



Programming SmartNICs: From Packet Processing to Programmable Transport

SmartNIC Architectures and Programming Models

Mario Baldi

Distinguished Architect

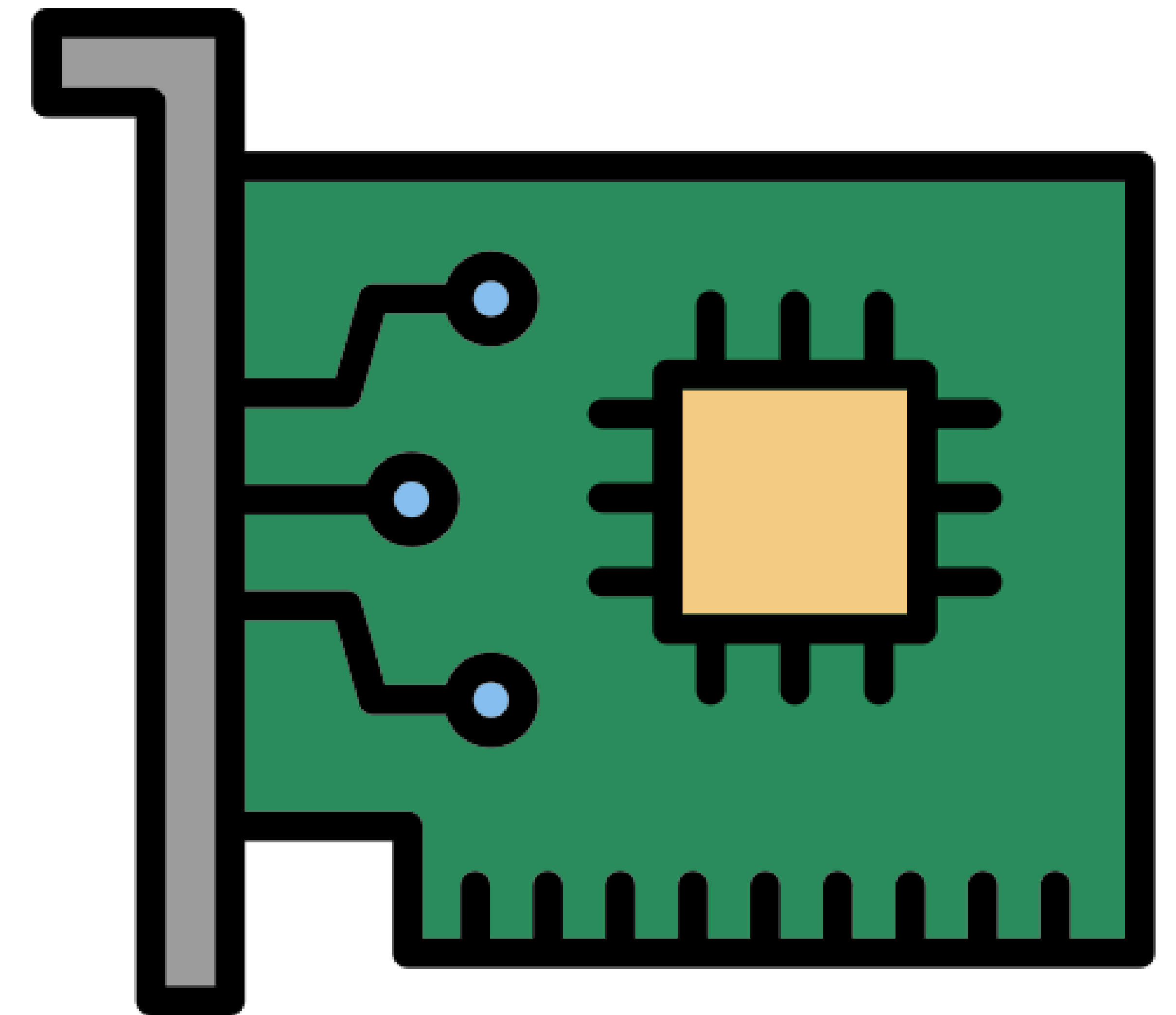
NVIDIA, Networking Software and Systems Architecture

SmartNIC - The Origin

First there was fixed function silicon

Well-established low-level functionalities

Physical and Data-Link
OSI Layers



SmartNIC - The Origin

Then came the Cloud

Move networking out of the CPU

- Spare CPU cycles
 - CPU time is money
- Tunneling
 - Virtual private clouds
 - Provider specific solutions
 - Custom control plane
- Separate ownership of networking and compute
 - Separate administration domains
 - Separate security domains
 - Hypervisor was an alternative
 - Not suitable for bare metal cloud
 - Uses CPU



SmartNIC Requirements

Move networking out of the CPU

- Spare CPU cycles
 - CPU time is money
- Tunneling
 - Virtual private clouds
 - Provider specific solutions
 - Custom control plane
- Separate ownership of networking and compute
 - Separate administration domains
 - Separate security domains
 - Hypervisor was an alternative
 - Not suitable for bare metal cloud

