DPU-Direct: Localizing Remote Accelerators for Disaggregated Datacenters

Yunkun Liao

目录

CONTENTS

01 Background and Motivation

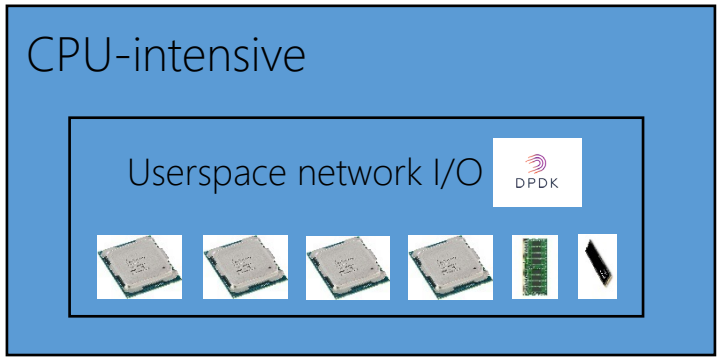
Datacenter Workloads have Diverse Resource Needs

Datacenter Workloads have Diverse Resource Needs

- Different workloads requires diverse resource configurations.

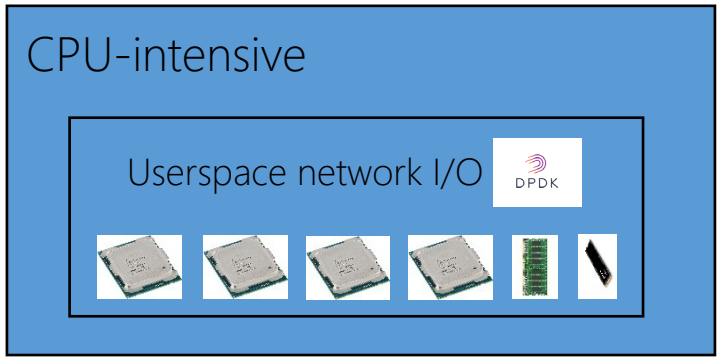
Datacenter Workloads have Diverse Resource Needs

- Different workloads requires diverse resource configurations.



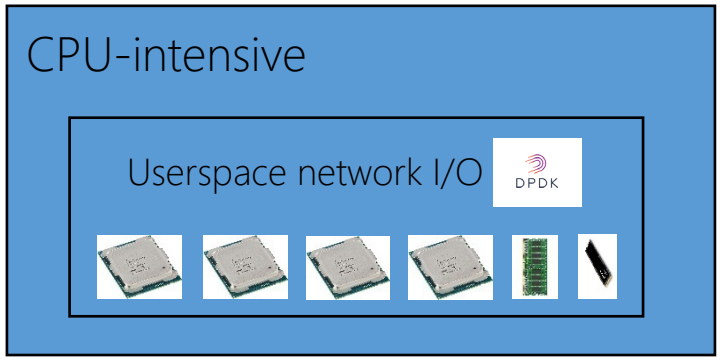
Datacenter Workloads have Diverse Resource Needs

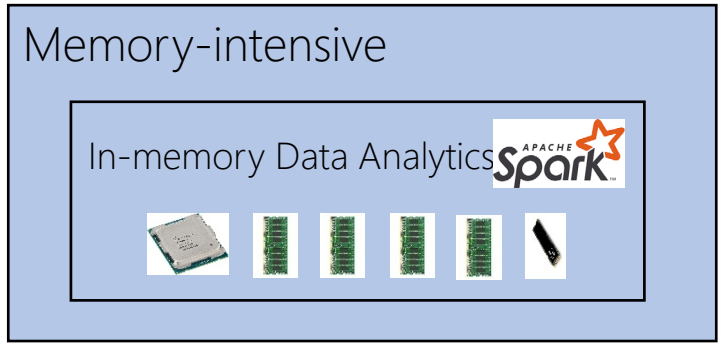
- Different workloads requires diverse resource configurations.



Datacenter Workloads have Diverse Resource Needs

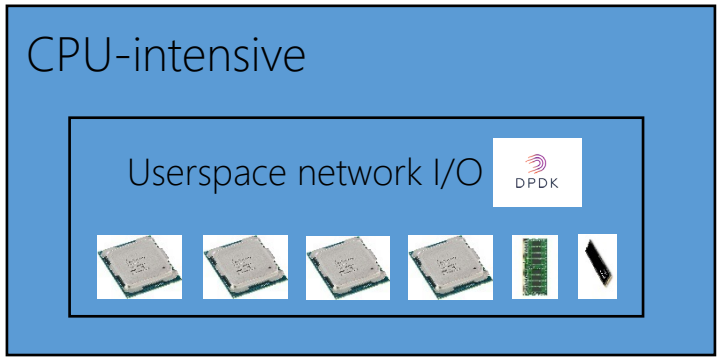
- Different workloads requires diverse resource configurations.

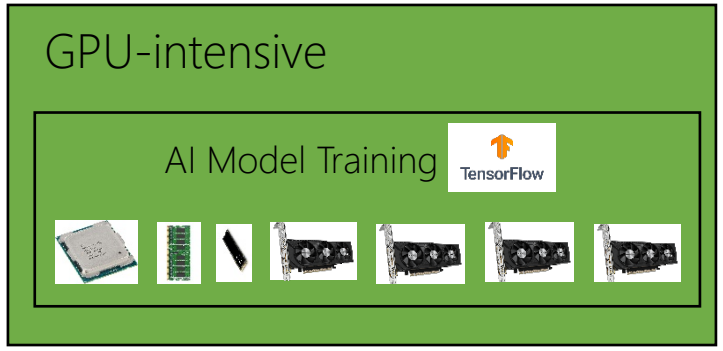


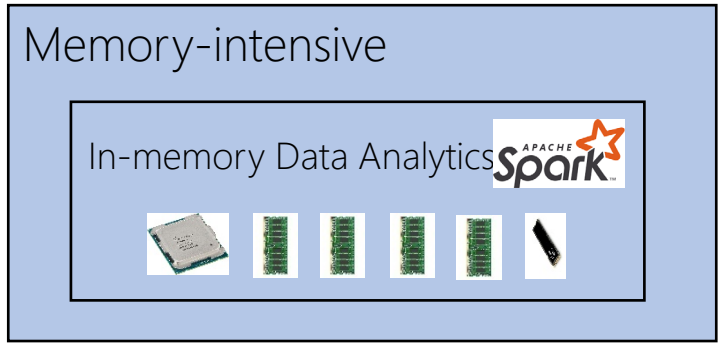


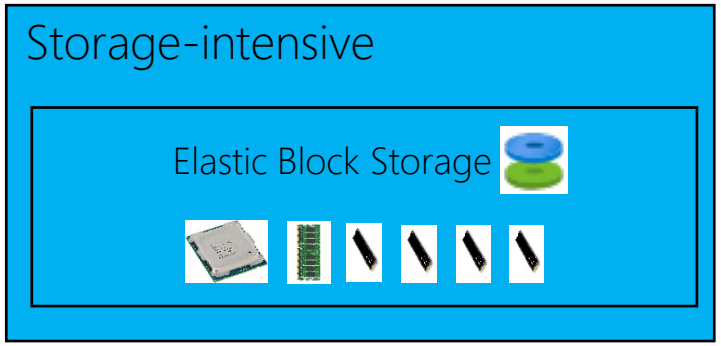
Datacenter Workloads have Diverse Resource Needs

- Different workloads requires diverse resource configurations.



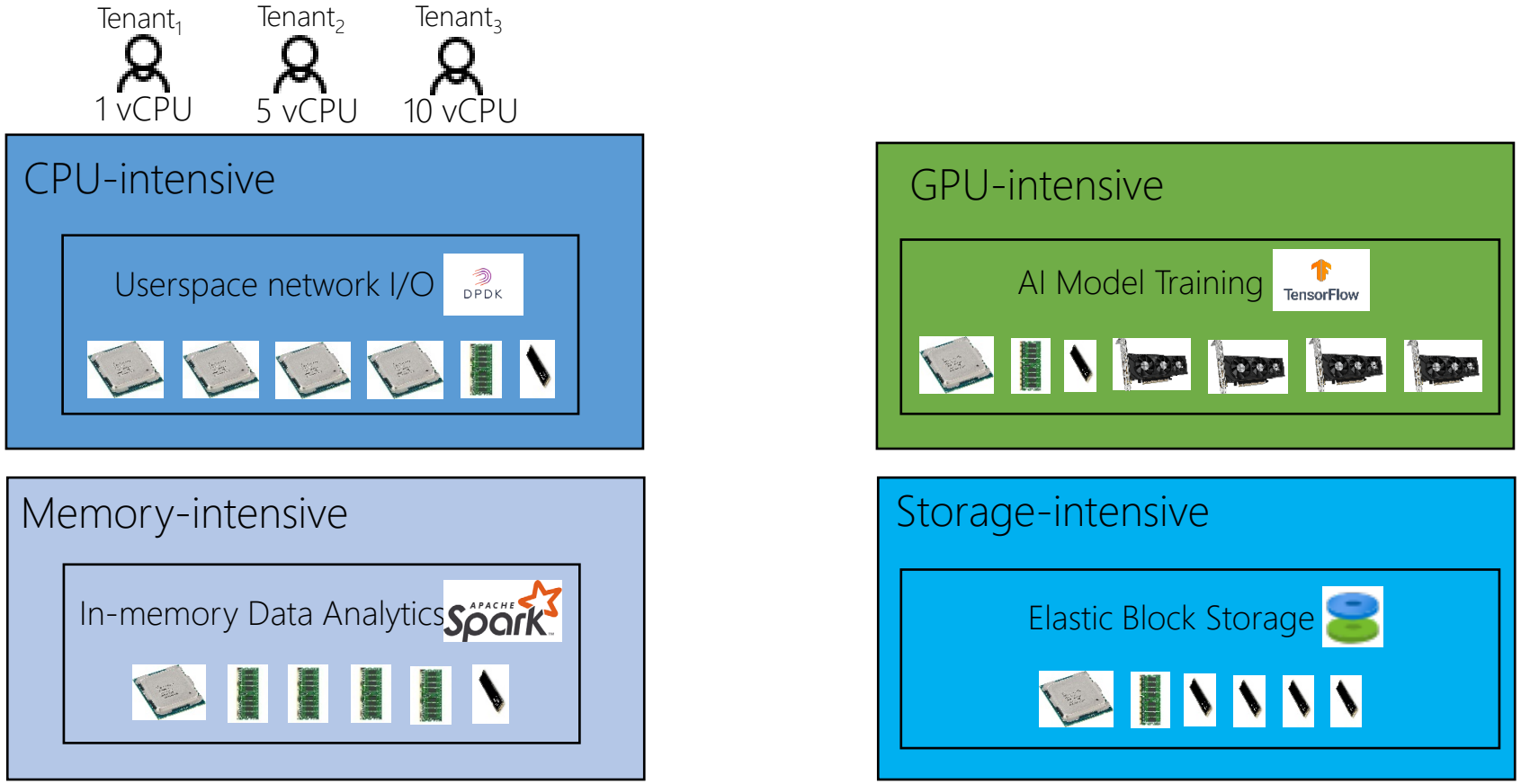






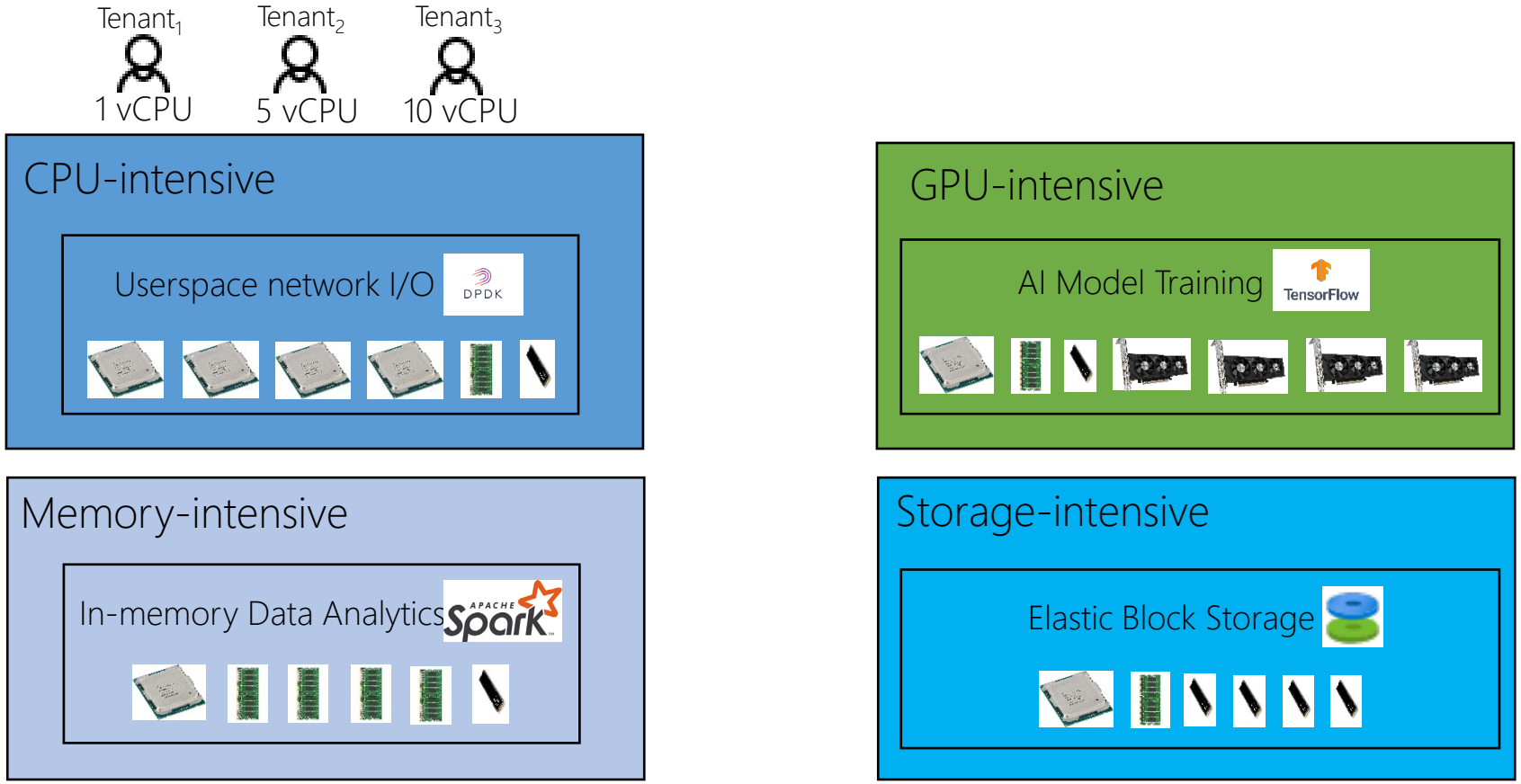
Datacenter Workloads have Diverse Resource Needs

- Different workloads requires diverse resource configurations.
- Tenants sharing the same workload also exhibit different requirements.



Datacenter Workloads have Diverse Resource Needs

- Different workloads requires diverse resource configurations.
- Tenants sharing the same workload also exhibit different requirements.



How to use resources efficiently?

Huge Demands for Scale-out and Cloud-Native Computing

Huge Demands for Scale-out and Cloud-Native Computing

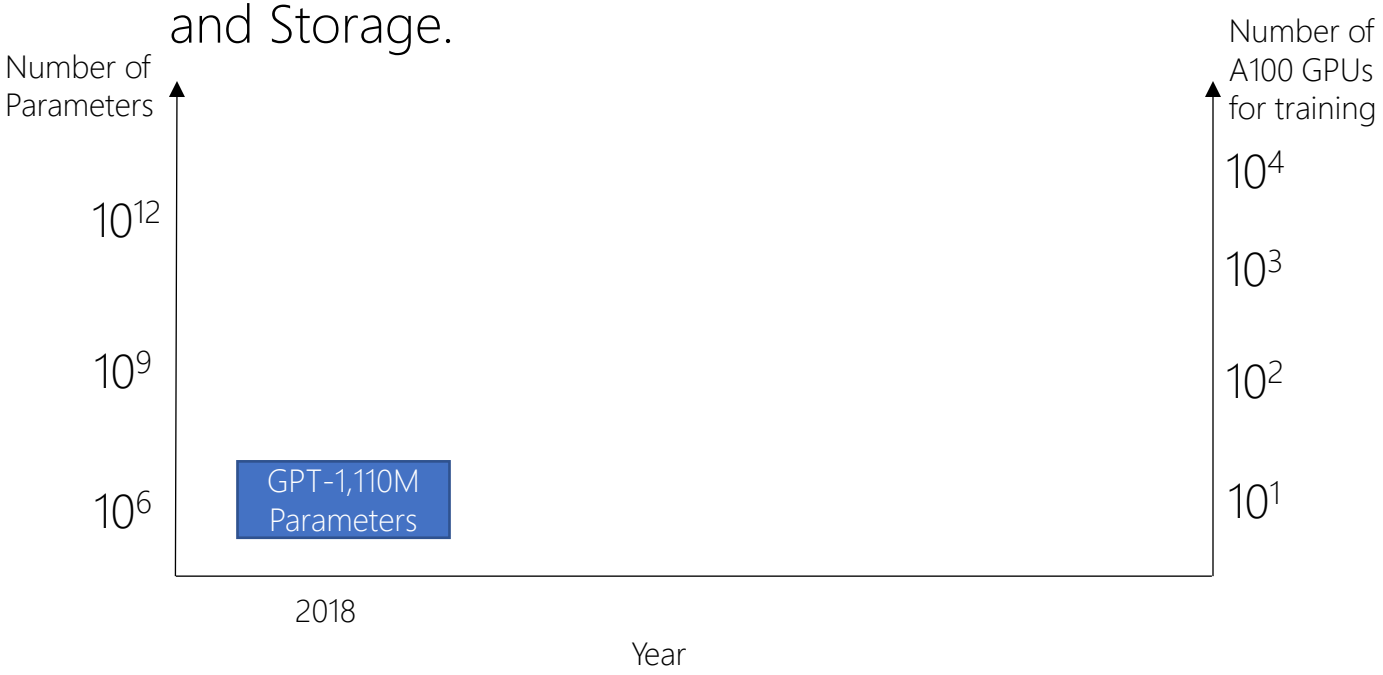
Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.

Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.

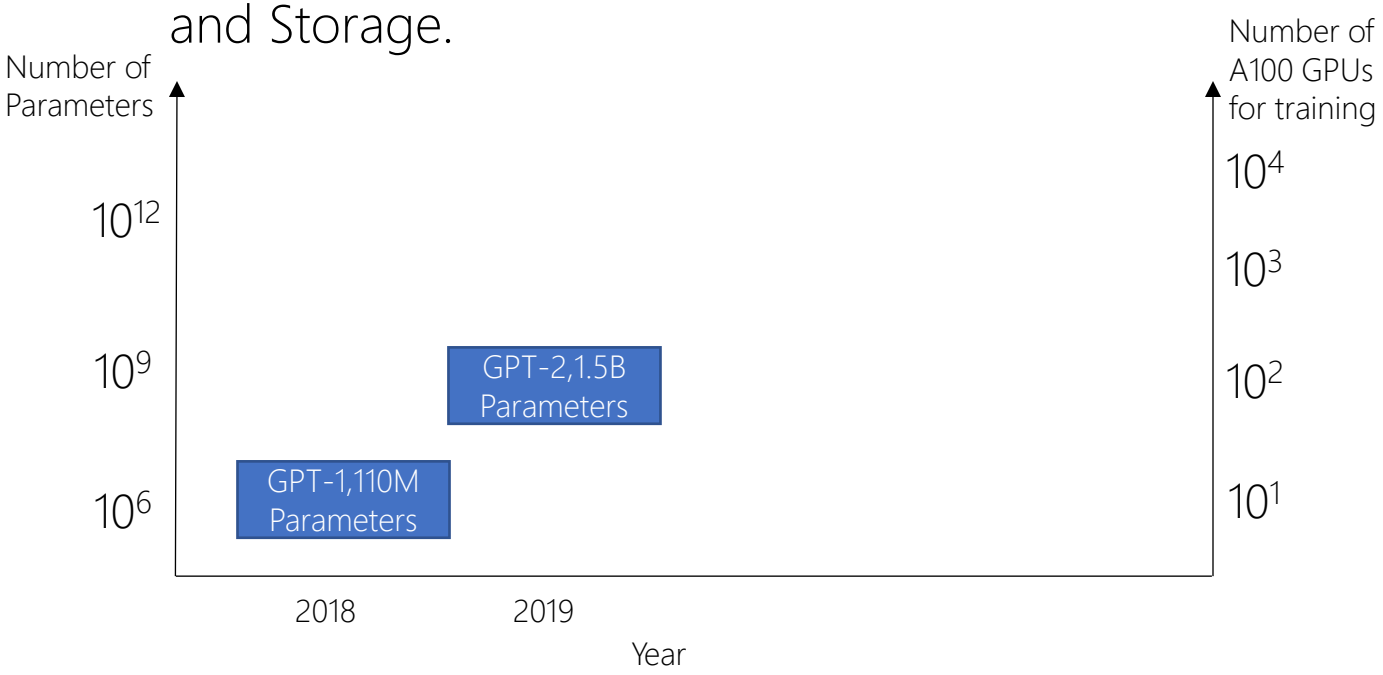
Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



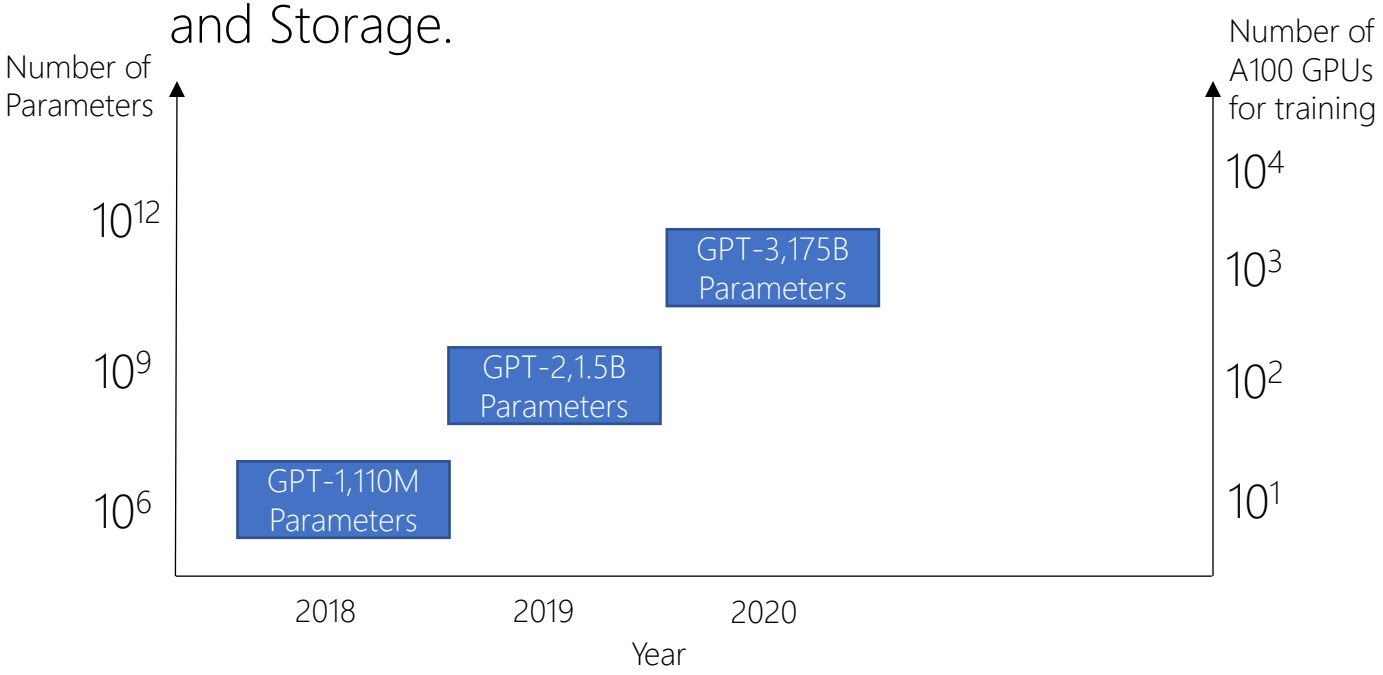
Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



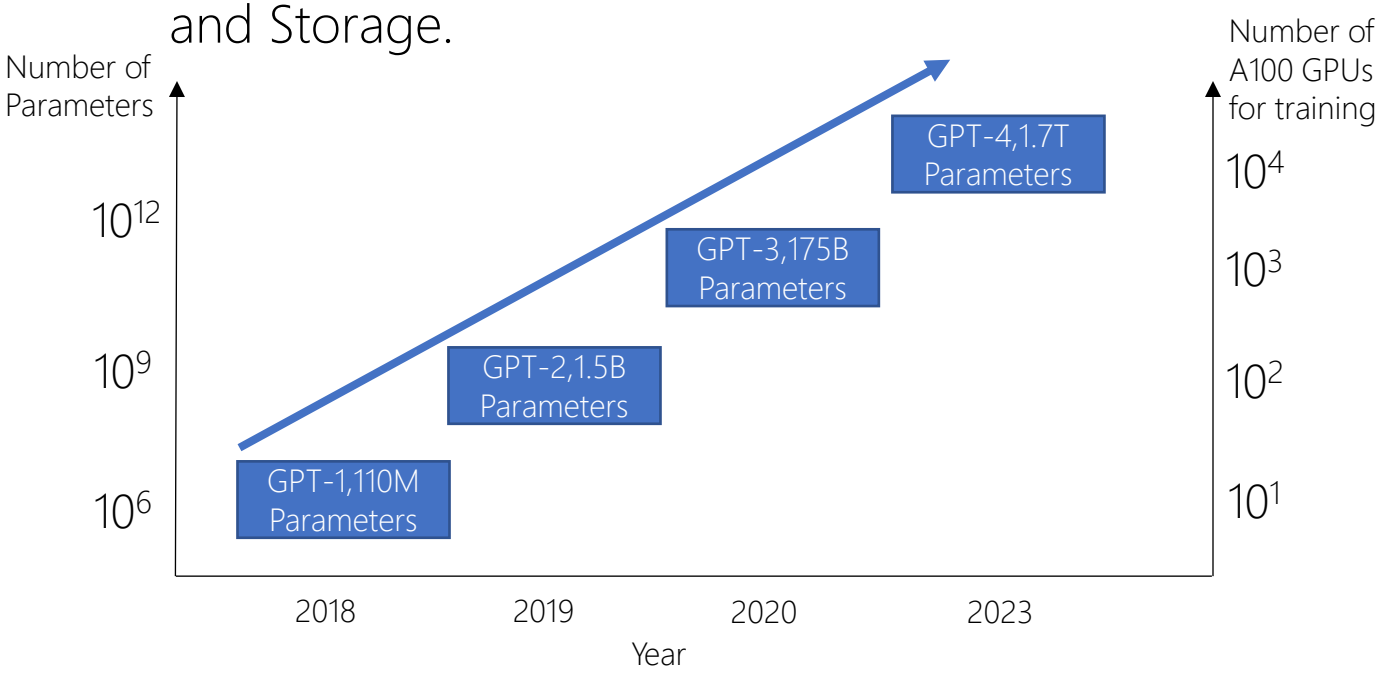
Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



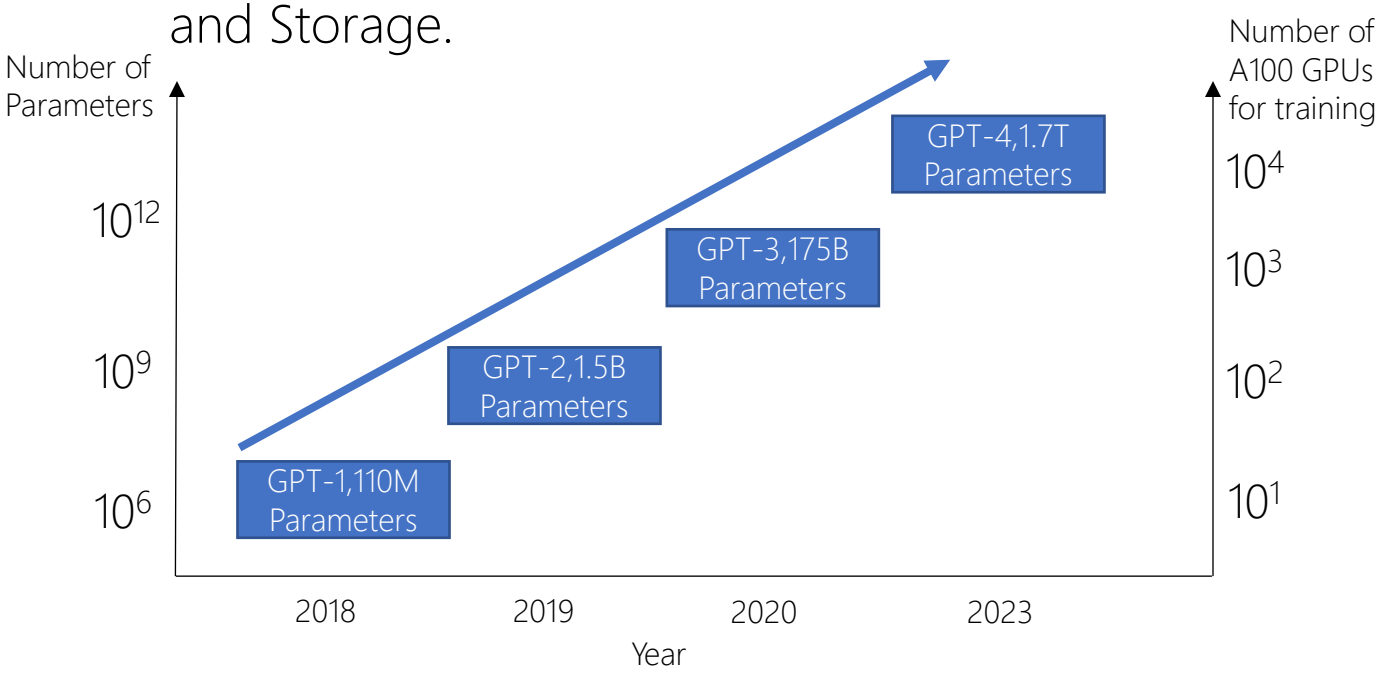
Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



Huge Demands for Scale-out and Cloud-Native Computing

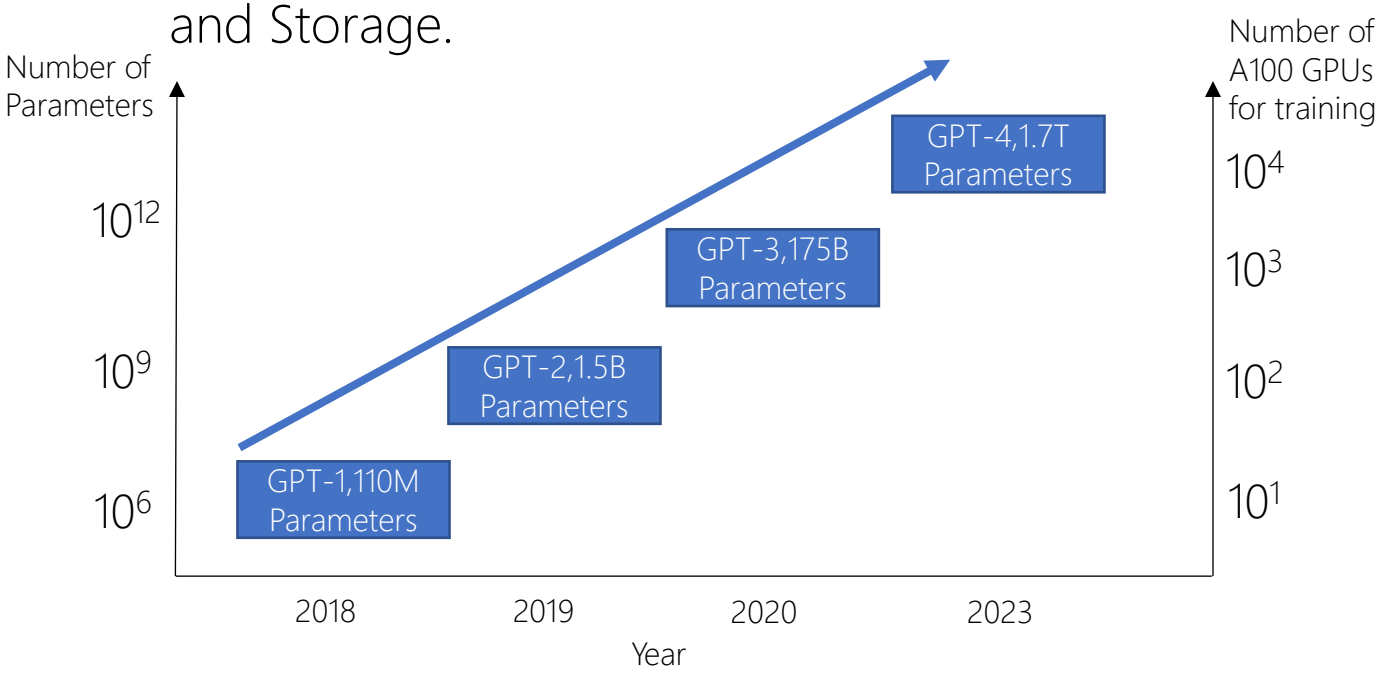
Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



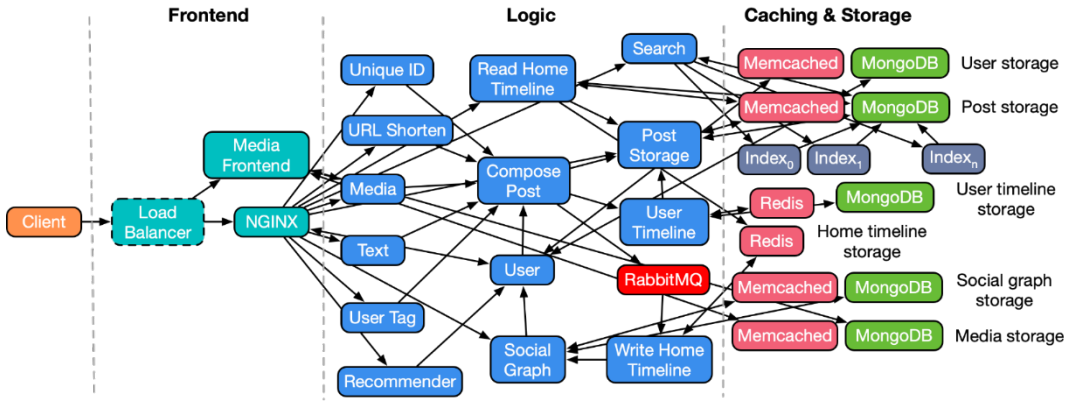
Trend 2: Cloud-Native applications get more distributed, and loosely coupled.

Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



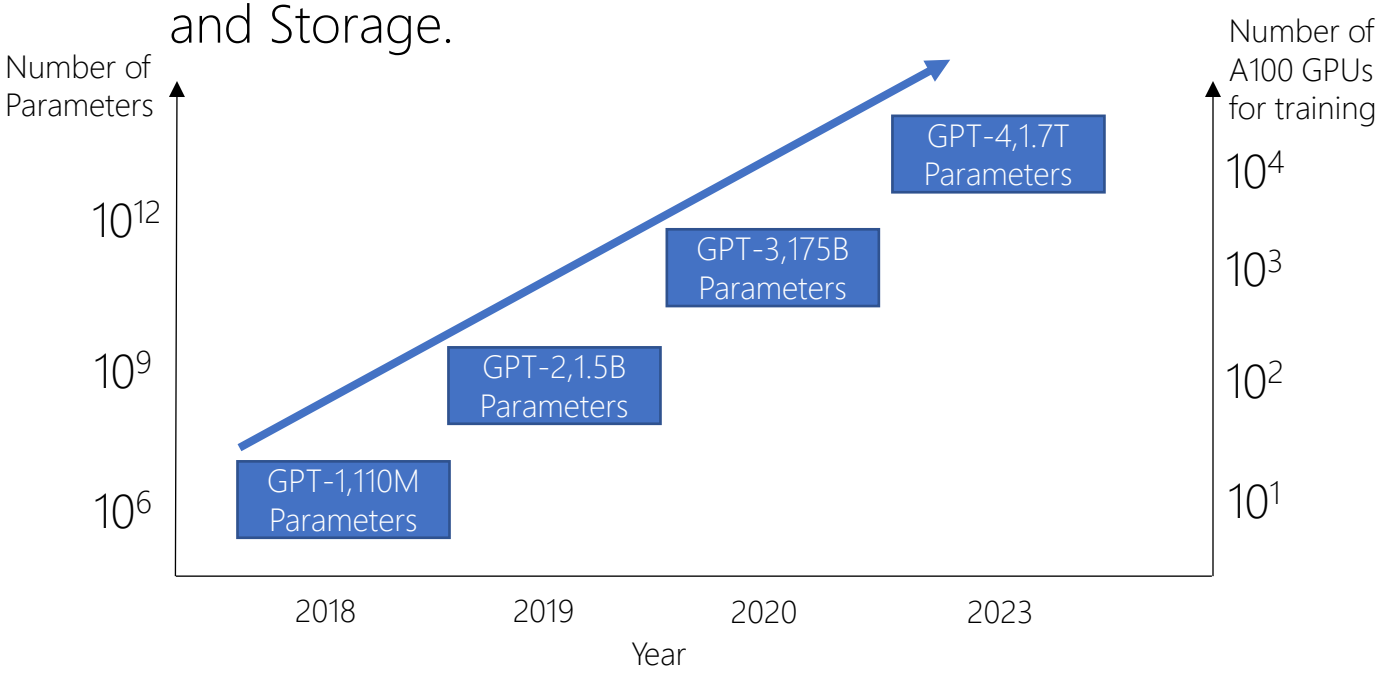
Trend 2: Cloud-Native applications get more distributed, and loosely coupled.



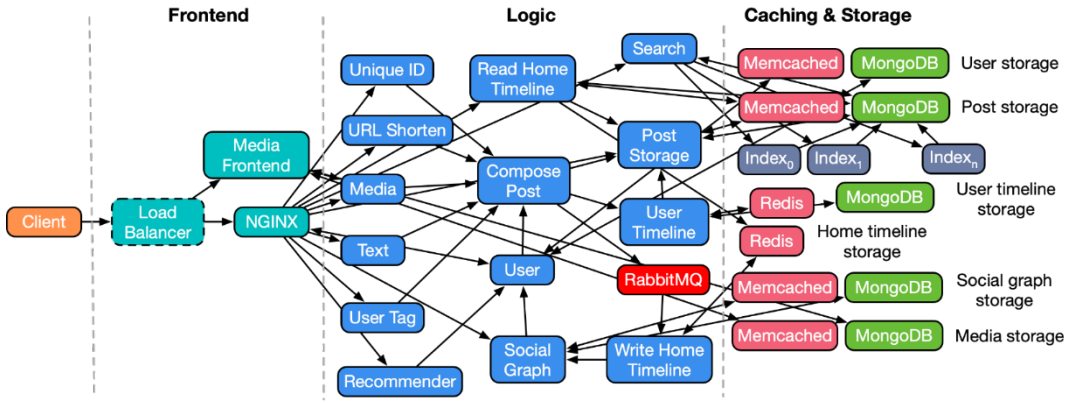
Social network application based on microservice architecture[1]

Huge Demands for Scale-out and Cloud-Native Computing

Trend 1: Exponential Growth of Large Language Model Training in both Computing and Storage.



Trend 2: Cloud-Native applications get more distributed, and loosely coupled.



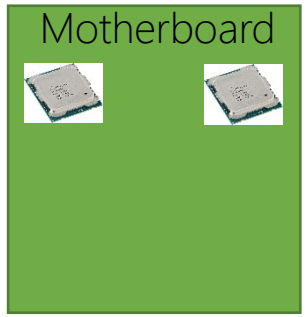
Social network application based on microservice architecture[1]

How to match the datacenter infrastructure with the computing trends?

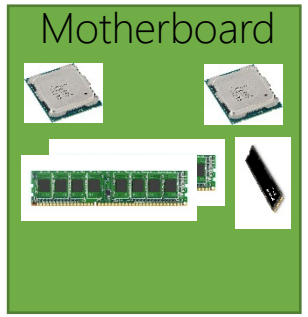
Drawbacks of Monolithic-Server-driven Datacenter

Drawbacks of Monolithic-Server-driven Datacenter

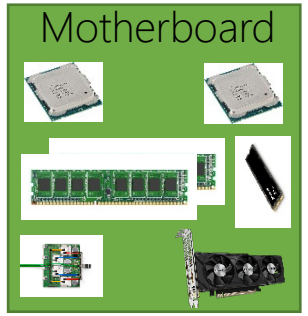
Drawbacks of Monolithic-Server-driven Datacenter



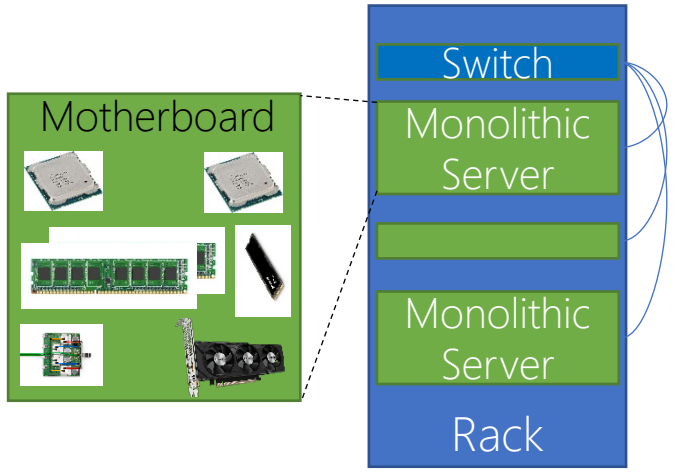
Drawbacks of Monolithic-Server-driven Datacenter



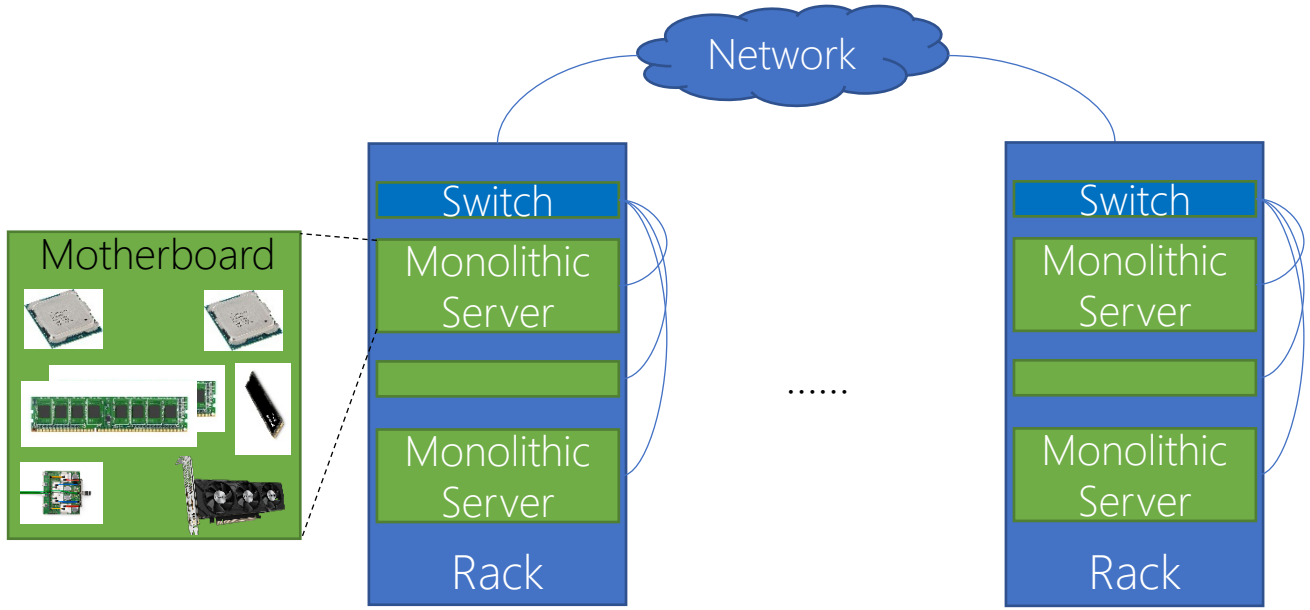
Drawbacks of Monolithic-Server-driven Datacenter



Drawbacks of Monolithic-Server-driven Datacenter

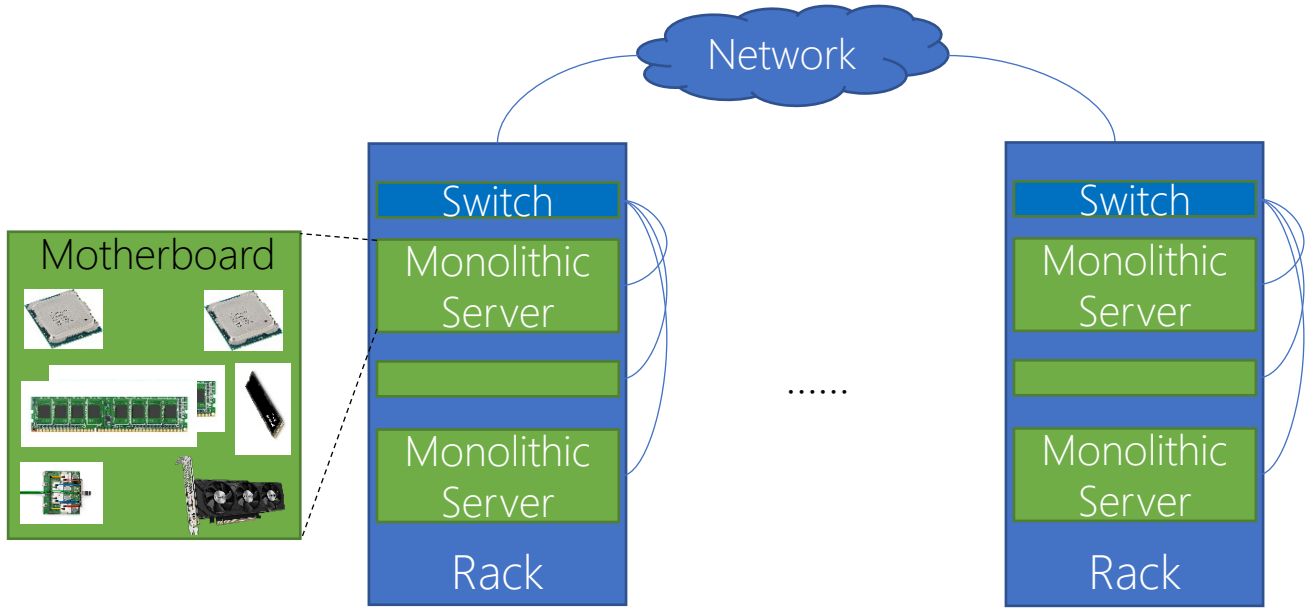


Drawbacks of Monolithic-Server-driven Datacenter



Monolithic-server-driven Datacenter Architecture

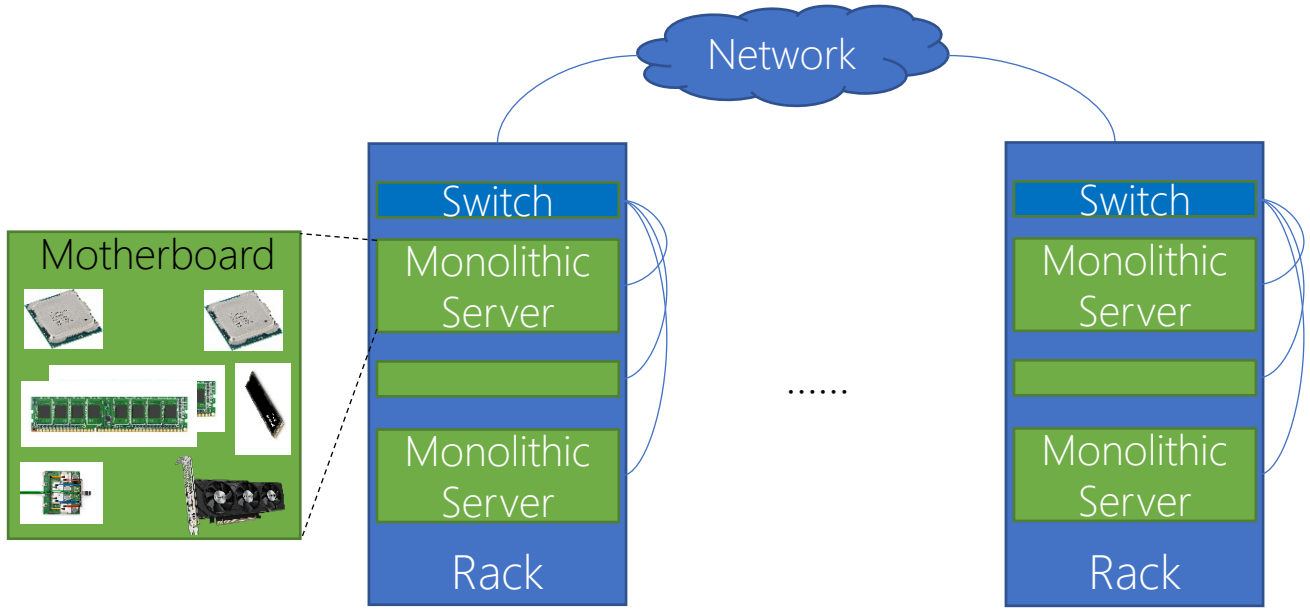
Drawbacks of Monolithic-Server-driven Datacenter



Monolithic-server-driven Datacenter Architecture

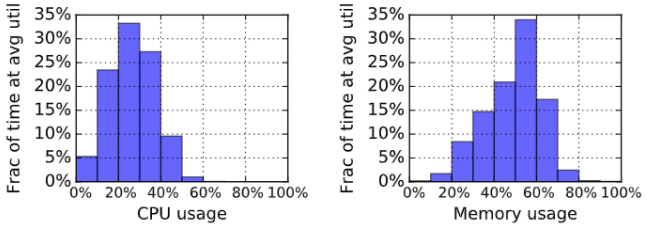
- Low resource utilization due to the mismatch between diverse workload and fixed server-resource configurations.

Drawbacks of Monolithic-Server-driven Datacenter



Monolithic-server-driven Datacenter Architecture

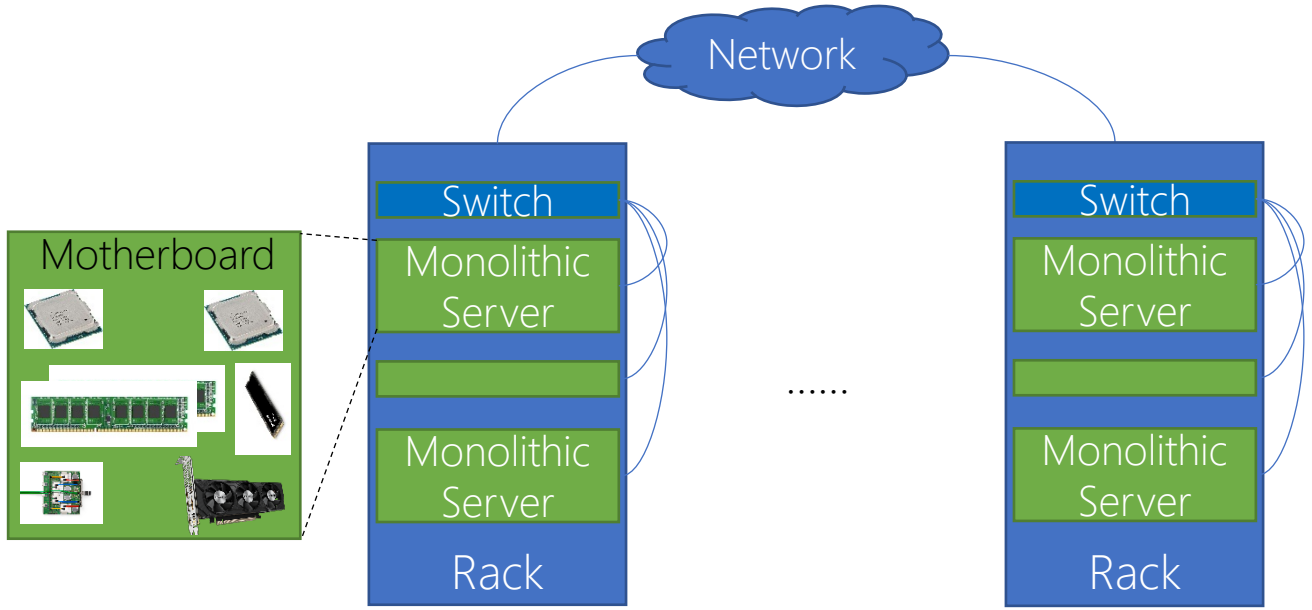
- Low resource utilization due to the mismatch between diverse workload and fixed server-resource configurations.



(a) Average CPU usage. (b) Average memory usage.

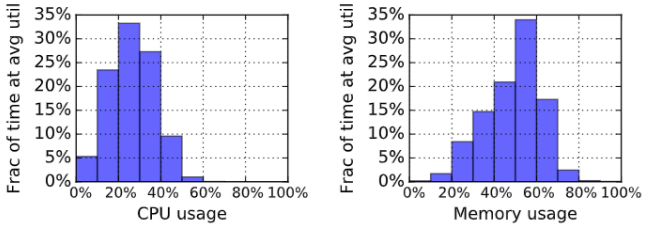
Alibaba cluster's CPU and memory usage [2]

Drawbacks of Monolithic-Server-driven Datacenter



Monolithic-server-driven Datacenter Architecture

- Low resource utilization due to the mismatch between diverse workload and fixed server-resource configurations.

