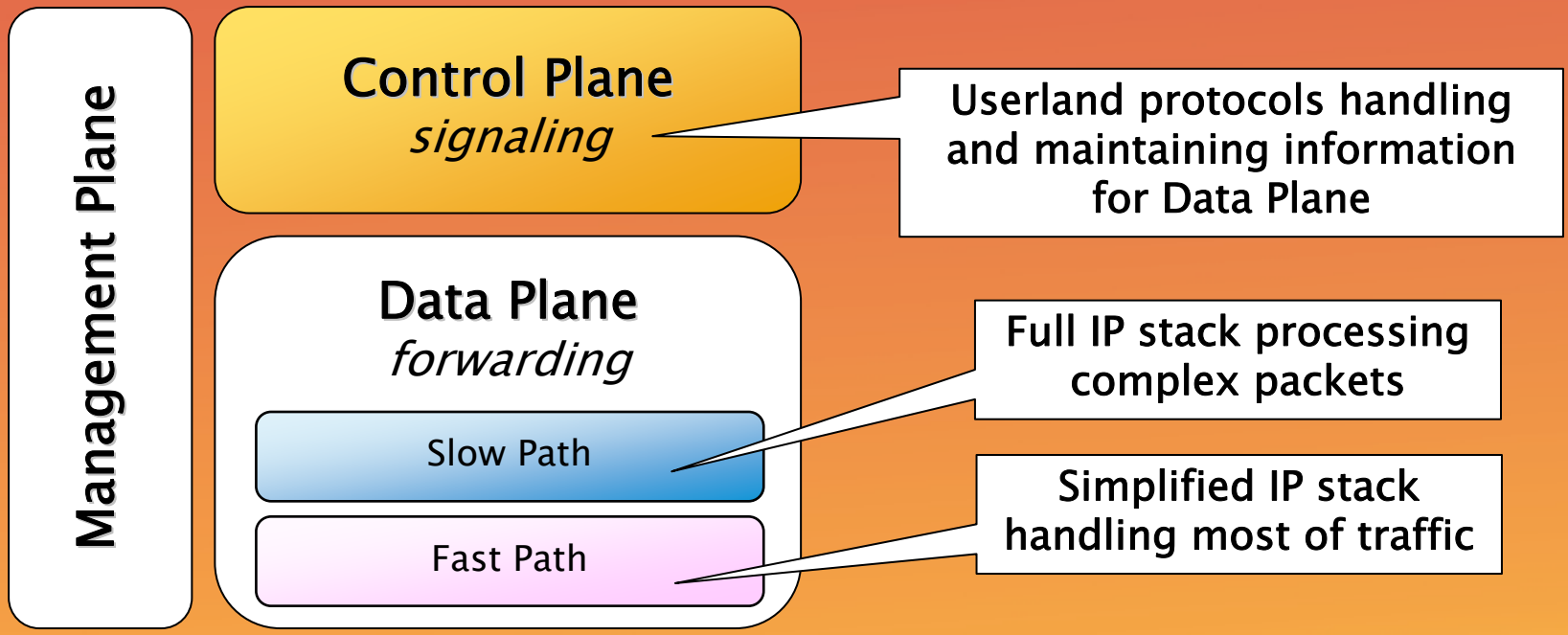
Software Design for Multicore-Based Telecommunications and Networking Equipment