Programmable, high-performance packet processing

- Header processing
- Encapsulation/decapsulation
 - E.g., for tunneling

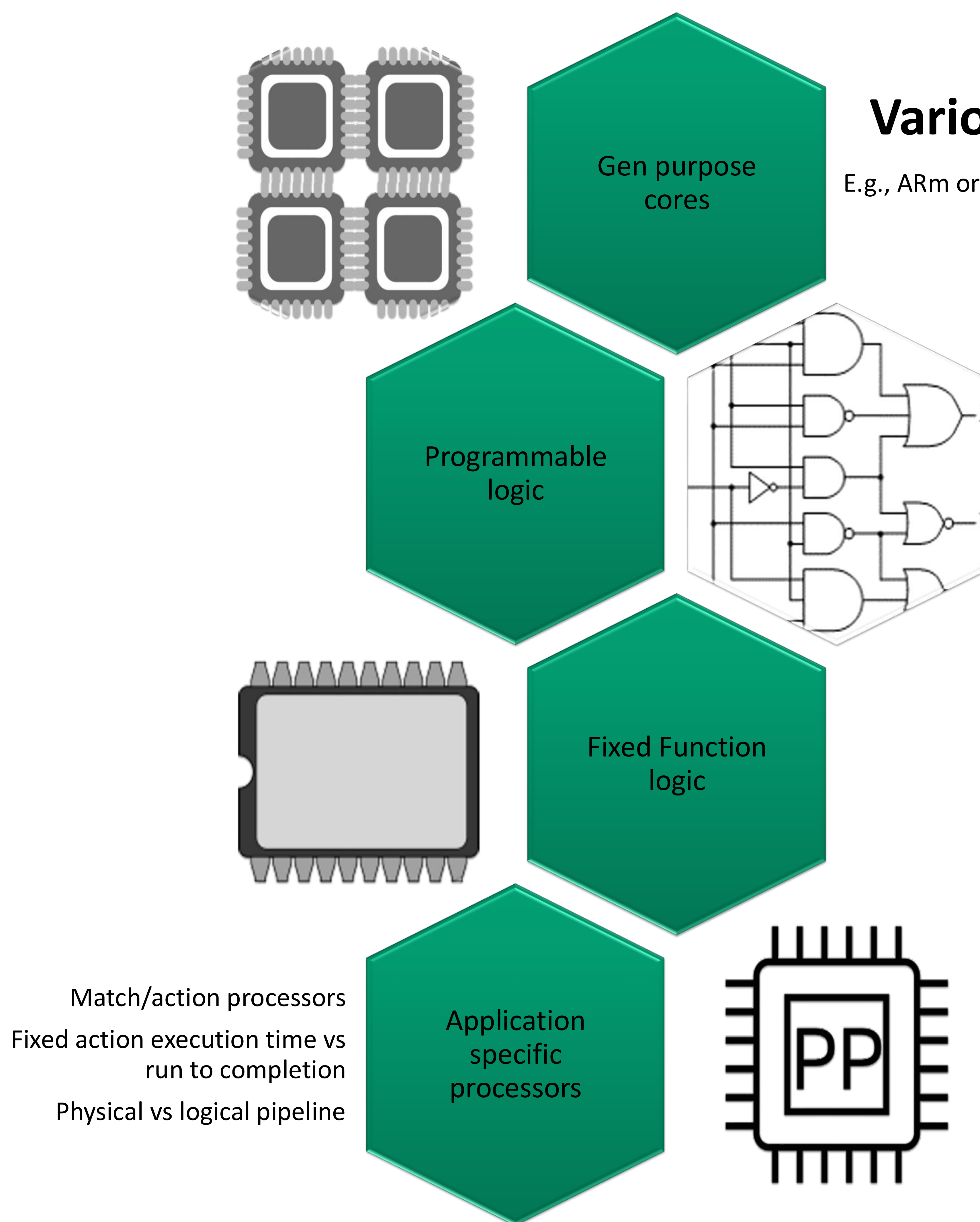
General purpose processing for control plane and management plane

Possibly separate management interface

Trusted computing platform

- E.g., TPM

Various designs



SuperNIC

- One or more components tightly integrated to process each packet at high performance
 - Fast data path