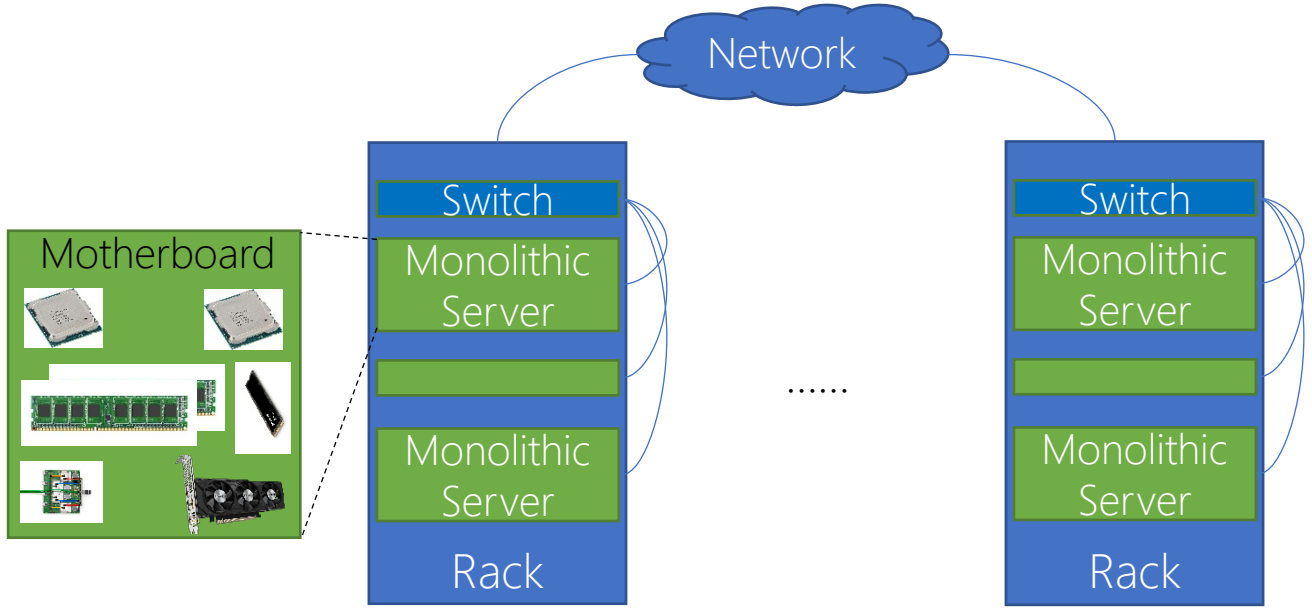


(a) Average CPU usage. (b) Average memory usage.

Alibaba cluster's CPU and memory usage [2]

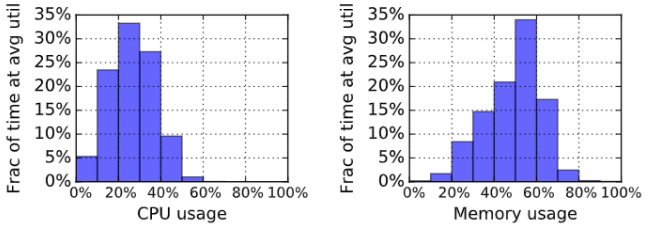
- Inflexible for independent scaling due to physical constraints

Drawbacks of Monolithic-Server-driven Datacenter



Monolithic-server-driven Datacenter Architecture

- Low resource utilization due to the mismatch between diverse workload and fixed server-resource configurations.



(a) Average CPU usage. (b) Average memory usage.

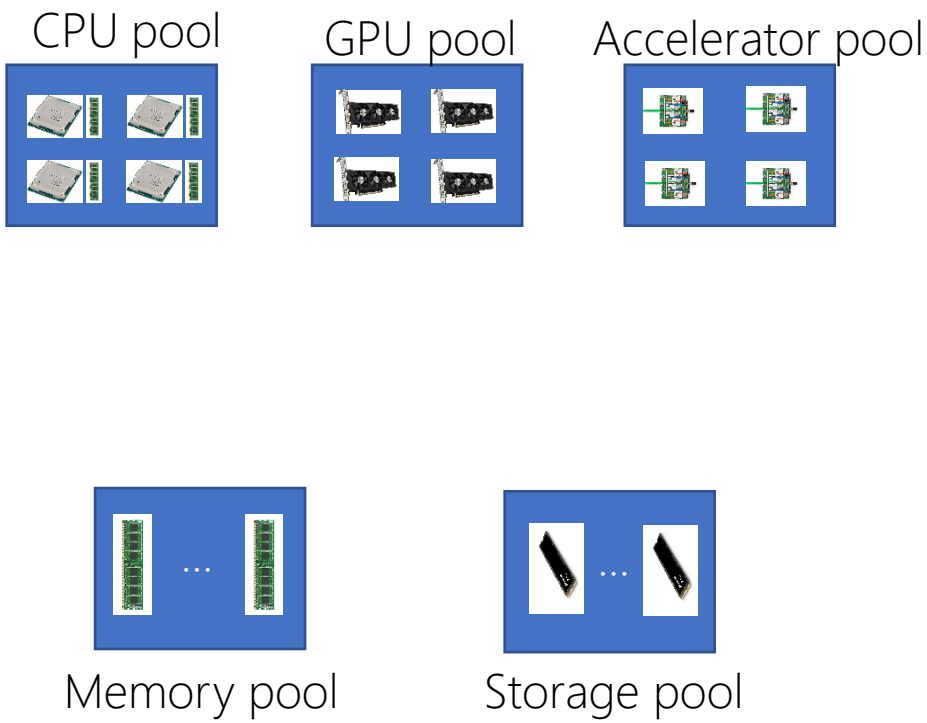
Alibaba cluster's CPU and memory usage [2]

- Inflexible for independent scaling due to physical constraints

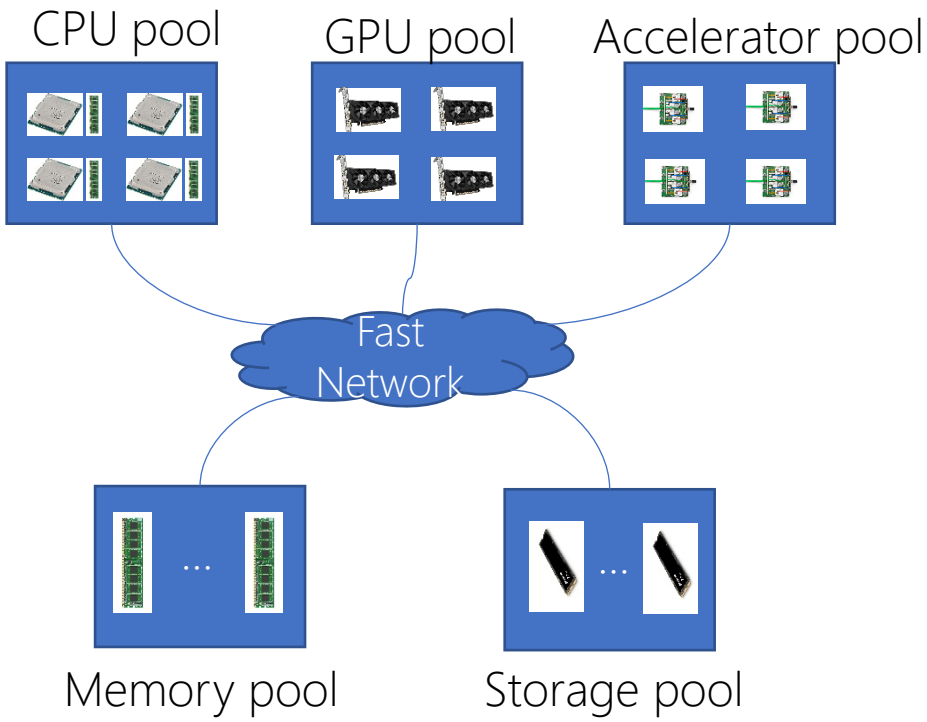
Monolithic server couples resources on a single motherboard, underutilized, inflexible.

Disaggregated Datacenter is the Path Forward

Disaggregated Datacenter is the Path Forward

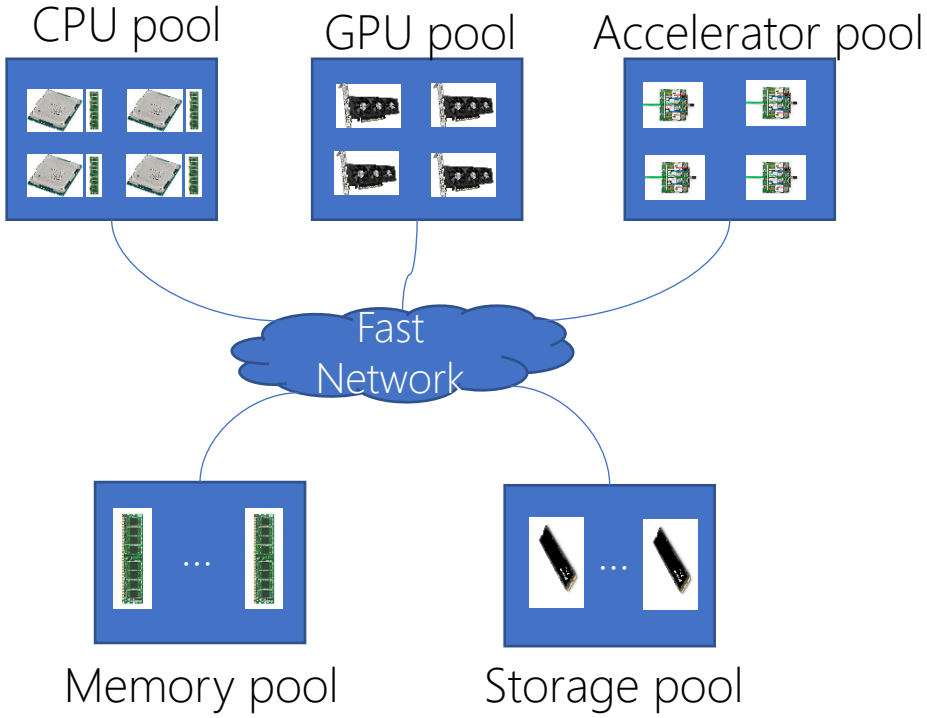


Disaggregated Datacenter is the Path Forward



Disaggregated Datacenter (DDC) Architecture

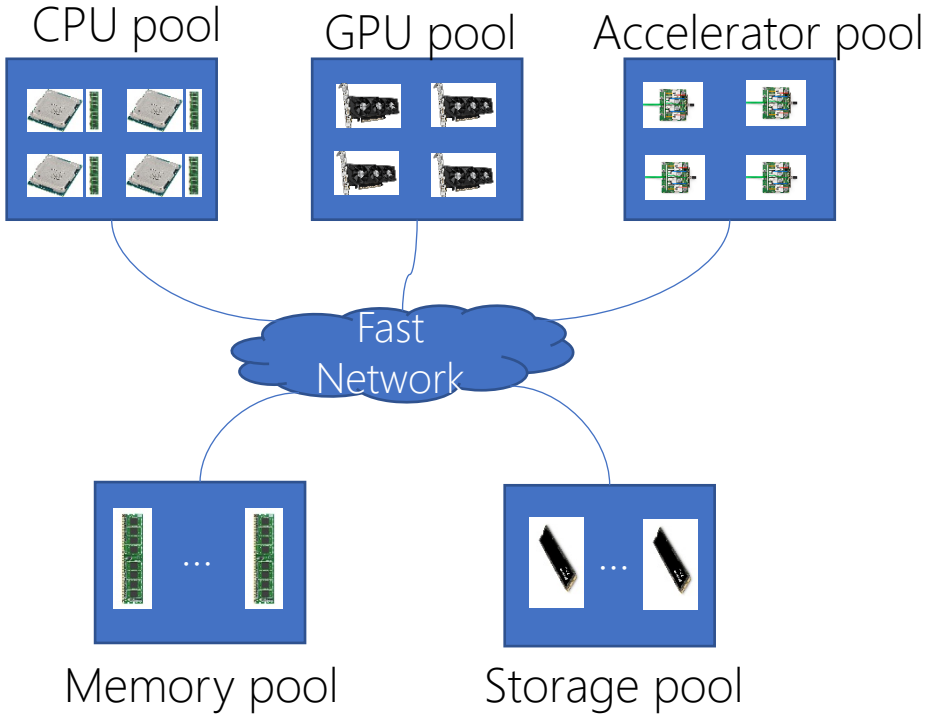
Disaggregated Datacenter is the Path Forward



Disaggregated Datacenter (DDC) Architecture

- Greater resource utilization due to flexibility in resource configurations

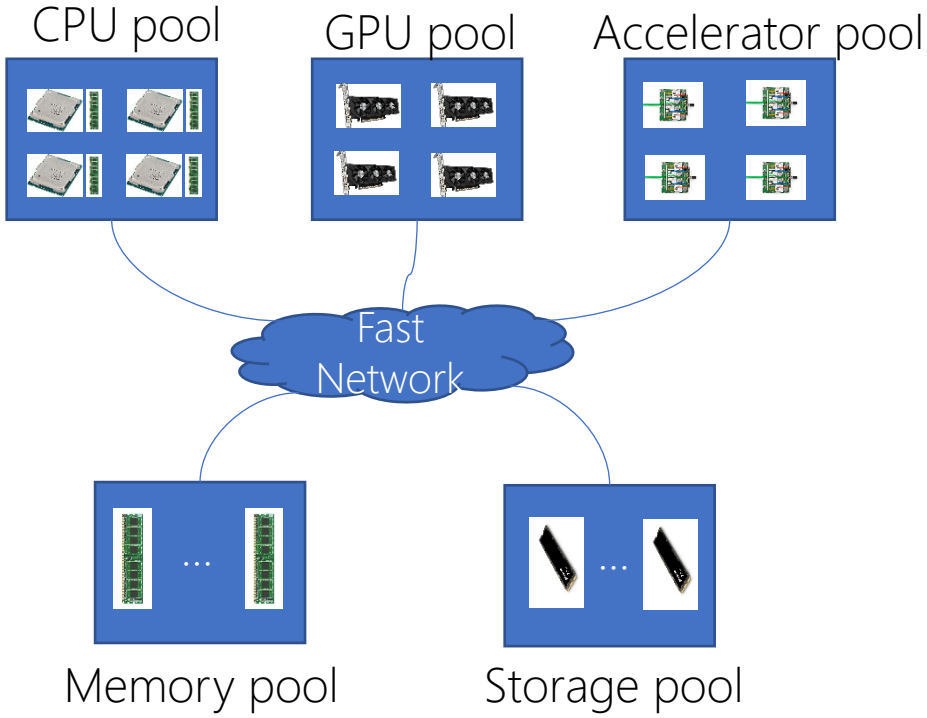
Disaggregated Datacenter is the Path Forward



Disaggregated Datacenter (DDC) Architecture

- Greater resource utilization due to flexibility in resource configurations
- Independent scaling and upgrading due to modularity

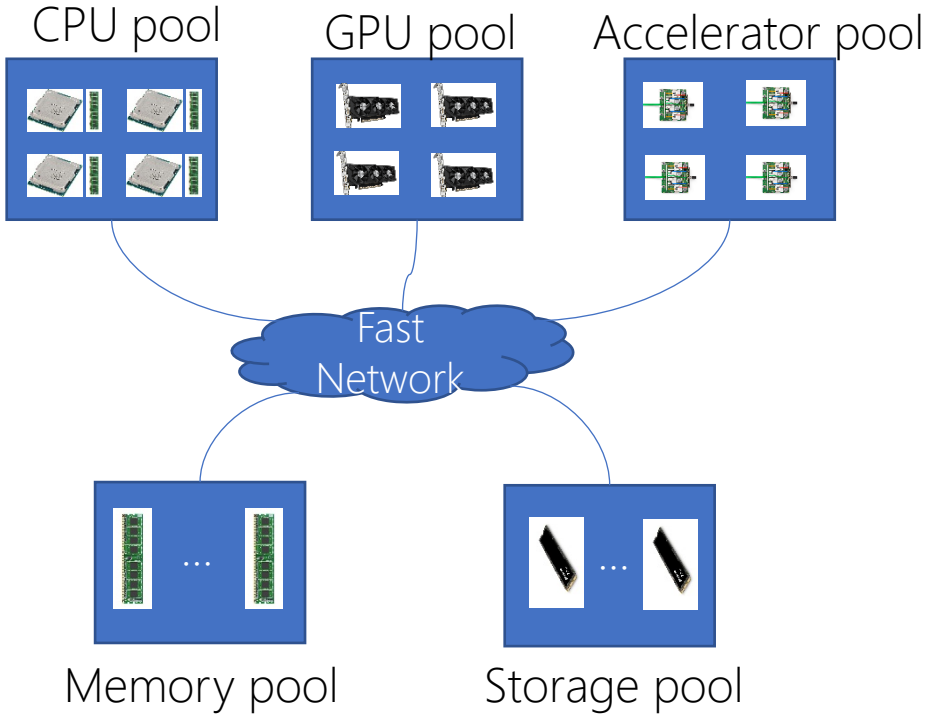
Disaggregated Datacenter is the Path Forward



Disaggregated Datacenter (DDC) Architecture

- Greater resource utilization due to flexibility in resource configurations
- Independent scaling and upgrading due to modularity
- Better energy efficiency and lower OPEX due to better resource configuration

Disaggregated Datacenter is the Path Forward



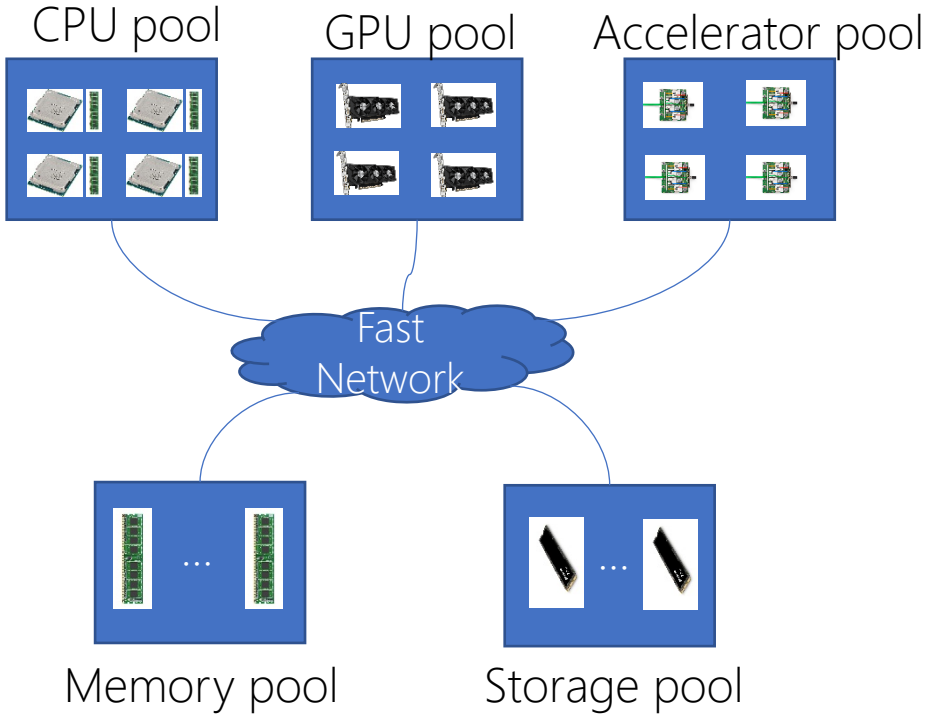
Disaggregated Datacenter (DDC) Architecture