Eric CARMES – 6WIND CEO
eric.carmes@6wind.com

Multicore Design Challenges

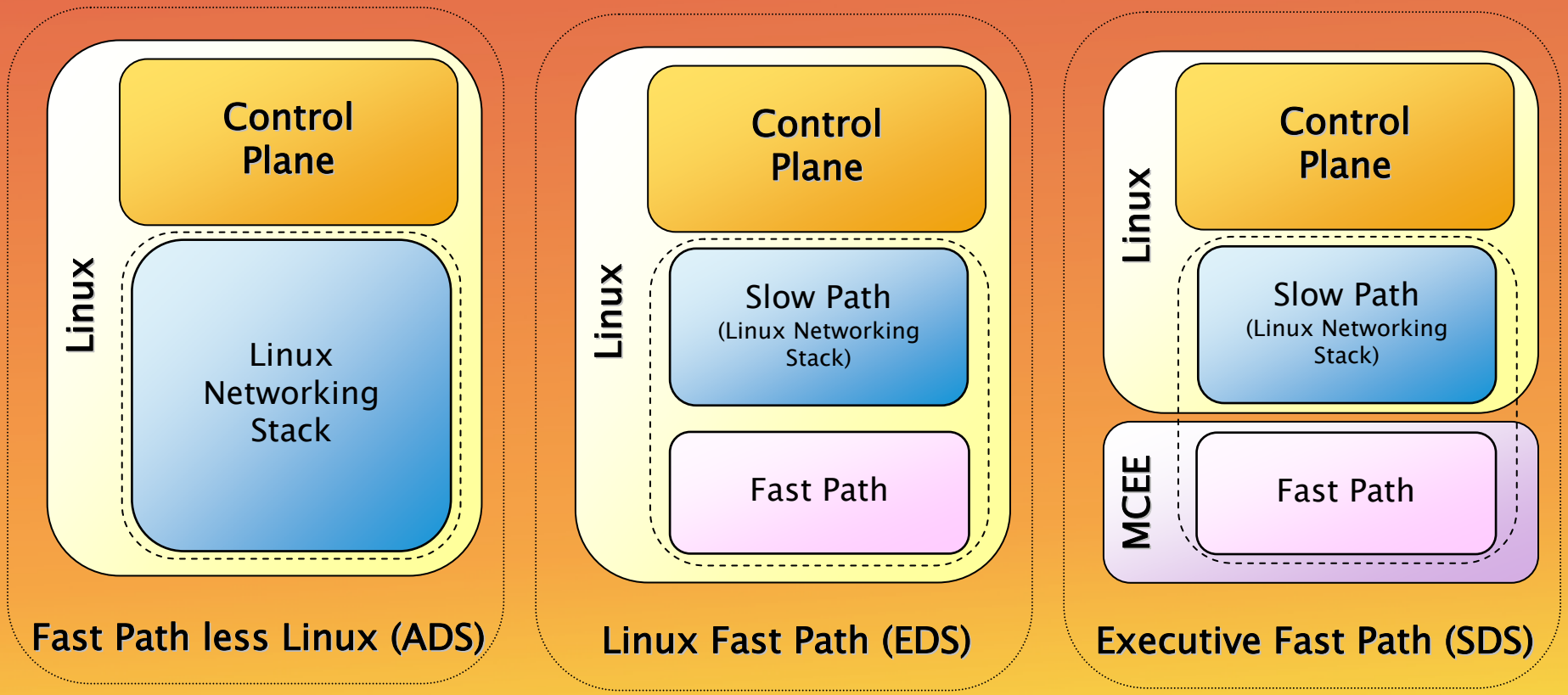
- Linux-based vs. executive-based software architectures
- Ability to Always Maximize Performance
- Distribution of software (Control Plane, Slow Path and Fast Path) over cores
- How to make multicore transparent for applications
- Architecture extensibility over multi-chip and multi-board hardware architectures

Network Equipment Architecture



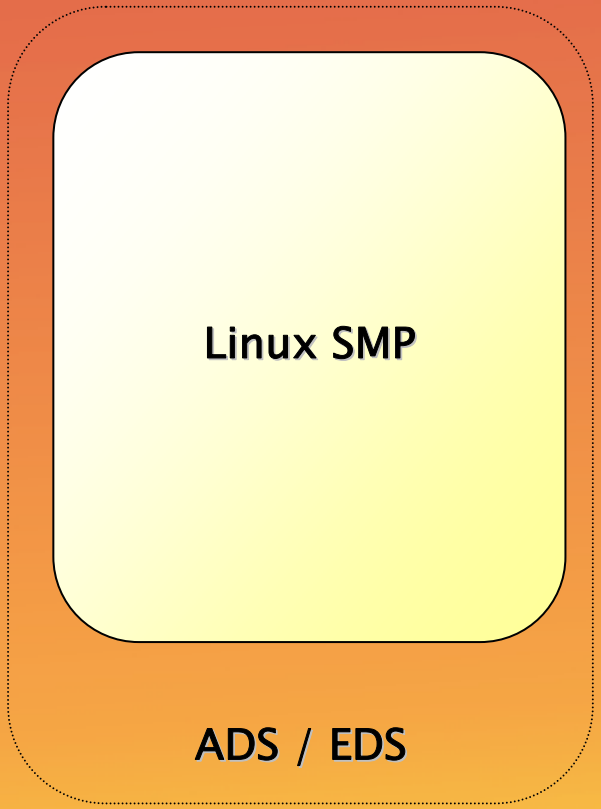
- **Control Plane and Data Plane can be co-localized or distributed over multiple processors**
- **High performance is achieved by running Fast Path on dedicated hardware**

Architecture Options



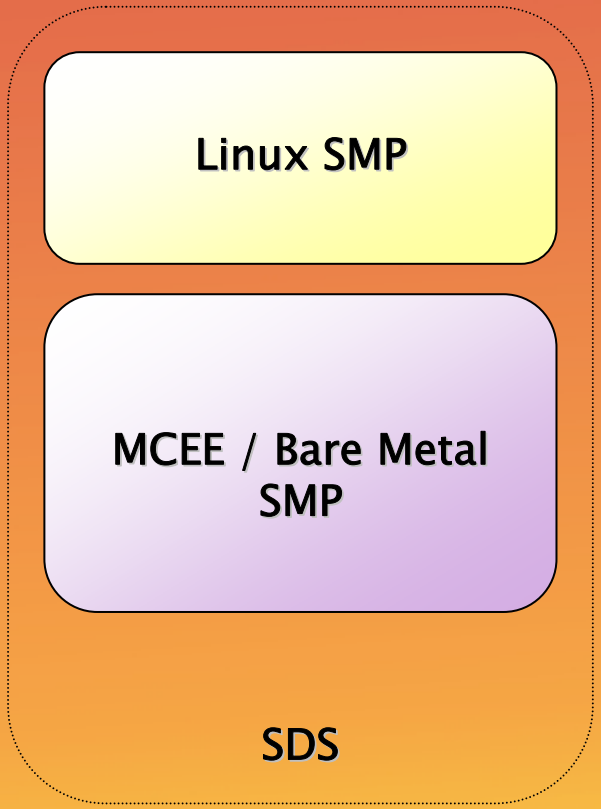
MCEE: Multicore Execution Environment (Bare Metal)

AMP or SMP?