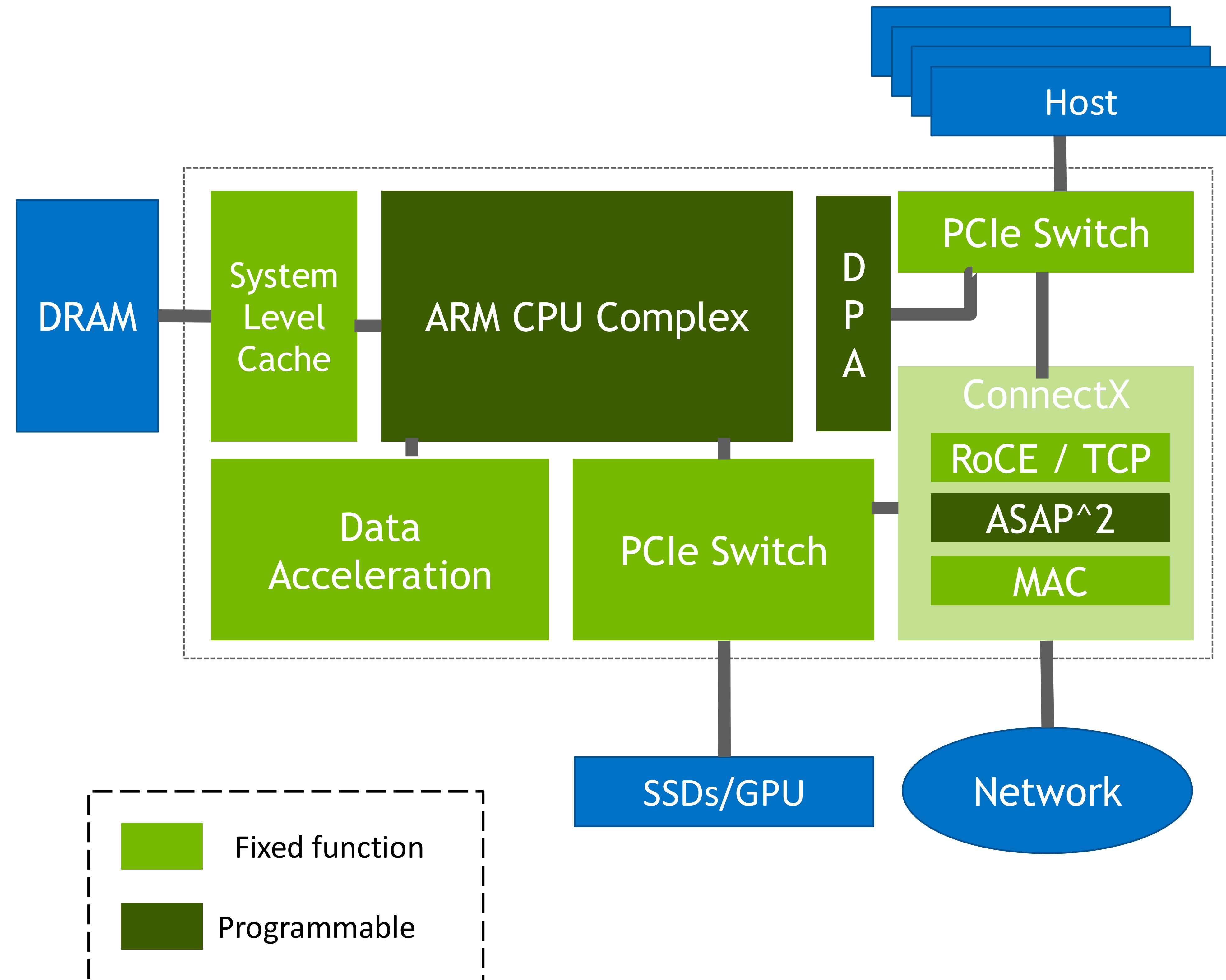
DPU/IPU

- Addition of general-purpose cores more loosely integrated for processing of selected packets
 - Slow data path or control plane
 - E.g., ARM or RISC-V
 - Mostly an extension of the host CPU
 - Spare CPU cycles
 - Separate security and administrative domain
 - Provide controlled environment