- Greater resource utilization due to flexibility in resource configurations
- Independent scaling and upgrading due to modularity
- Better energy efficiency and lower OPEX due to better resource configuration

Examples:

- Alibaba's LegoBase[3]: database based on memory disaggregation

Disaggregated Datacenter is the Path Forward

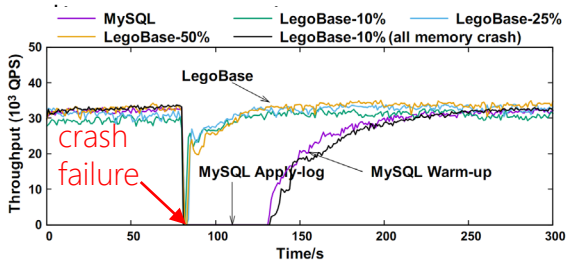


Disaggregated Datacenter (DDC) Architecture

- Greater resource utilization due to flexibility in resource configurations
- Independent scaling and upgrading due to modularity
- Better energy efficiency and lower OPEX due to better resource configuration

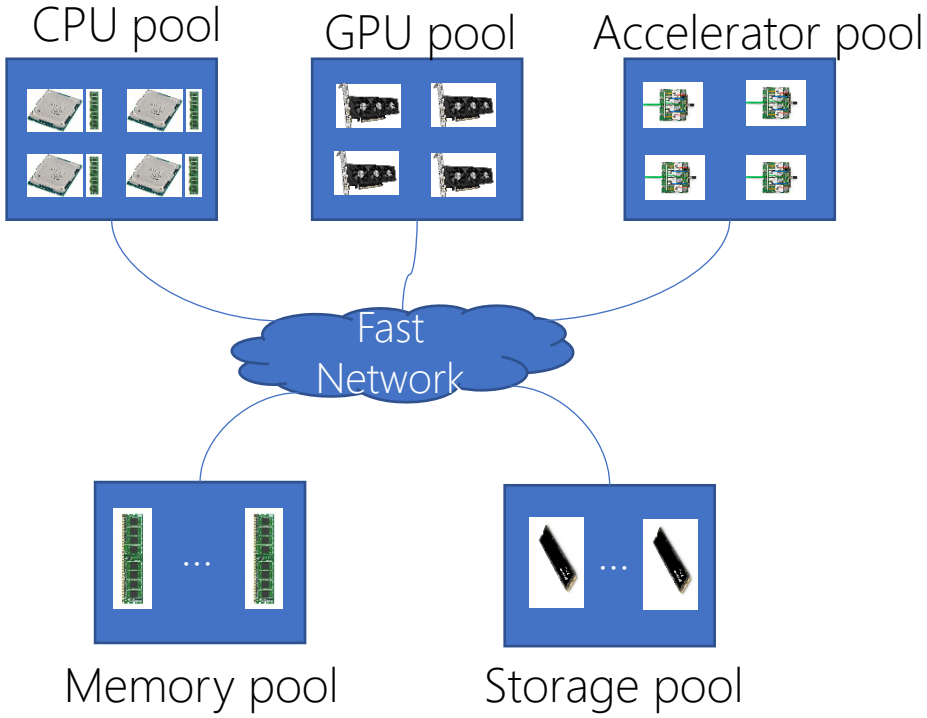
Examples:

- Alibaba's LegoBase[3]: database based on memory



Faster recovery in case of crash

Disaggregated Datacenter is the Path Forward

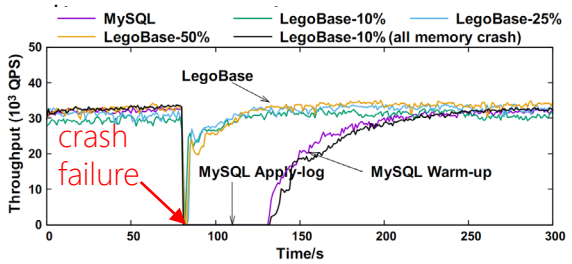


Disaggregated Datacenter (DDC) Architecture

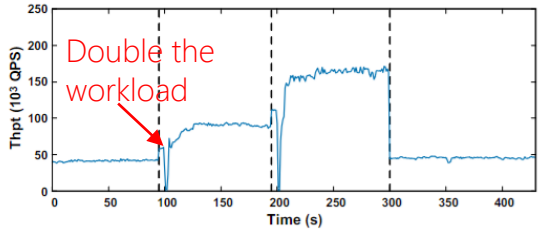
- Greater resource utilization due to flexibility in resource configurations
- Independent scaling and upgrading due to modularity
- Better energy efficiency and lower OPEX due to better resource configuration

Examples:

- Alibaba's LegoBase[3]: database based on memory

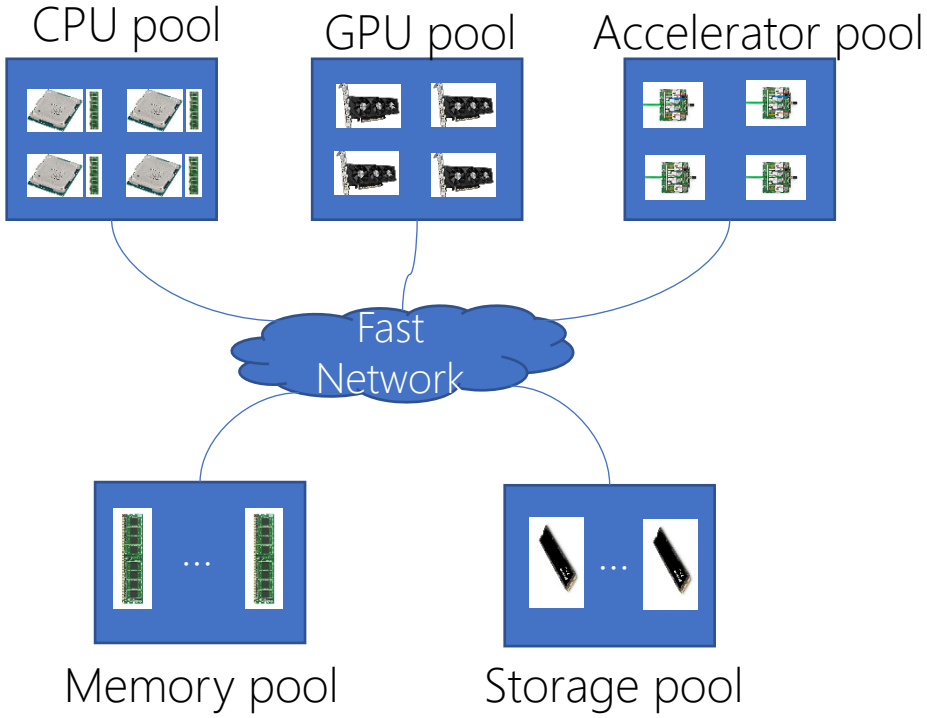


Faster recovery in case of crash



Fast scaling in terms of a sudden workload increase

Disaggregated Datacenter is the Path Forward

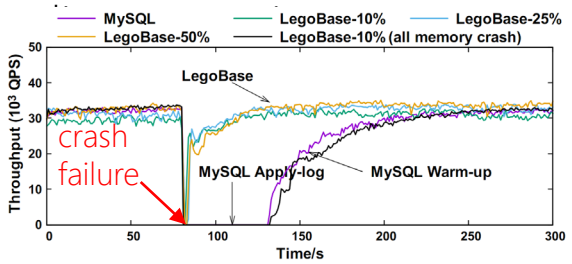


Disaggregated Datacenter (DDC) Architecture

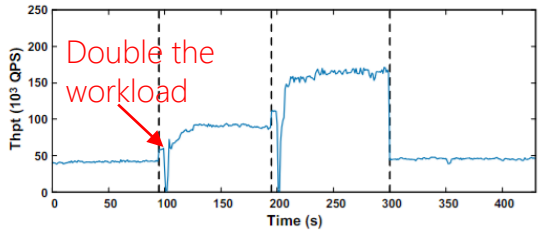
- Greater resource utilization due to flexibility in resource configurations
- Independent scaling and upgrading due to modularity
- Better energy efficiency and lower OPEX due to better resource configuration

Examples:

- Alibaba's LegoBase[3]: database based on memory



Faster recovery in case of crash

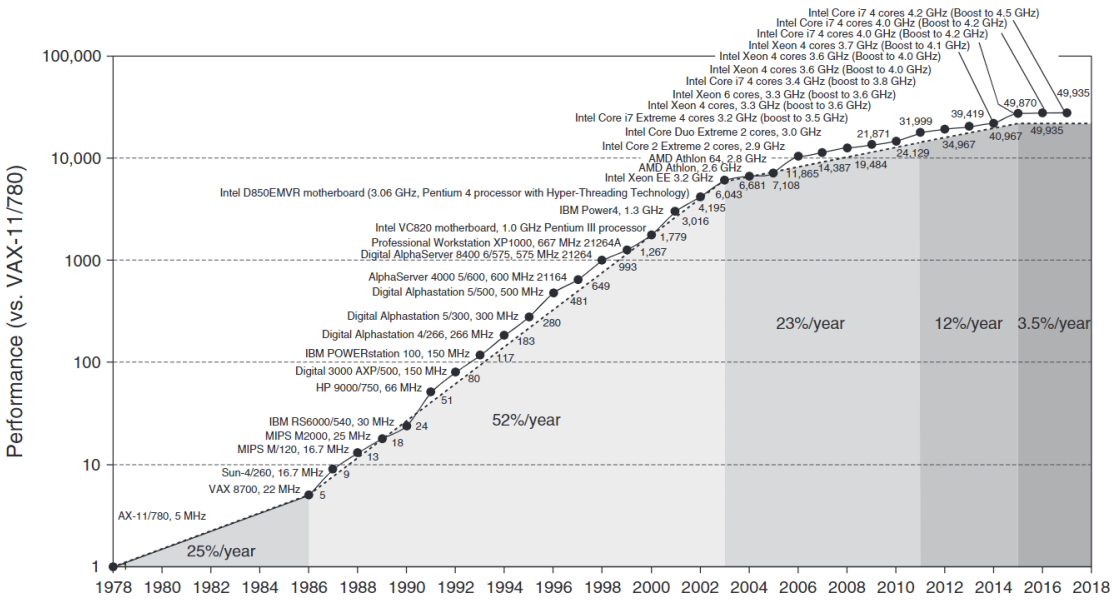


Fast scaling in terms of a sudden workload increase

Disaggregated datacenter decouples resources by separating, pooling and composing.

Accelerator Disaggregation

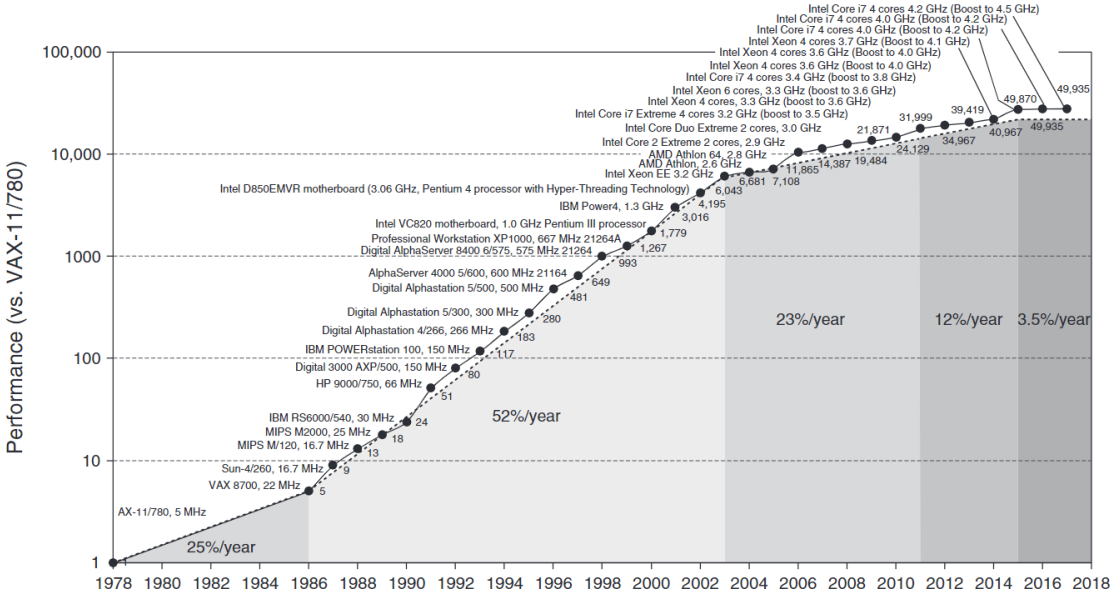
Accelerator Disaggregation



Growth in processor performance over 40 years[4]

Accelerator Disaggregation

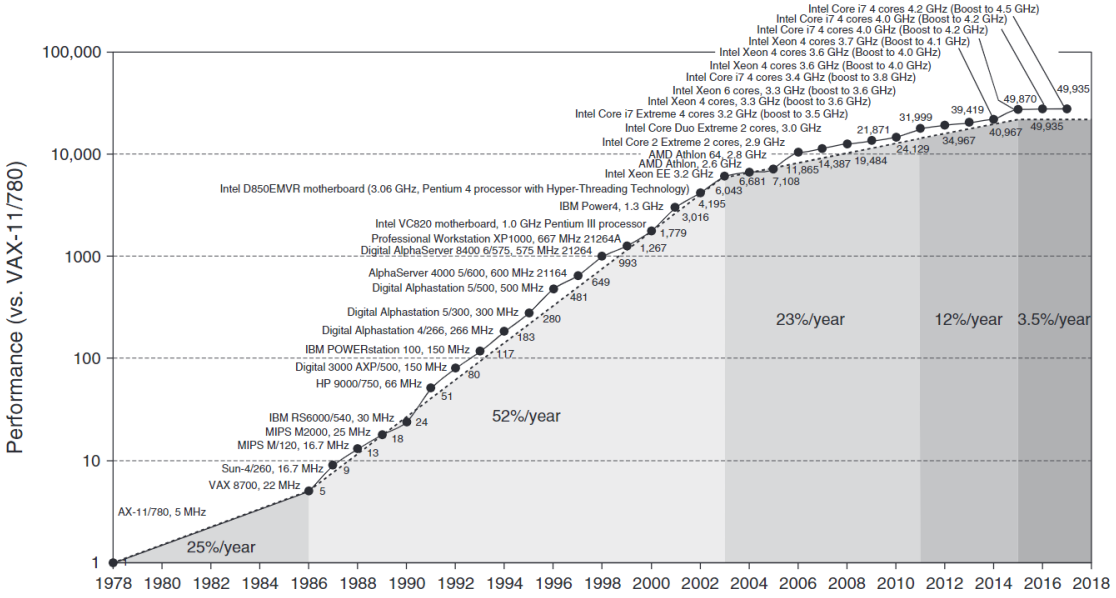
- Accelerators can continue scaling perf and perf/W
  - Speedup, high energy efficiency



Growth in processor performance over 40 years[4]

Accelerator Disaggregation

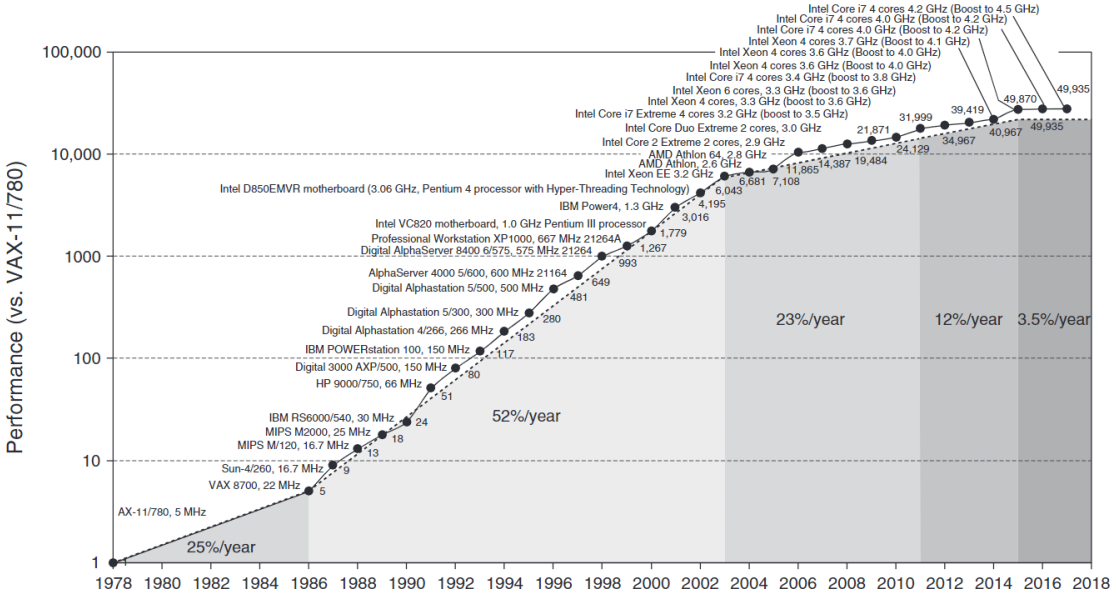
- Accelerators can continue scaling perf and perf/W
  - Speedup, high energy efficiency
- Accelerators are everywhere in the cloud.
  - Deep learning accelerator
  - Video transcoding accelerators
  - Database accelerators
  - Datacenter-tax accelerator (Compression/Decompression,…)



Growth in processor performance over 40 years[4]

Accelerator Disaggregation

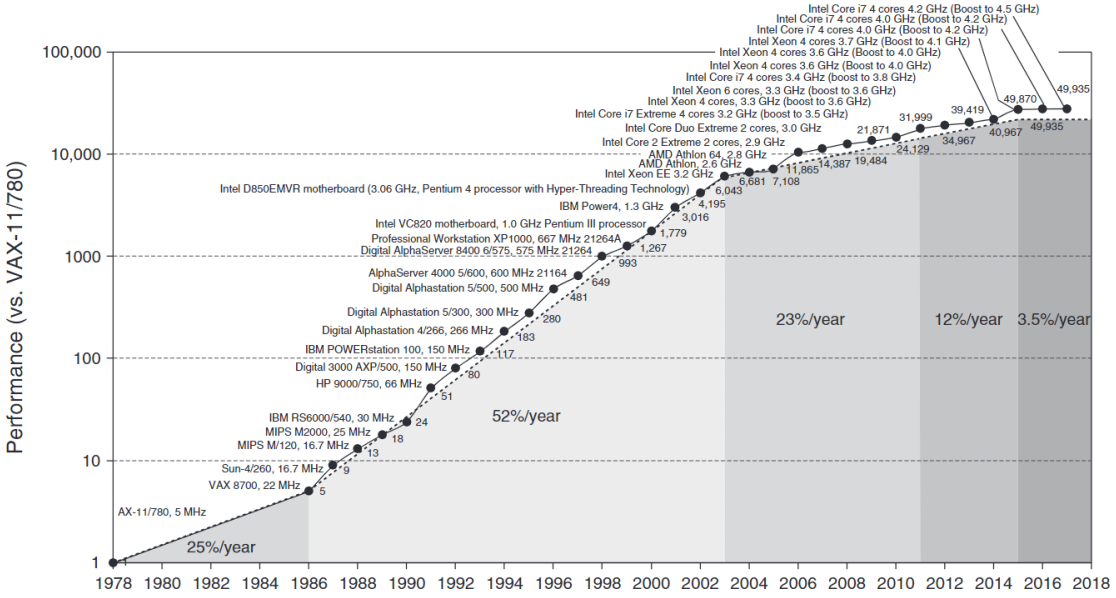
- Accelerators can continue scaling perf and perf/W
  - Speedup, high energy efficiency
- Accelerators are everywhere in the cloud.
  - Deep learning accelerator
  - Video transcoding accelerators
  - Database accelerators
  - Datacenter-tax accelerator (Compression/Decompression,…)
- Disaggregation for accelerators: enables scaling-out and ubiquitous acceleration service, independently of server hardware.