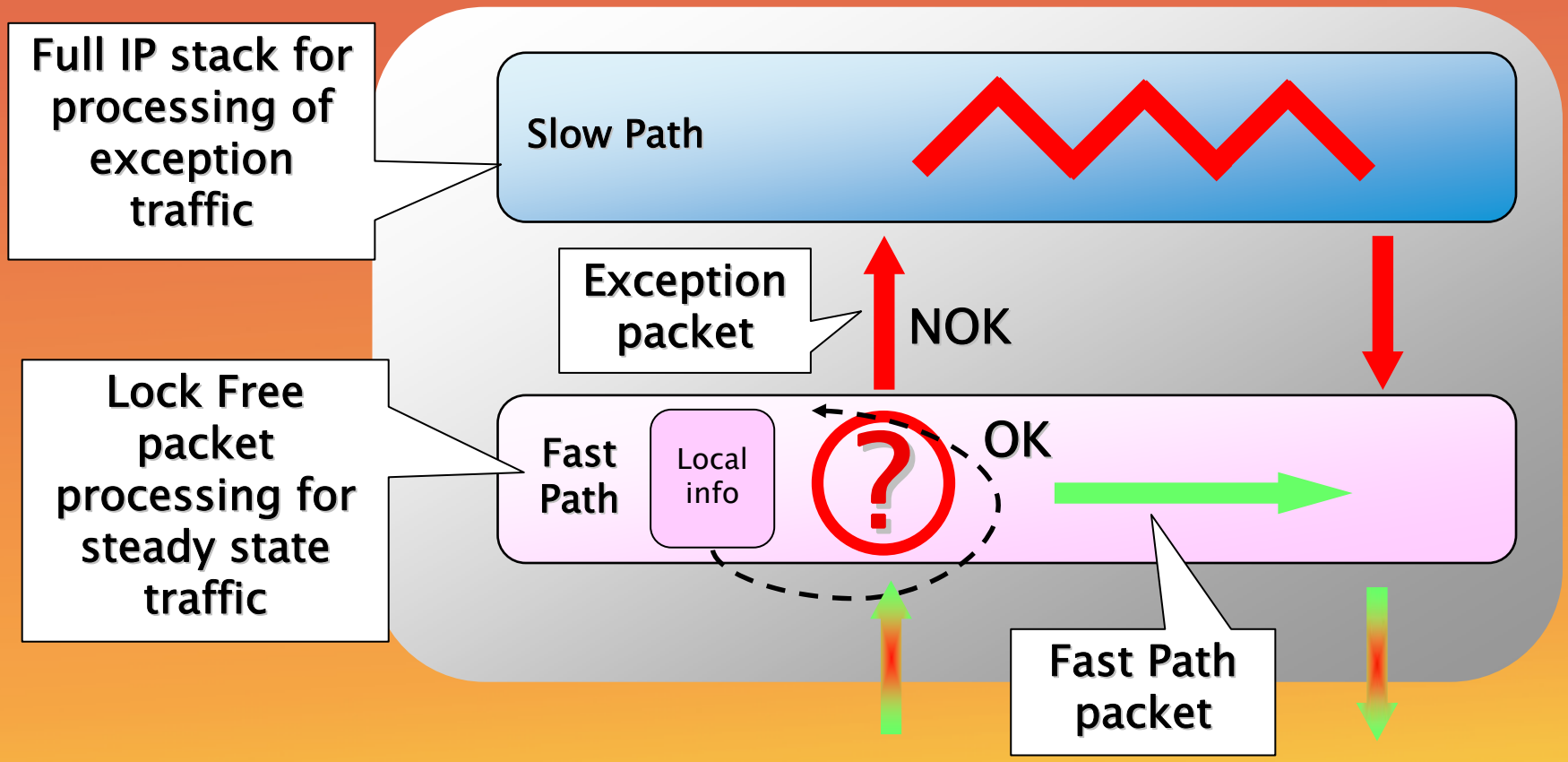
- For ADS & EDS, Linux is used in SMP mode on all cores

AMP or SMP?