An Example DPU: NVIDIA BlueField-3

DPA – Data Path Accelerator

ASAP^2 – Accelerated Switching and Packet Processing



NVIDIA BlueField-3 DPU

For real and in numbers

22 Billion Transistors

400Gb/s Ethernet & InfiniBand Connectivity

400Gb/s Crypto Acceleration

330M PPS, 80M PPS @ Scale

18M IOP/s Elastic Block Storage

300 Equivalent x86 Cores

DATA PATH ACCELERATOR

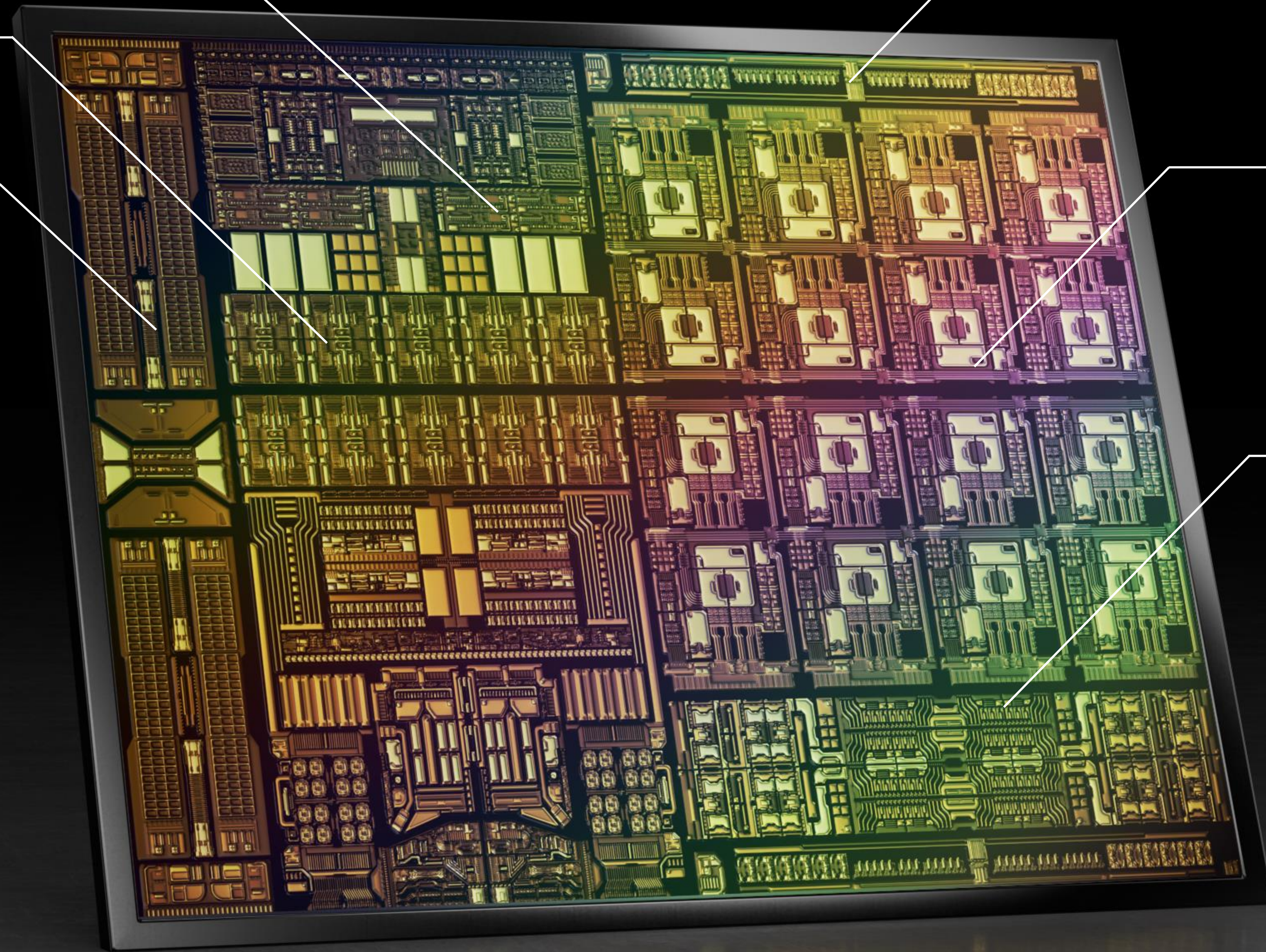