Growth in processor performance over 40 years[4]

Accelerator Disaggregation

- Accelerators can continue scaling perf and perf/W
  - Speedup, high energy efficiency
- Accelerators are everywhere in the cloud.
  - Deep learning accelerator
  - Video transcoding accelerators
  - Database accelerators
  - Datacenter-tax accelerator (Compression/Decompression,…)
- Disaggregation for accelerators: enables scaling-out and ubiquitous acceleration service, independently of server hardware.



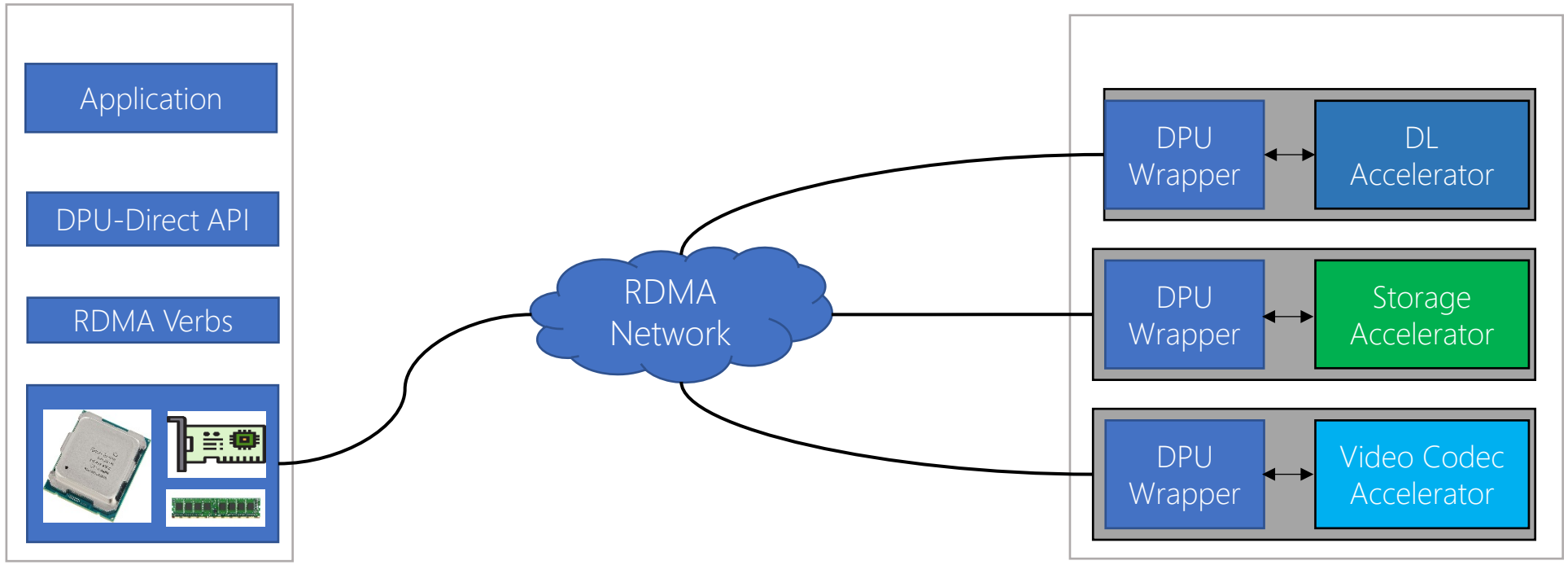
Growth in processor performance over 40 years[4]

Accelerator disaggregation: make accelerators available to every user.

02 System Overview and Mechanisms

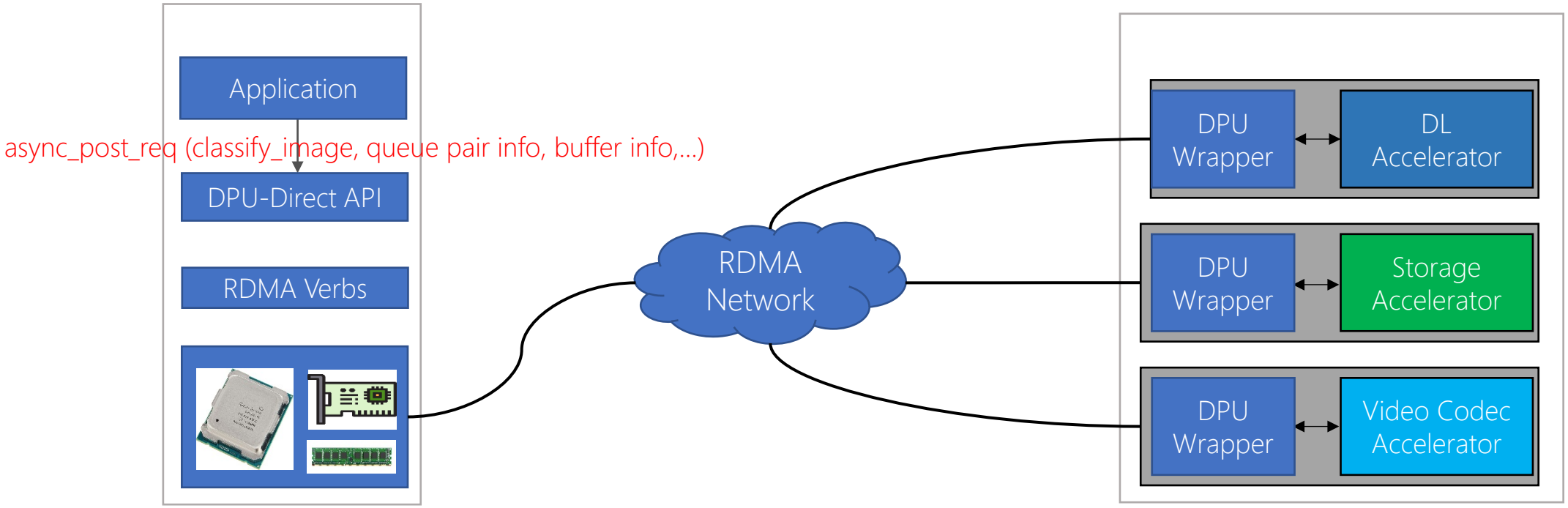
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



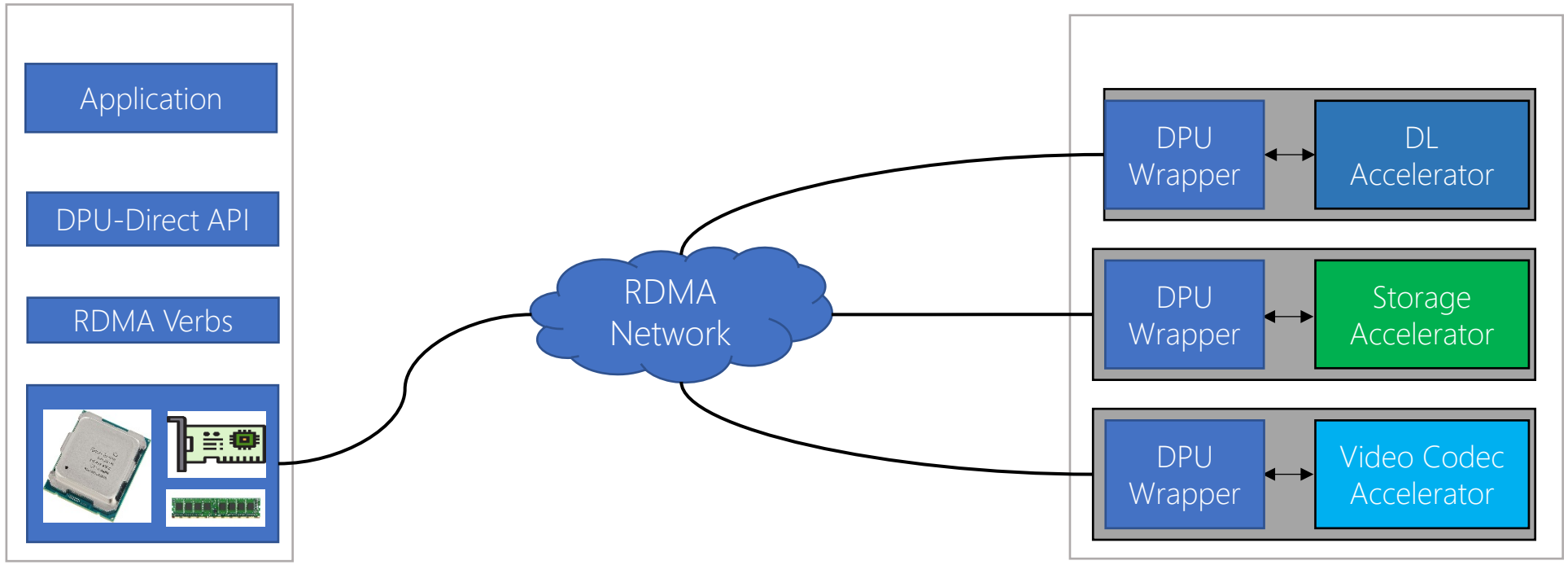
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



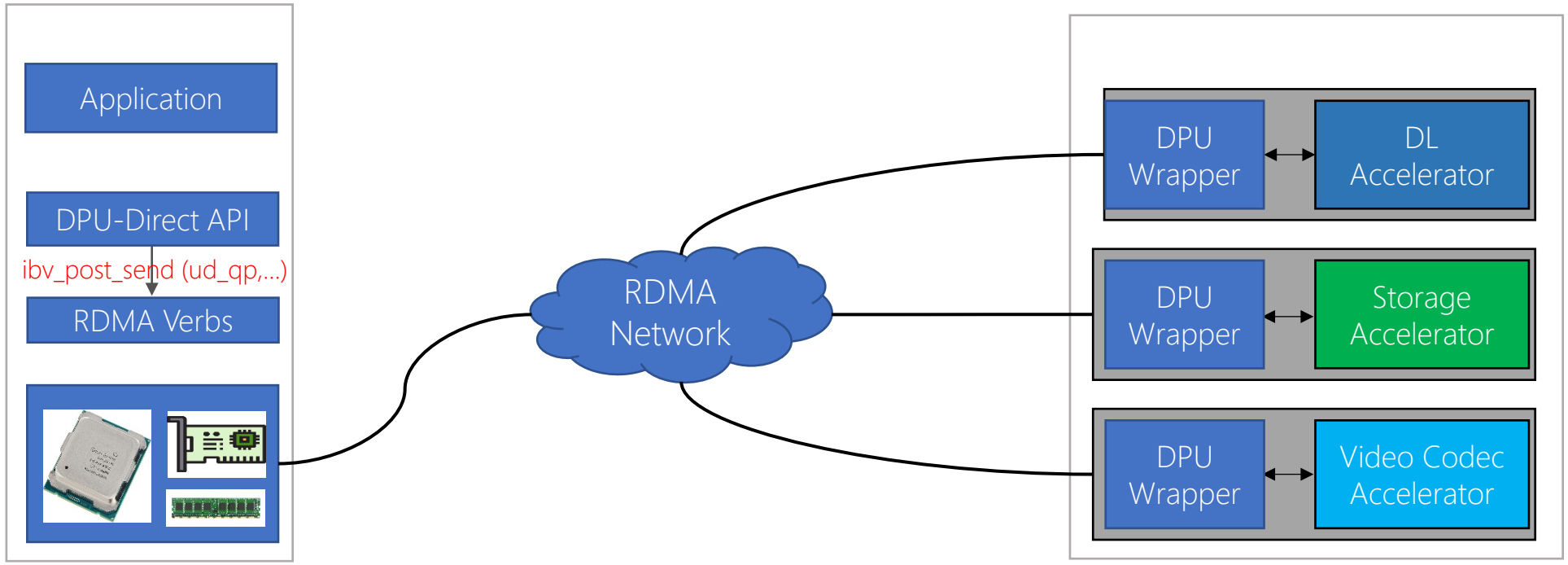
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



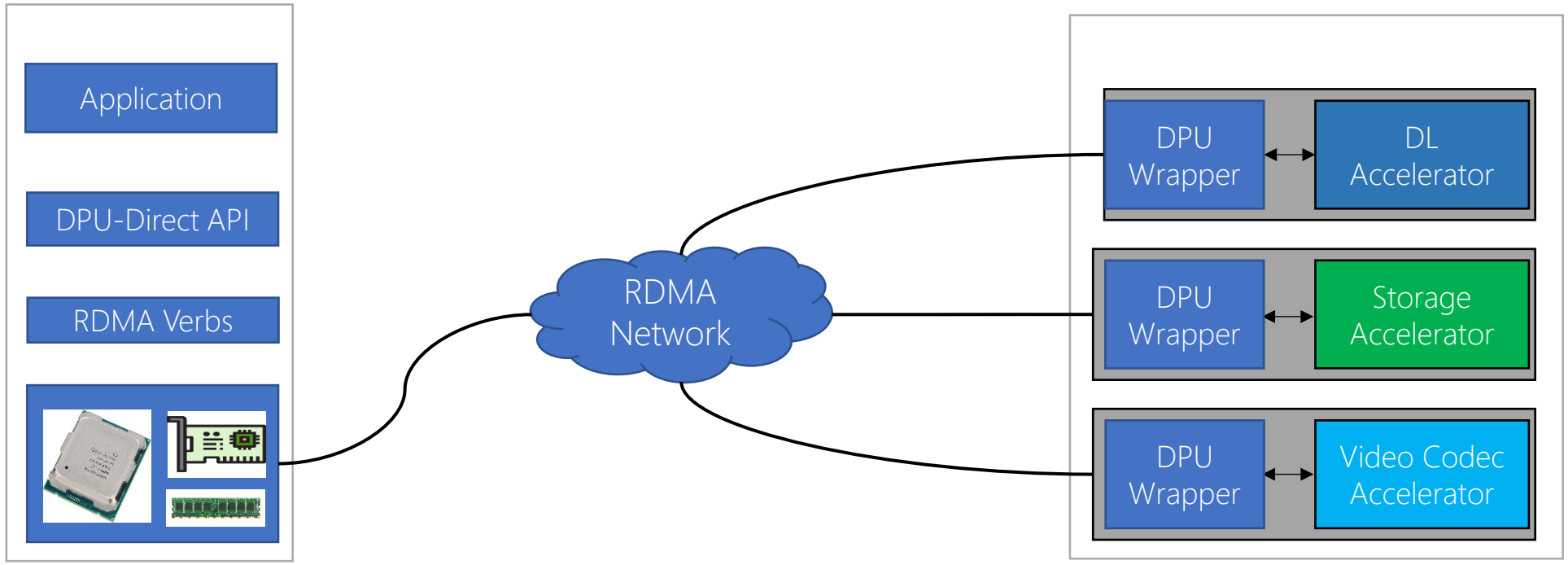
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