- For SDS, core allocation for Linux and Bare Metal is done at boot time
- In each environment, software run in SMP mode

Fast Path Architecture



- **Progressive approach for implementing Fast Path functions**
- **Step by step feature migration in Fast Path**

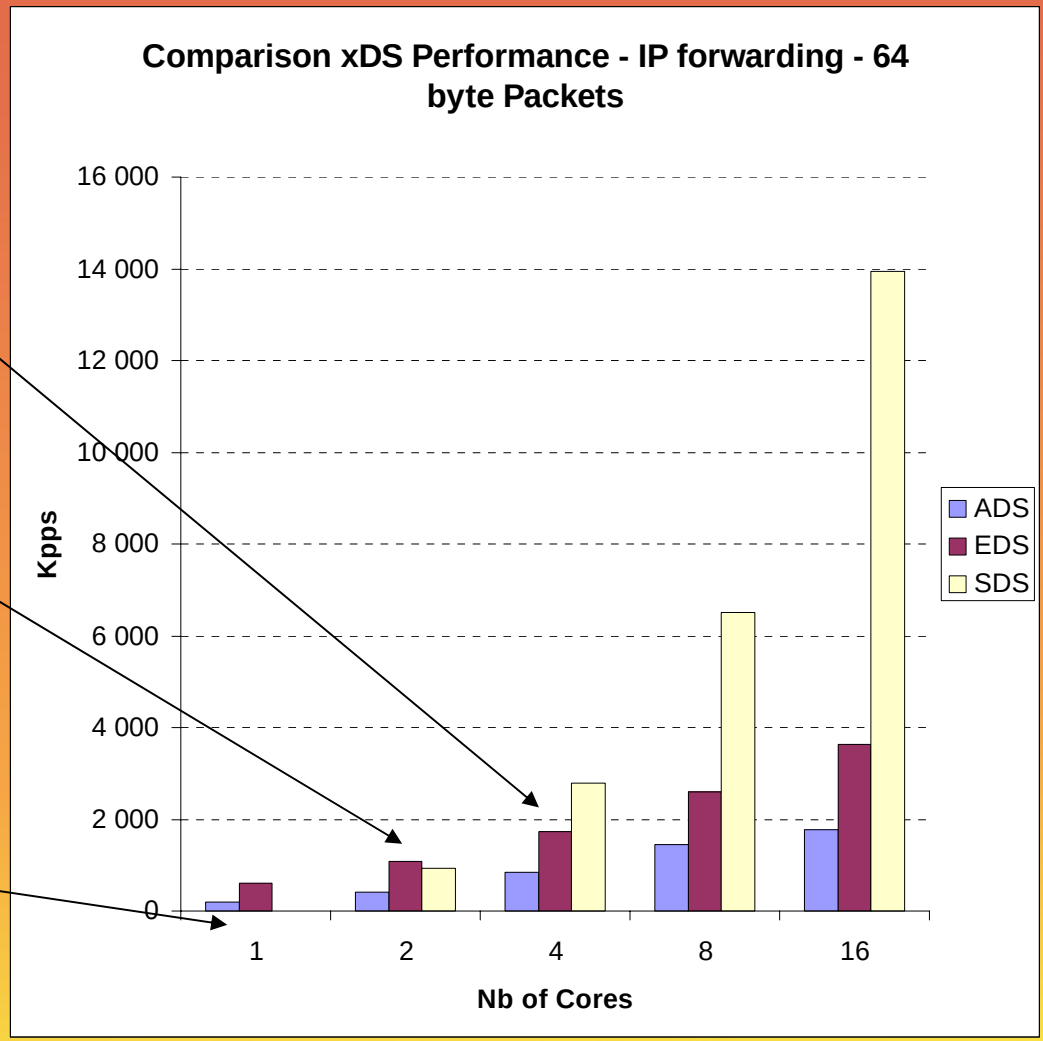
Performance Comparison

SDS is more efficient for 4 cores and more

EDS & SDS performance are similar for 2 cores configuration

EDS far better than ADS

SDS cannot work on 1 core (1 core: Linux)



Based on Cavium Octeon 38xx

Portability on different platforms

- Portable run-to-completion model for Fast Path packet processing
- Some Fast Path networking features depend on hardware features:
  - Crypto engines: Standard API are recommended (OCF) to interface both synchronous and asynchronous engines
  - QoS: This feature requires some platform adaptations to make the best use of available hardware (HW queues, HW coded algorithms...)
- Some parts of the Fast Path have to be optimized according to the hardware platform
  - Future generation of multicore will implement more dedicated accelerators

What does transparency mean (examples)?