CONNECTX-7

PCIe GEN 5.0

DDR5 MEMORY INTERFACE

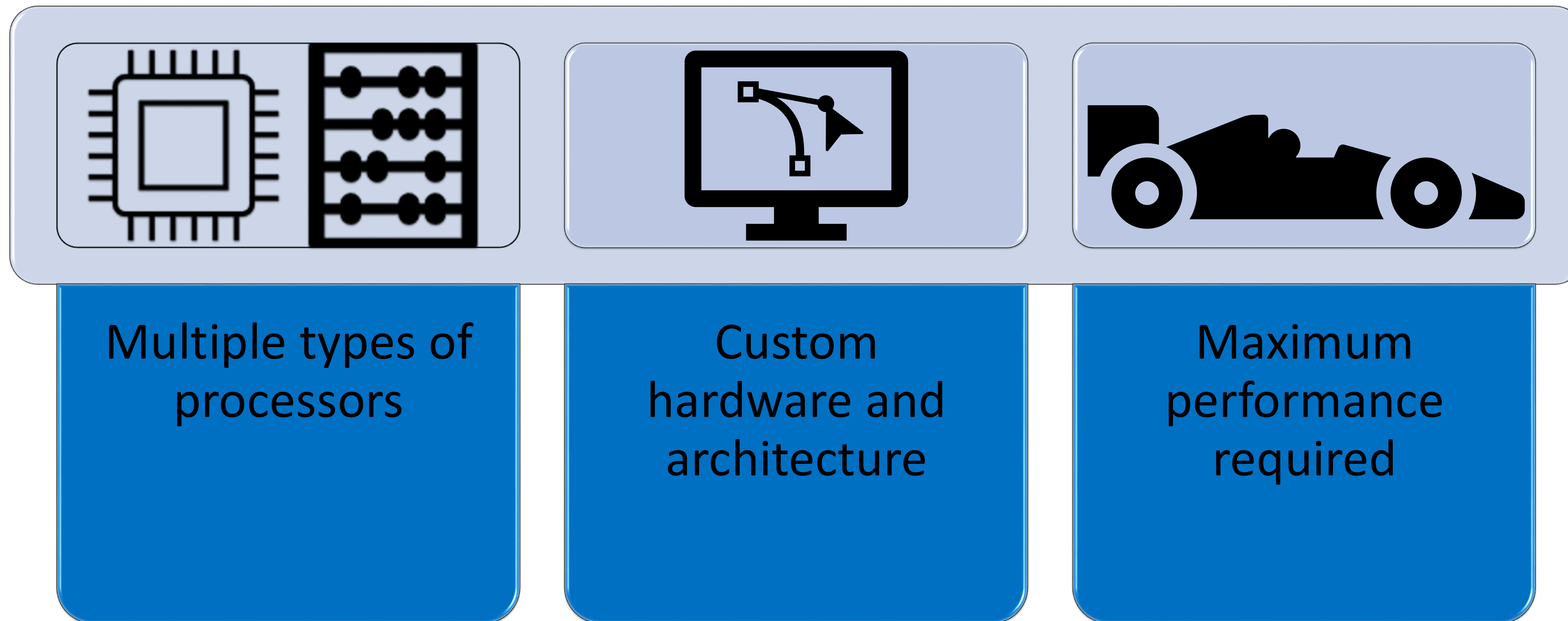
ARM CORES

ACCELERATION
ENGINES

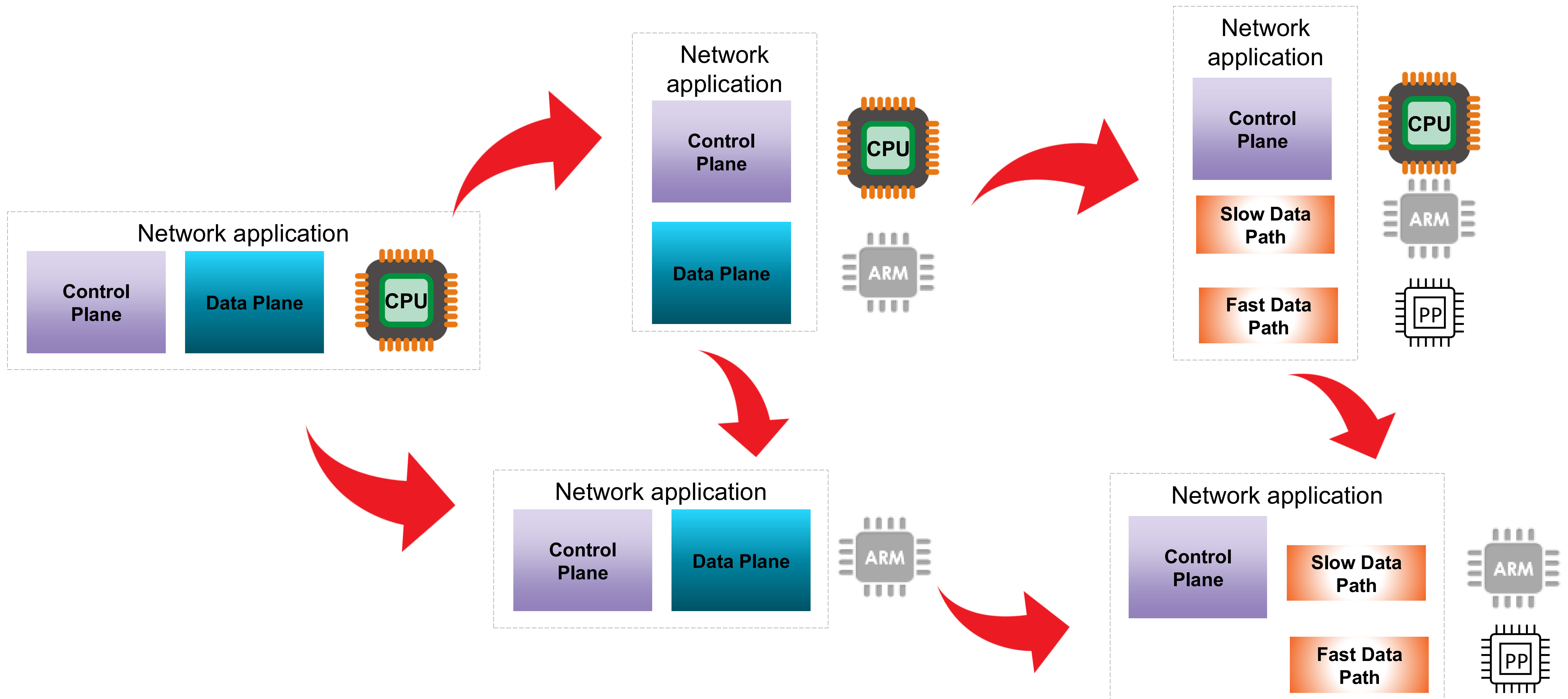


SmartNIC Programming: a Challenge

SmartNIC programming is complex in nature



Various Ways of and Paths to DPU Deployment



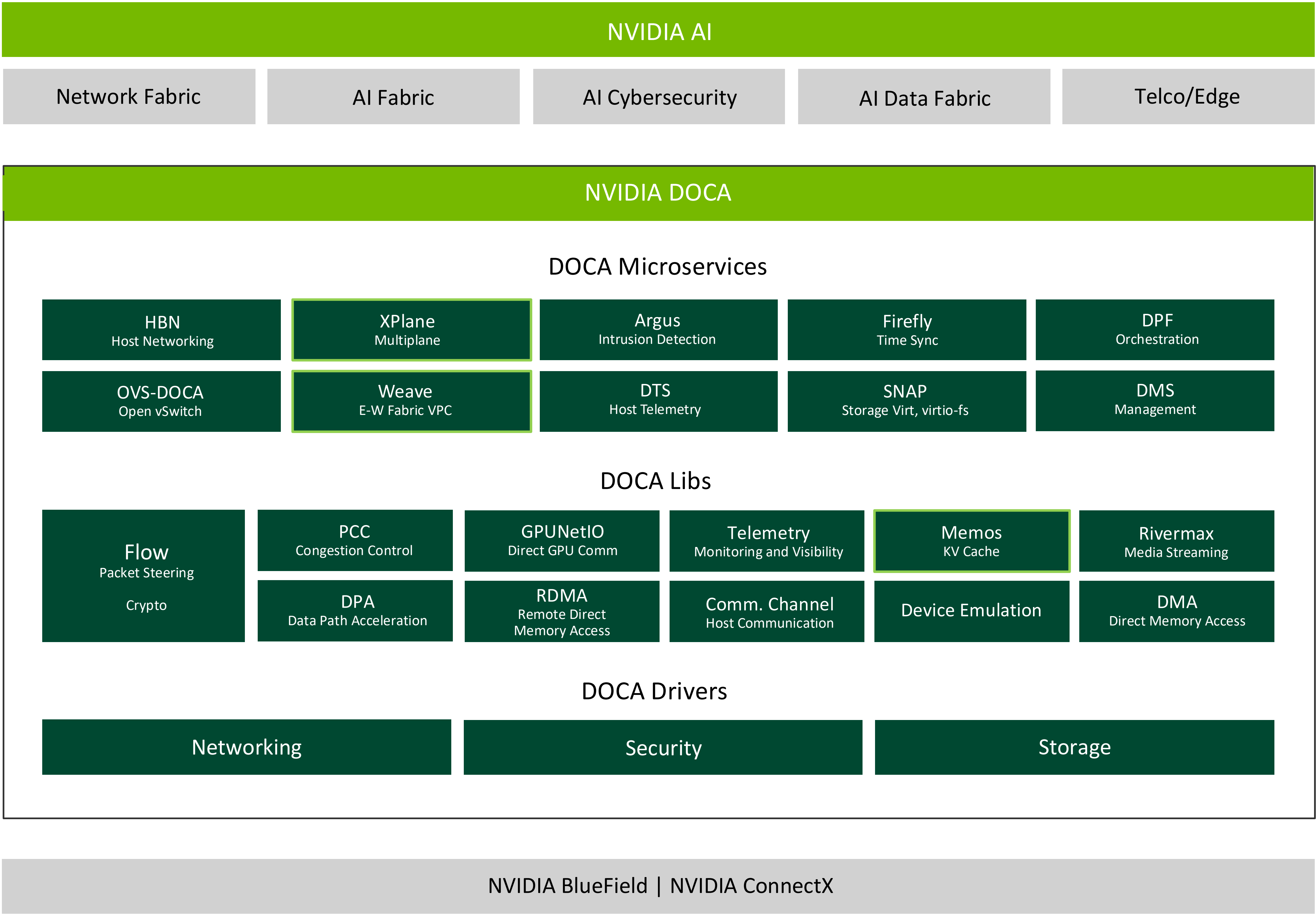
Proper Software Development Framework is Key