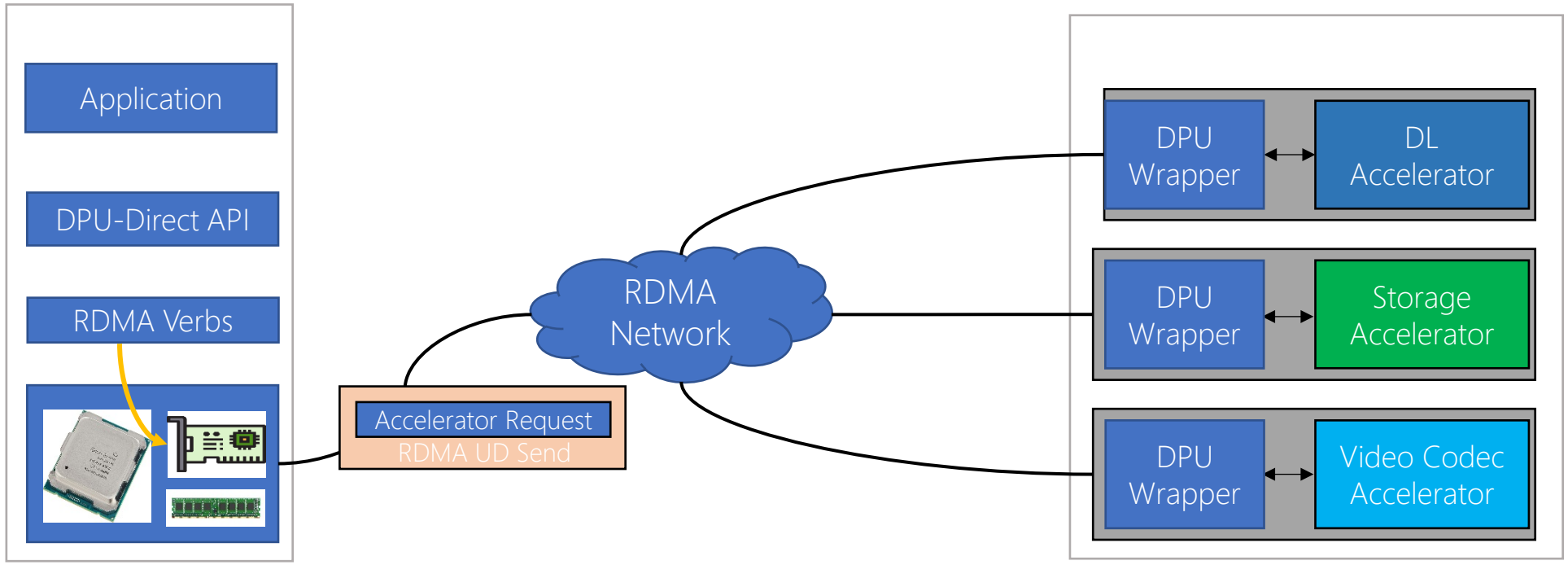
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



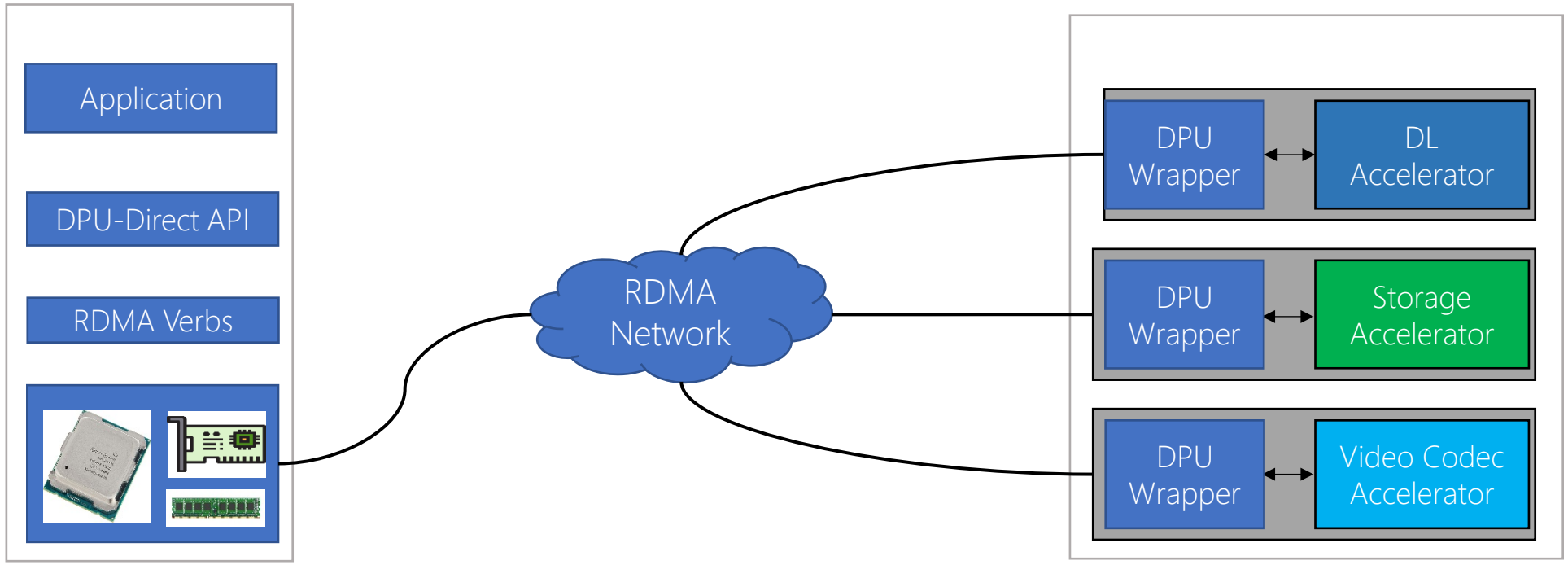
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



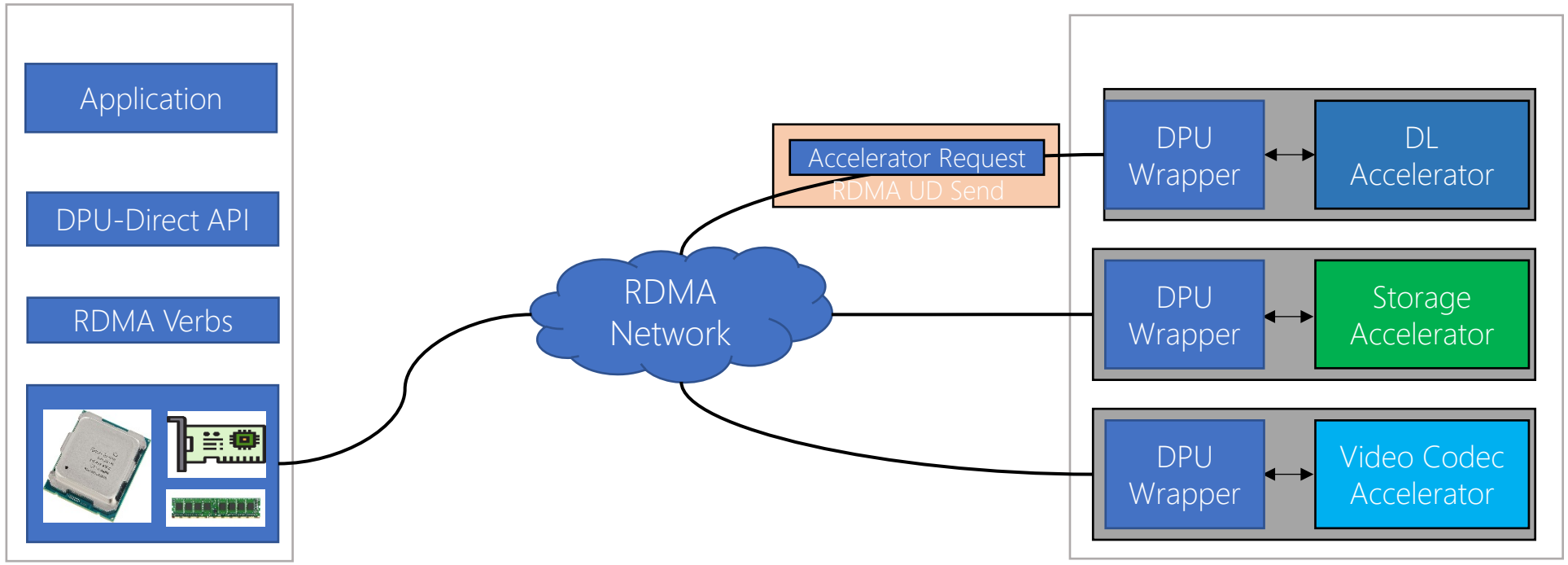
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



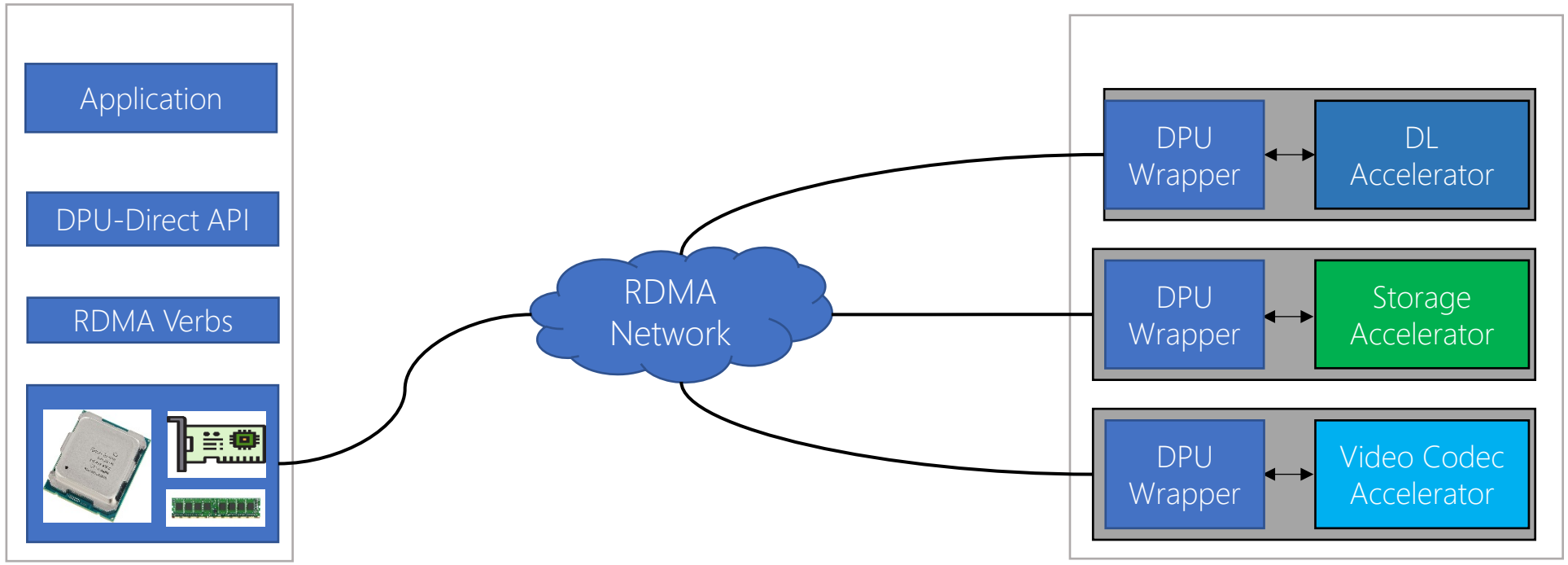
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



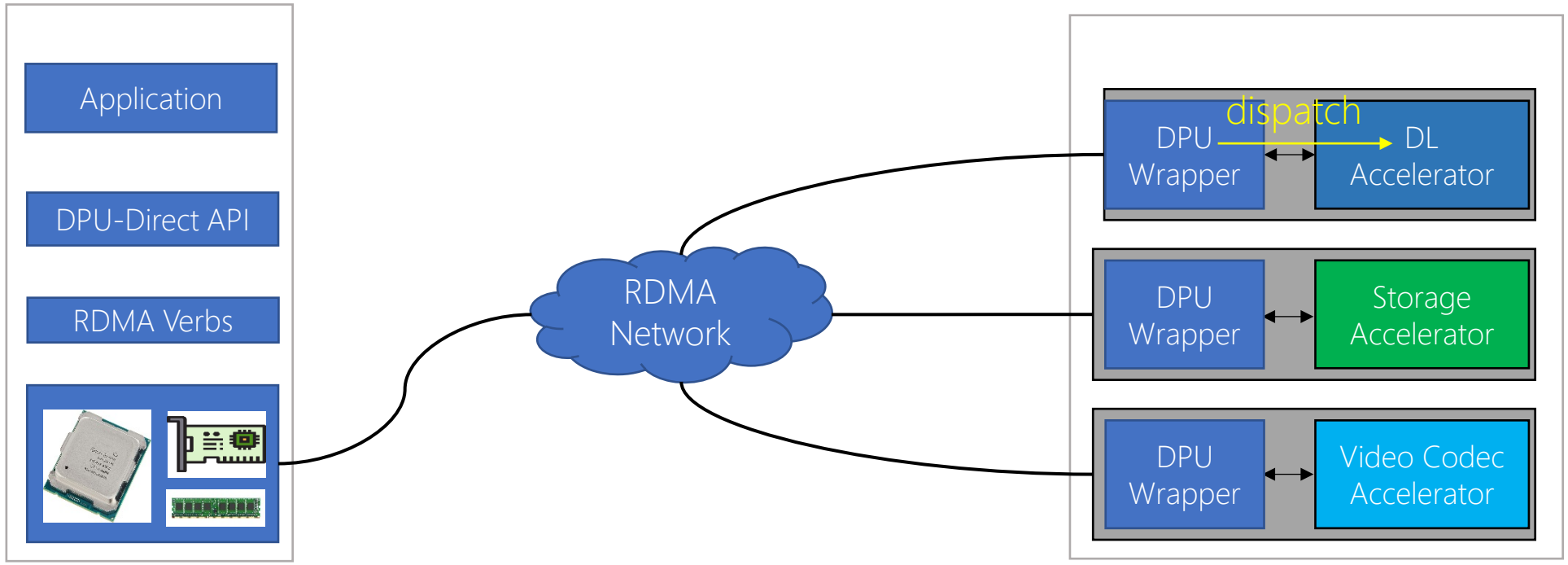
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



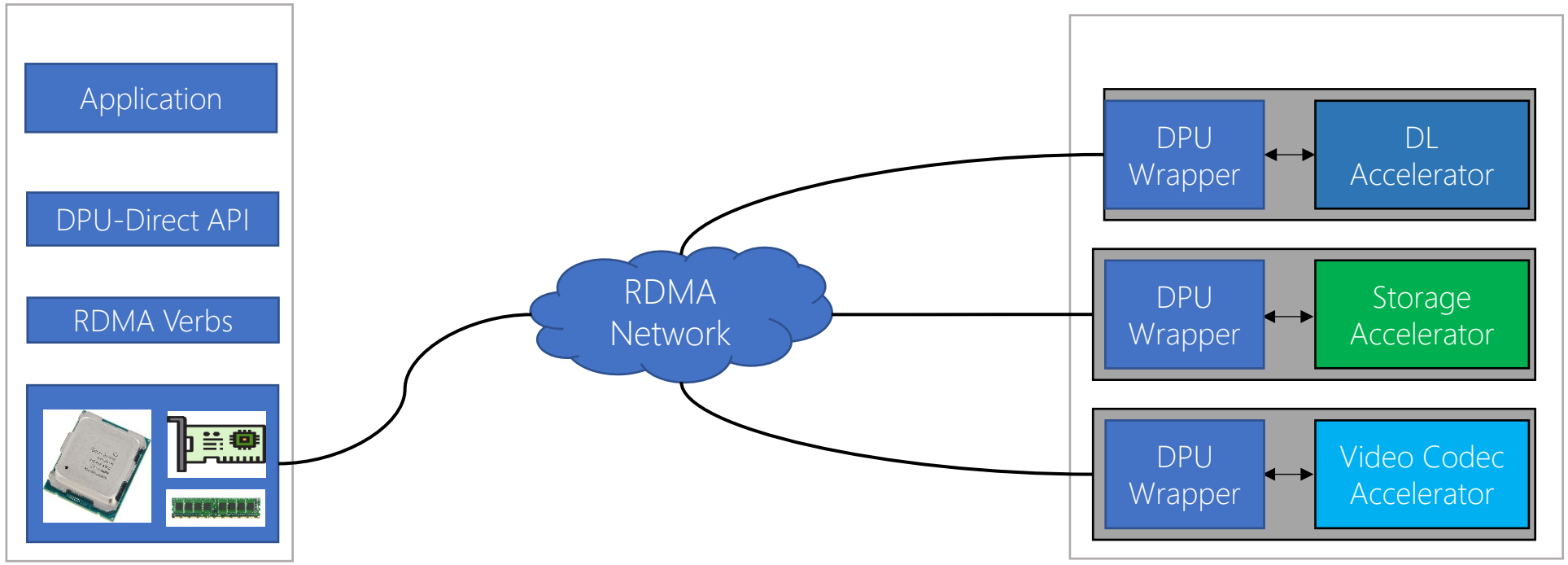
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



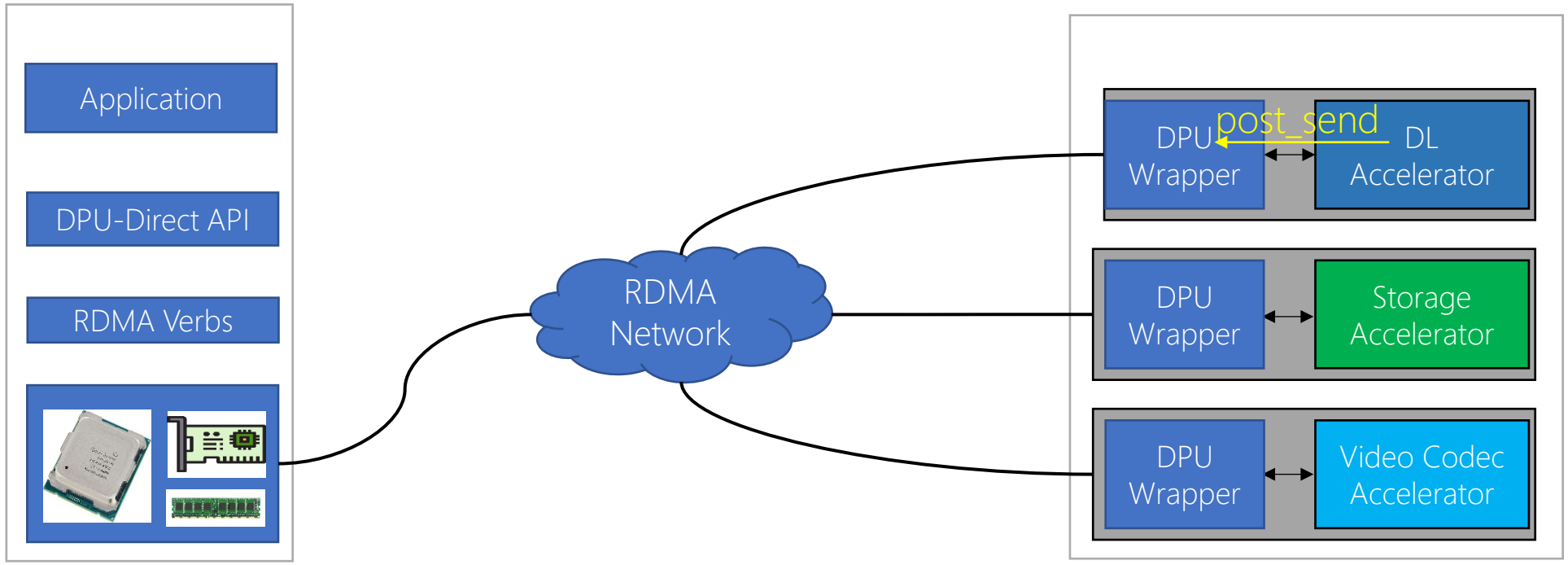
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