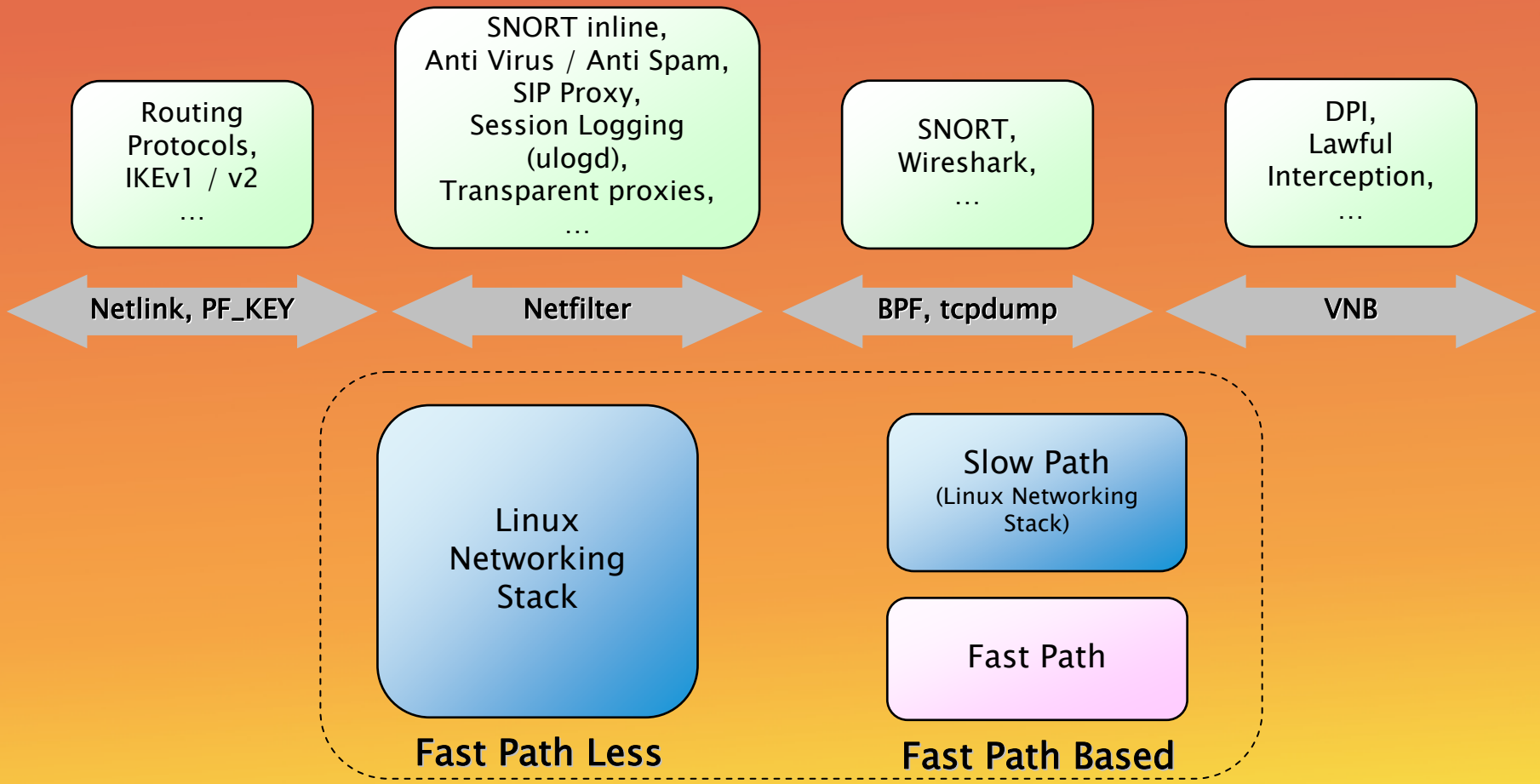
- An existing routing protocol shall run on multicore architectures without any modification
- Statistics shall transparently gather Fast Path & Slow Path information
- Networking stack shall not make the difference between a packet coming from an Ethernet driver or a complex Fast Path
- Tools such as tcpdump shall work...

Openness and Software Reuse

- Use and preserve standard APIs/Interfaces
  - Linux interfaces
    - NETLINK, XFRM, Netfilter/iptables, PF_KEY, /proc, setsockopt()
  - Hardware crypto (OCF)
    - IPsec, IKE, OpenSSL engines
  - Netfilter and tcpdump compatible API in Fast Path

➔ Existing software can be fully REUSED with Fast Path architectures

Examples of Software Reuse

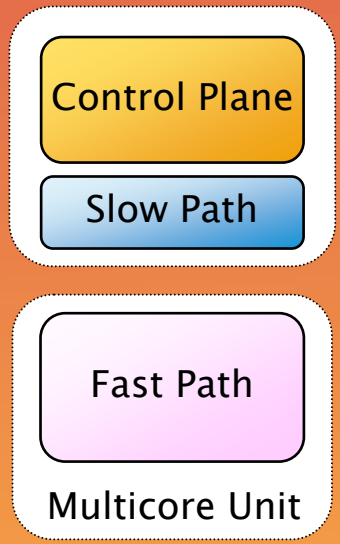


Extensions for Distributed Architectures

- Multicore technology is by essence scalable and can be used to design multi-chip architectures

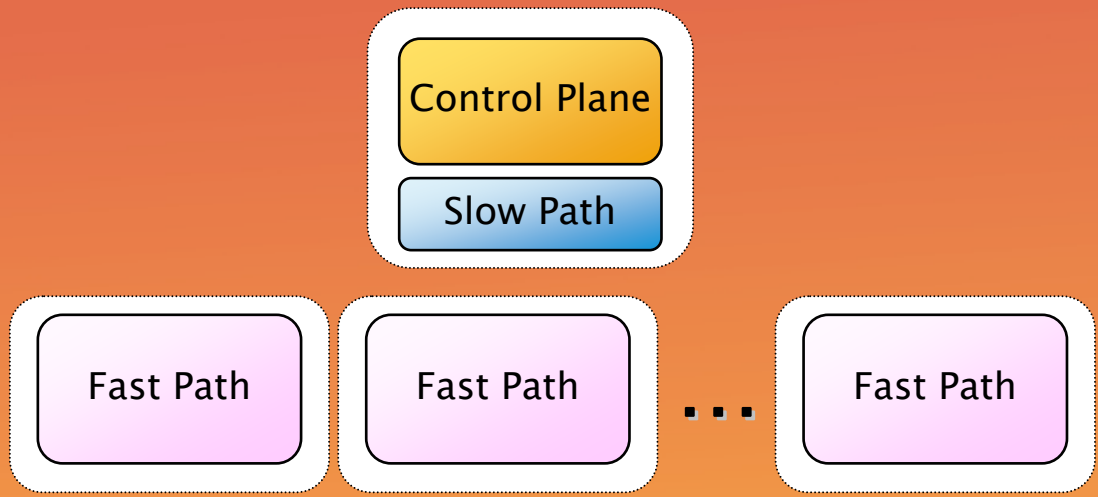
- Use cases
  - Control Plane and Fast Path on different chips
  - Fast Path Extension
  - High Availability Architecture