- Protocol processing in FPGAs or Application Specific Processors
 - Application specific language
 - E.g., P4, DOCA Flow, DOCA Pipeline Language
- Exception processing possibly in Arm
 - High-level language
 - E.g., C, C++
- Control and management in Arm
 - High-level language or script
 - No extreme performance requirements

*Flexible and
efficient
APIs!*

Efforts to enable a single overall program
that compiles across components are still not mature

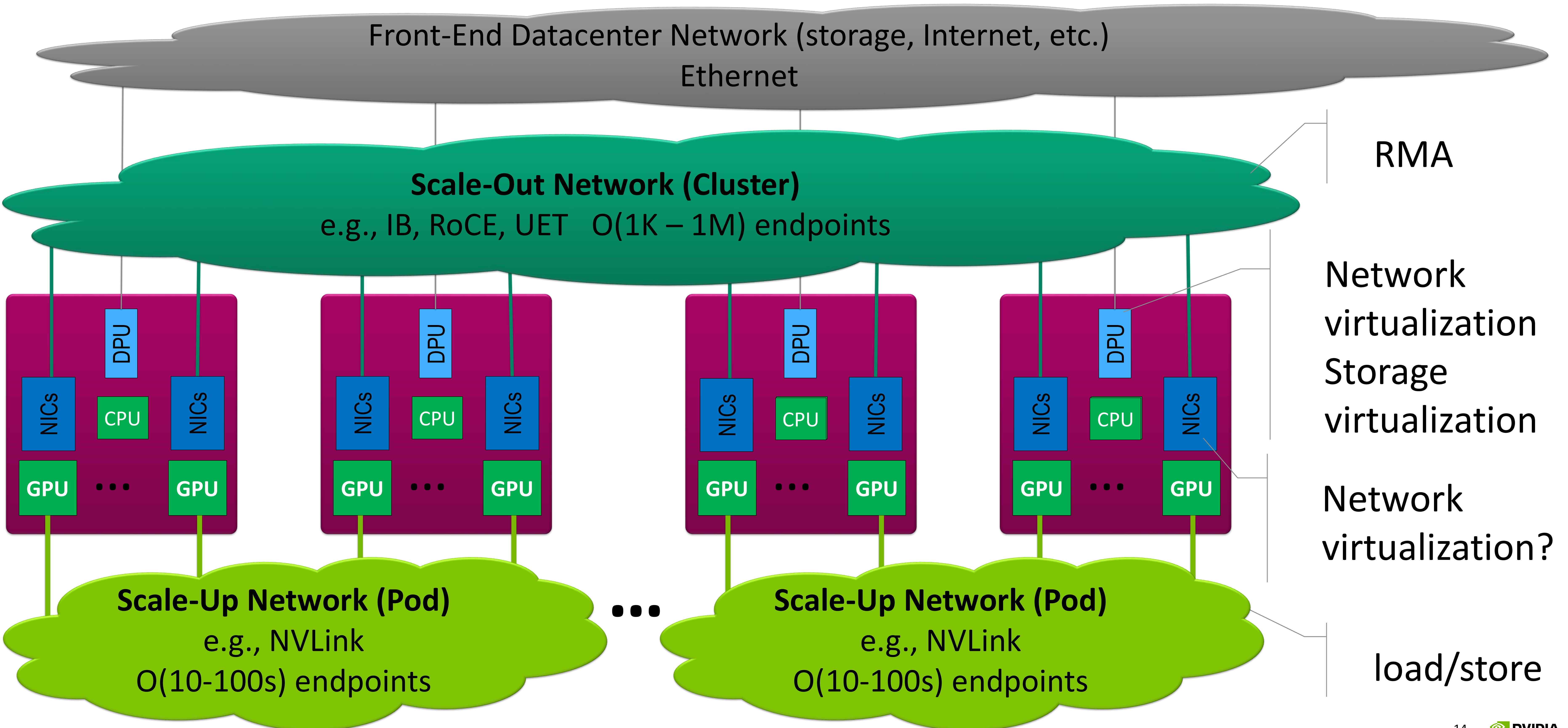
An Example Development Framework: NVIDIA DOCA