Use Case 1: Control Plane and Fast Path on different chips



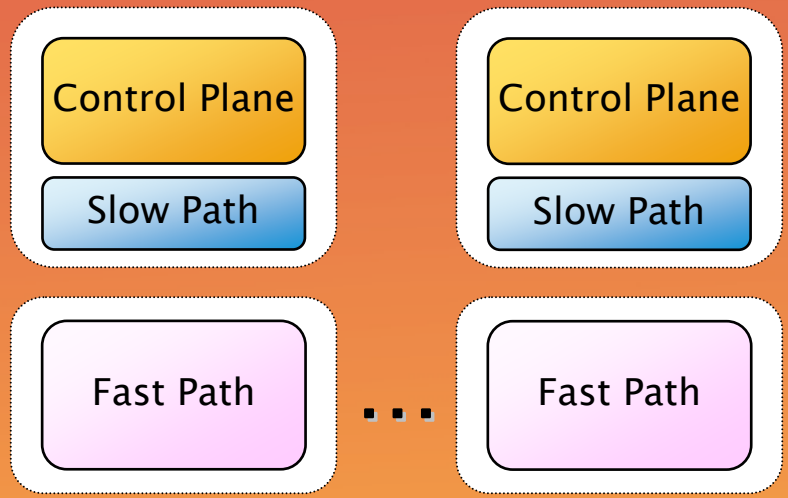
- Control Plane is implemented on a separate processor and communicates with Fast Path through an external interface
- Control Plane and Data Plane processors can be heterogeneous

Use Case 2: Fast Path Extension



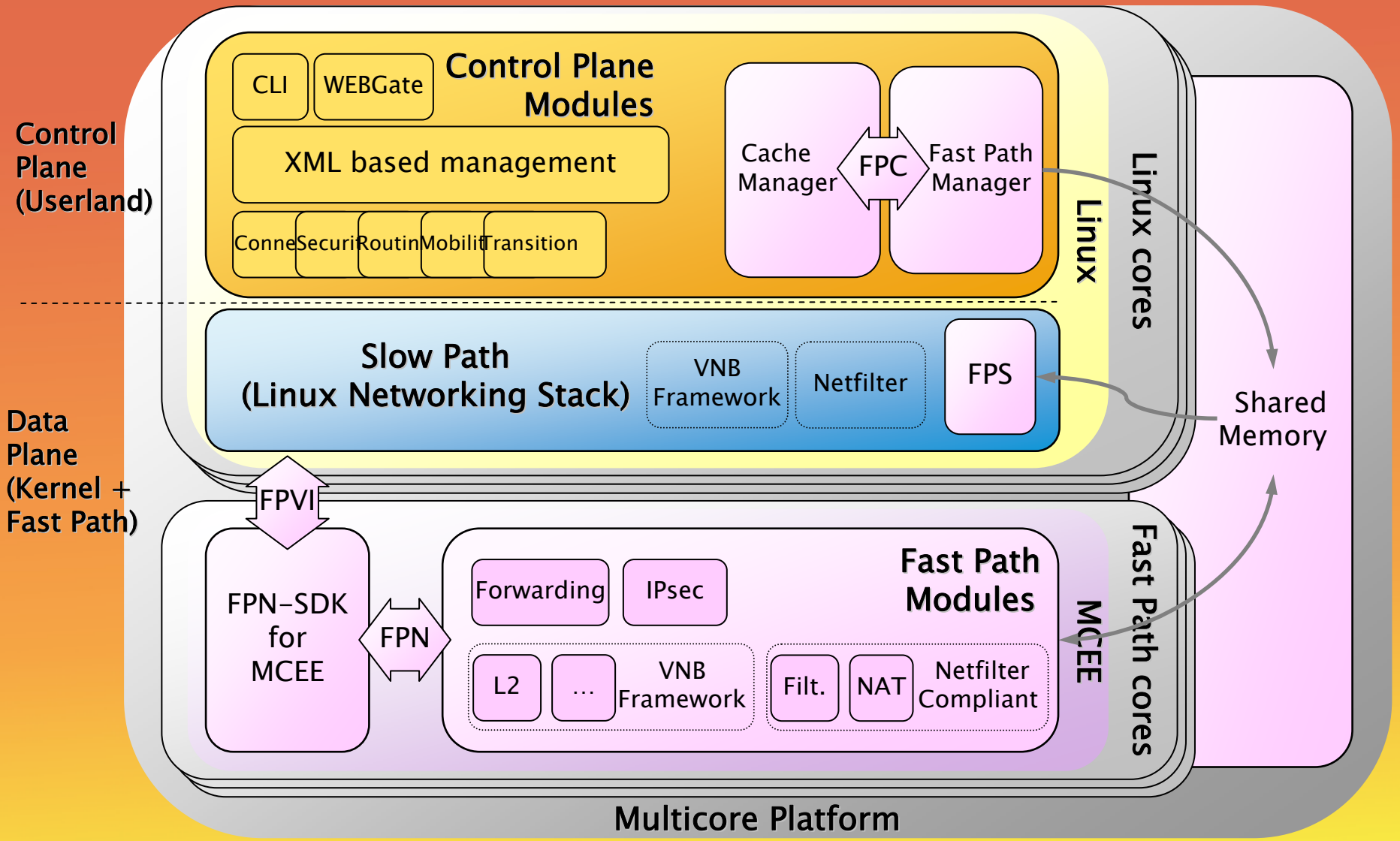
- Fast Path extension requires more than 1 multicore unit to achieve the expected levels of performance
- Fast Path functions are distributed over several units
- As for Use Case 1, Control Plane is implemented on a separate processor; Control Plane and Data Plane processors can be heterogeneous

Use Case 3: High Availability Architecture



- System architecture relies on redundant architecture to provide high availability
- Control Plane protocols have to maintain a coherent view of the system for all instances

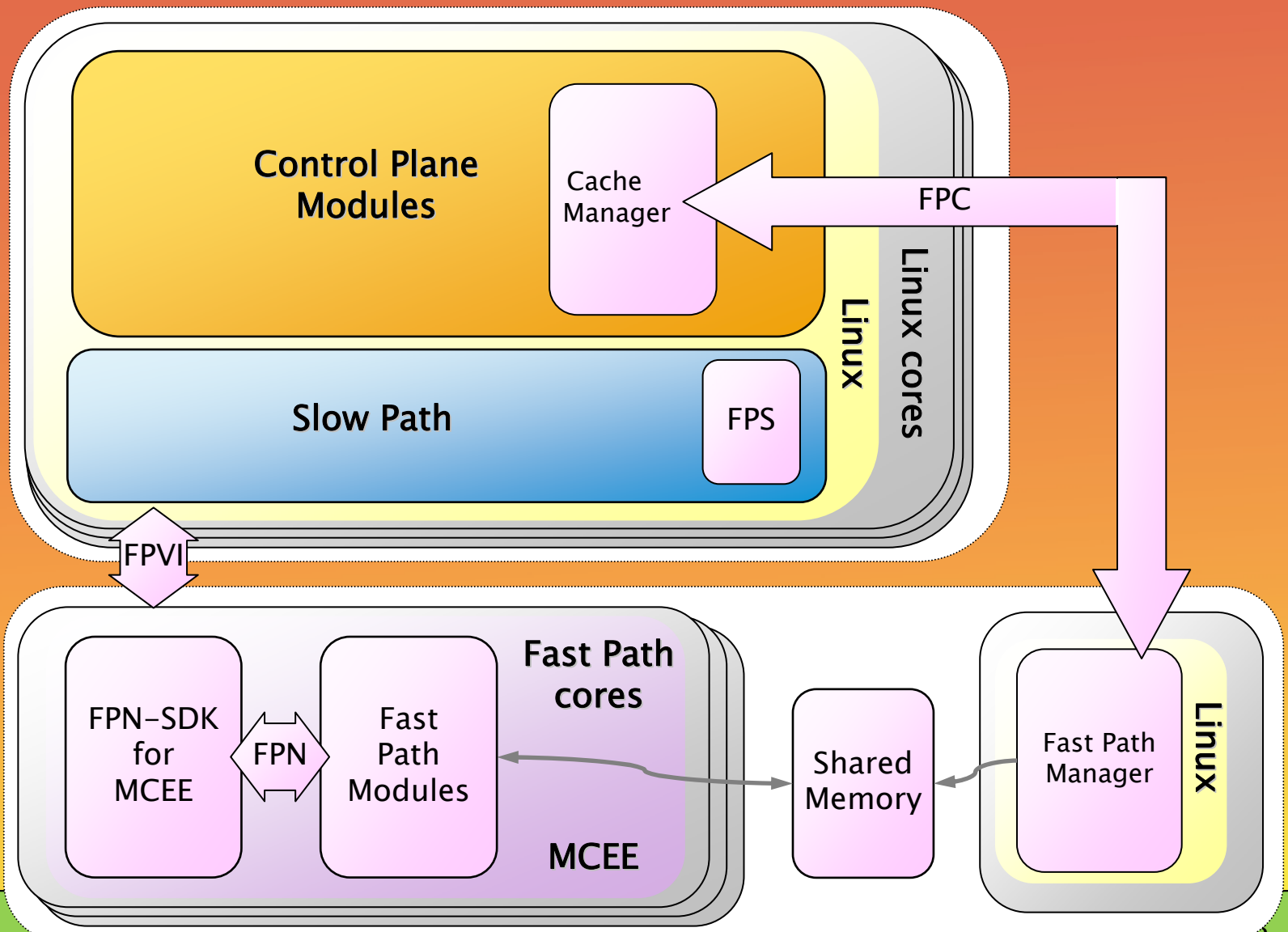
6WINDGate™ SDS (MCEE Fast Path)