SmartNICs in the Age of AI



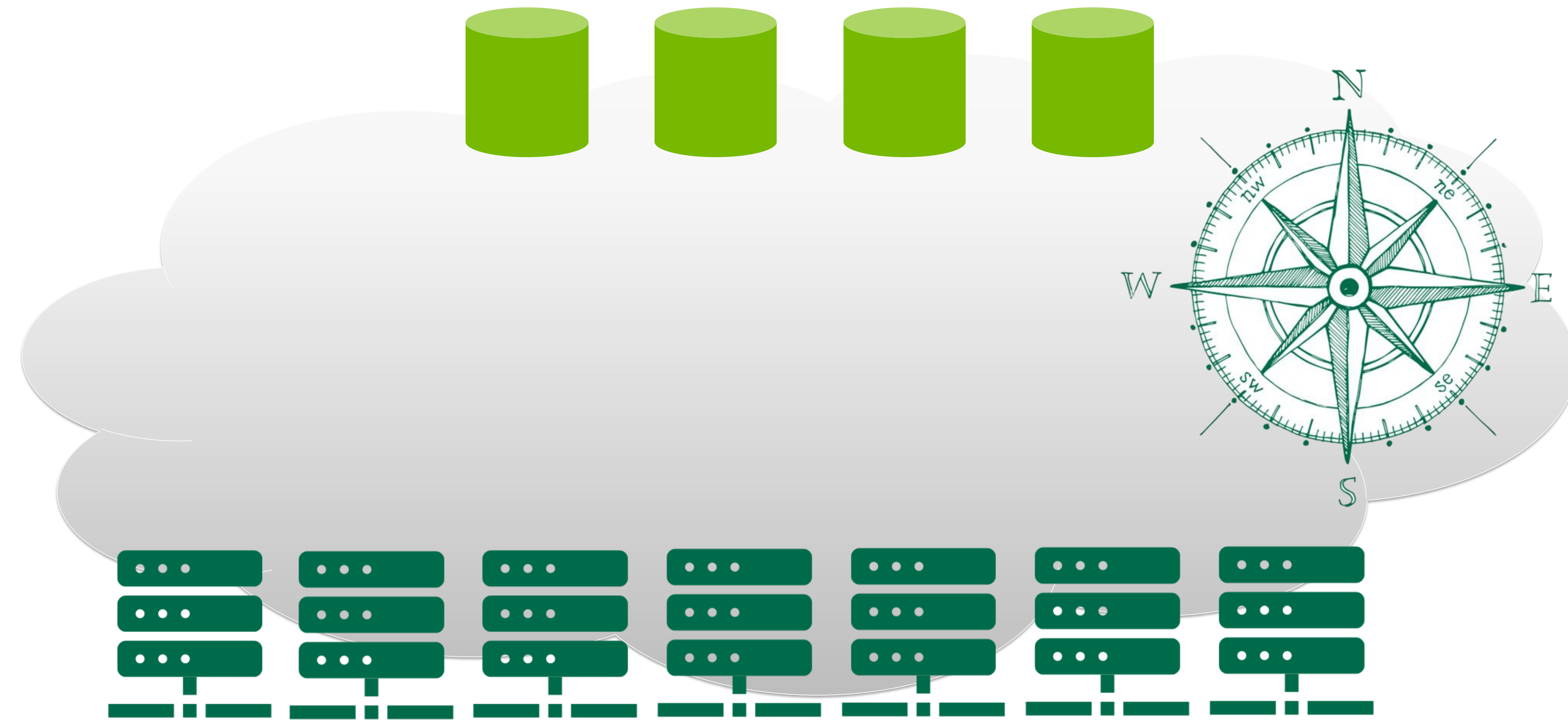
AI Datacenter Network Architecture



The Role of [Smart]NICs

In the scale-out network (a.k.a. East-West Traffic)

- Highly optimized, high performance, low cost
- Fixed, simple protocol stack
- Transport offload
 - Don't want to use CPU
 - Slow
 - Not on path from GPU
 - Don't want to use GPU
 - Not efficient
- With Agentic AI both CPU and GPU want to use this network



Is there a need for programmability?

The Transport Layer Might Need Programmability

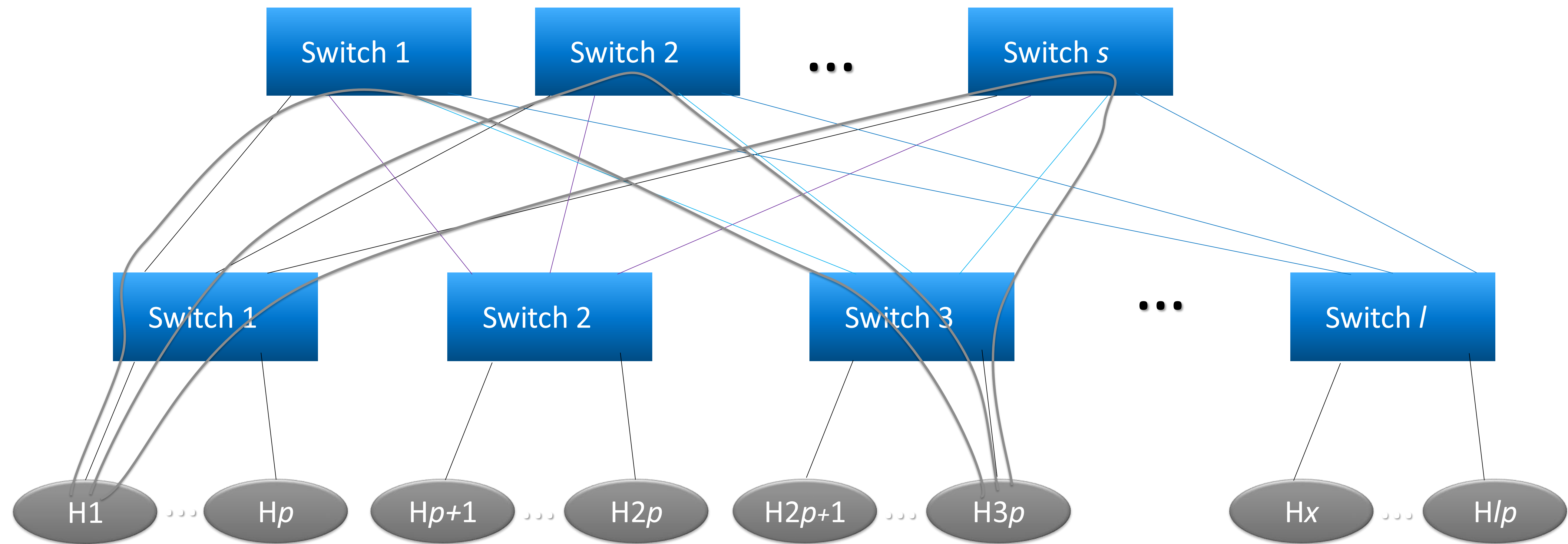
For both NIC vendors and NIC users

No universally accepted solution for

- Multipathing
- Congestion Control
- Congestion signaling

Various standard and vendor/user-specific solutions

- Infiniband
- RoCEv2
- UltraEthernet Transport
- Falcon
- Etc.



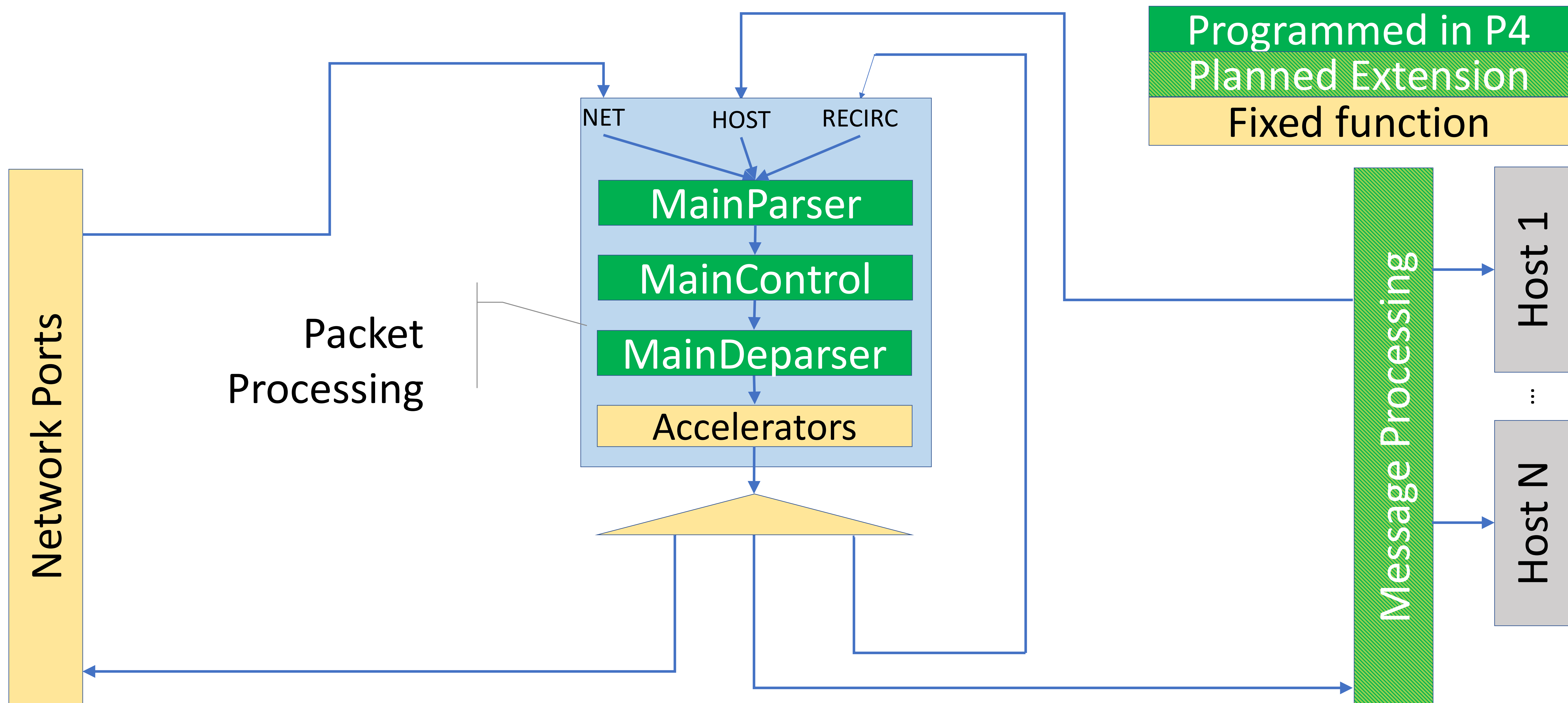
Transport Programmability Unique Features

- Traditional packet processing approaches are not sufficient
 - Typically stateless
 - Traditionally also packet processing hardware is stateless
 - Some stateful aspects introduced in existing languages, such as P4
 - Packet based, while transport is message-based
- Similarity with traditional packet processing: fast data path and slow data path
 - Sometimes called control path
- Do we need specialized programming models?
 - For example, programming across physical components

Focus of the next presentations

Portable NIC Architecture (PNA) v0.7

Transport is specific to the end point (NIC) – not present in the switch (PSA)



p4.org Architecture Workgroup Specification

<https://p4.org/wp-content/uploads/sites/53/p4-spec/docs/PNA-v0.7.html#sec-introduction>

PNA Novelties

Add-on-miss

- Create a new entry in a table directly in the data plane
- Without involving the control plane

Modifiable table entries

New externs

- E.g., crypto extern; IPsec extern

Tutorial Organization