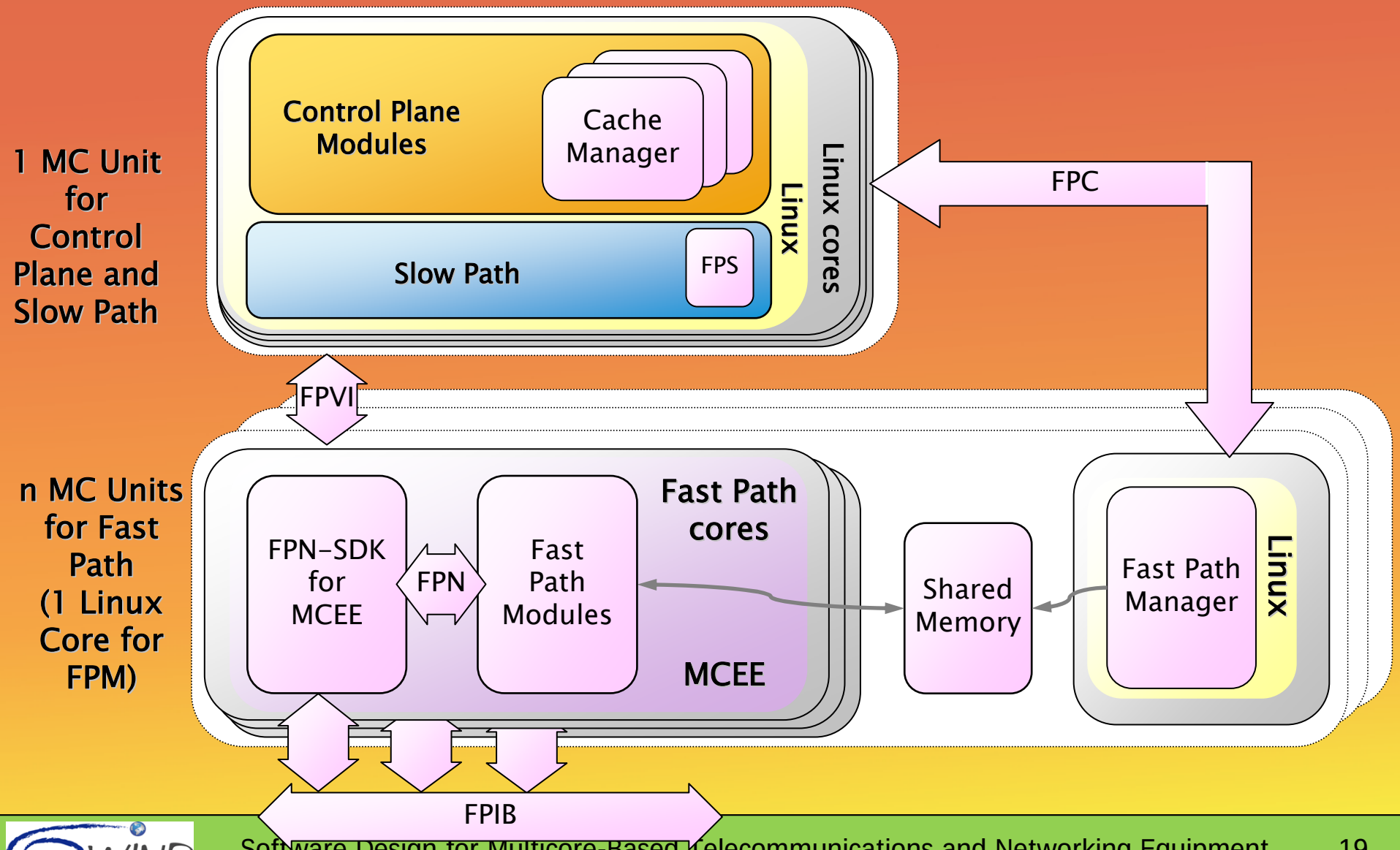
6WINDGate™ SDS for Distributed Architectures (CP to FP)

1 MC Unit for Control Plane and Slow Path

1 MC Unit for Fast Path (1 Linux Core for FPM)



6WINDGate™ SDS For Distributed Architectures (CP to n FP)



Software Architecture Shall Enable and Simplifying Multicore Designs

- Packet processing has to be specifically designed for multicore
- Multicore software architecture has to be flexible to fully benefit from multicore
  - Architecture has to scale from 2 to n cores
  - It has to be extensible over large multicore configurations
- Software reuse is a major requirement from Day 1

Simplify and Expedite Networking Designs Within Multicore

Thank YOU

www.6wind.com

Powered by 6WIND™