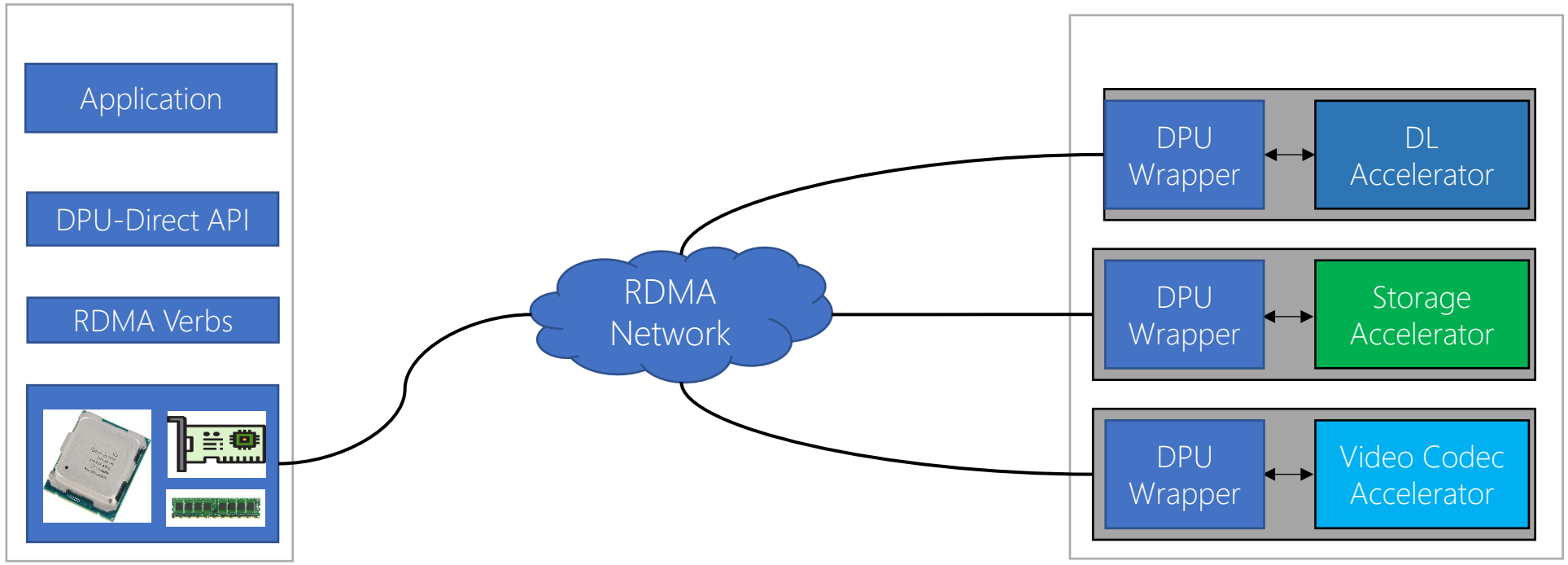
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



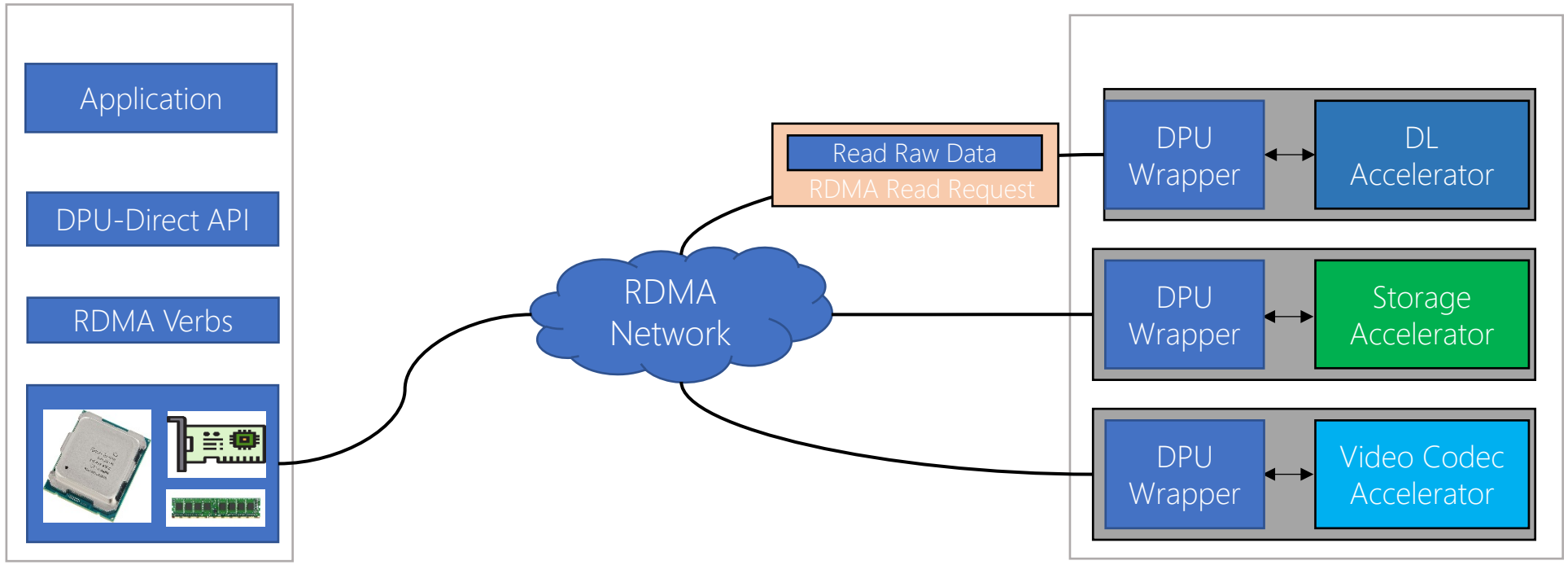
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



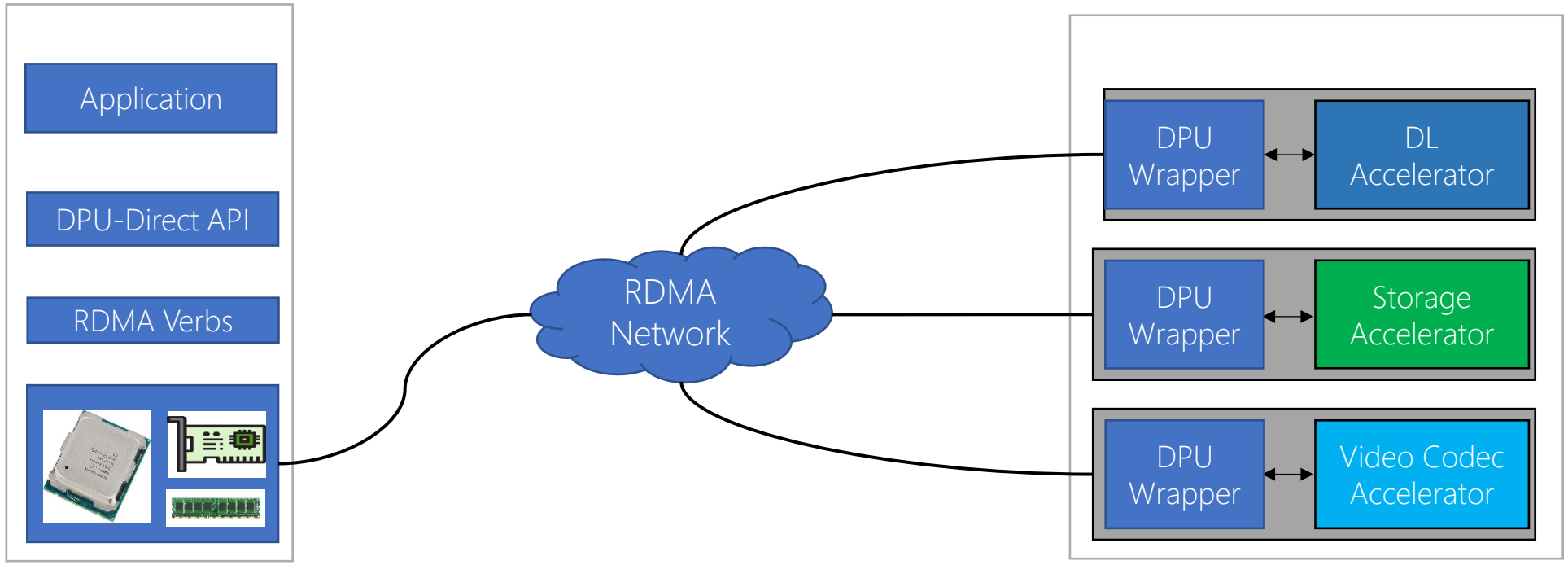
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



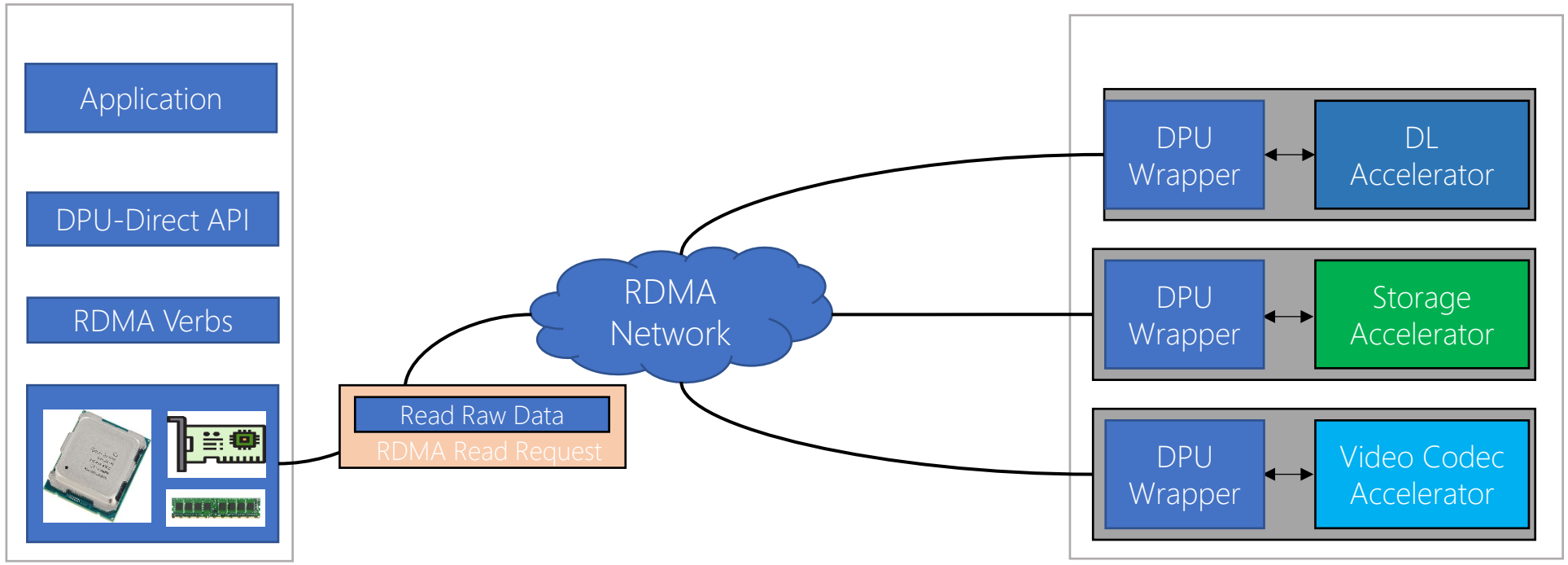
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



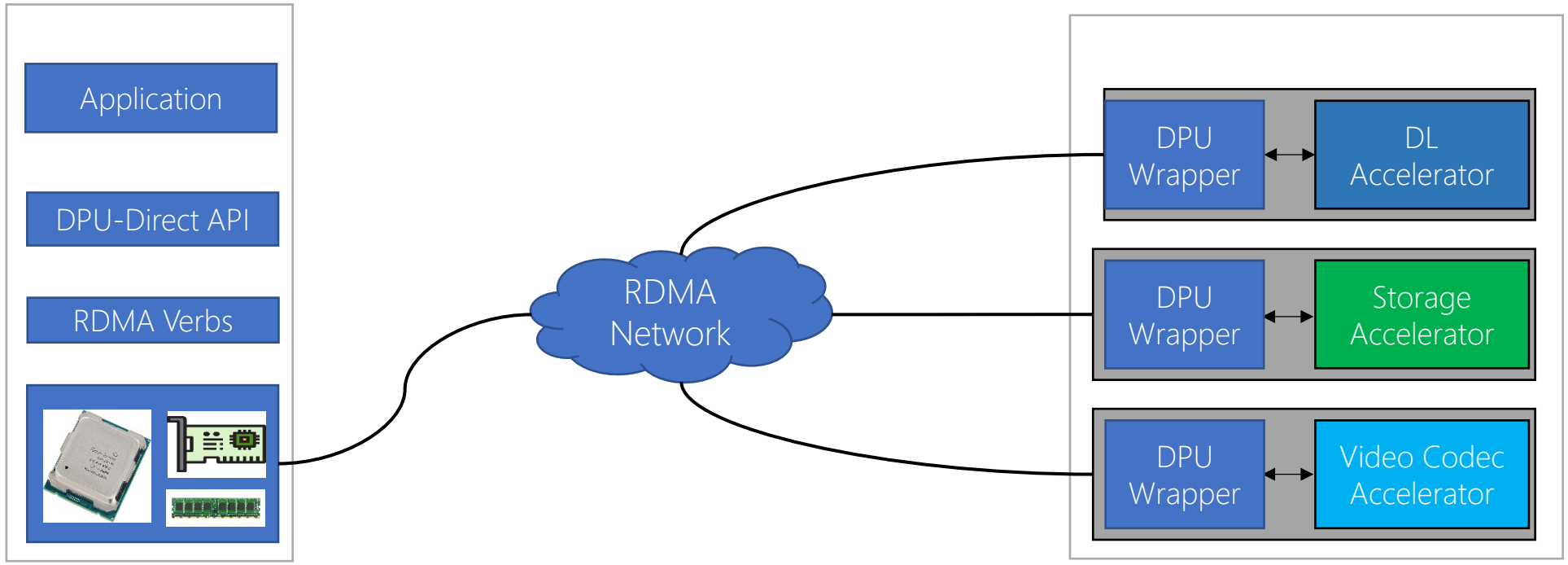
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



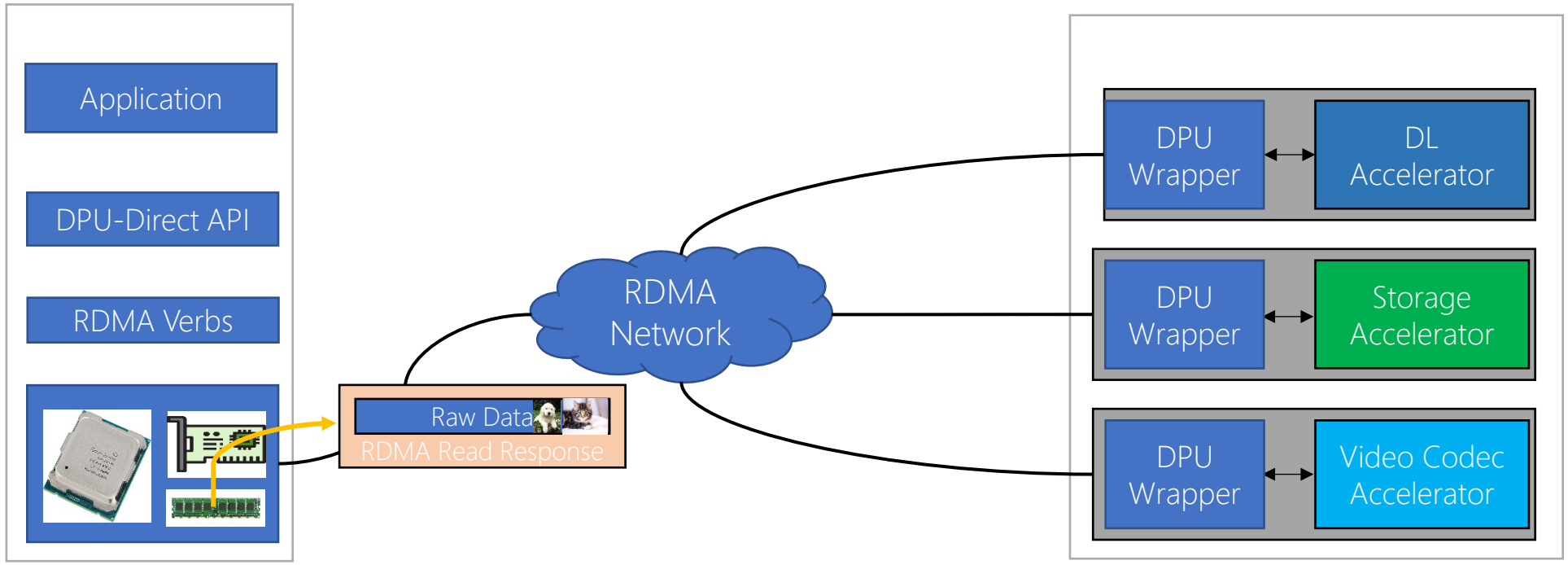
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



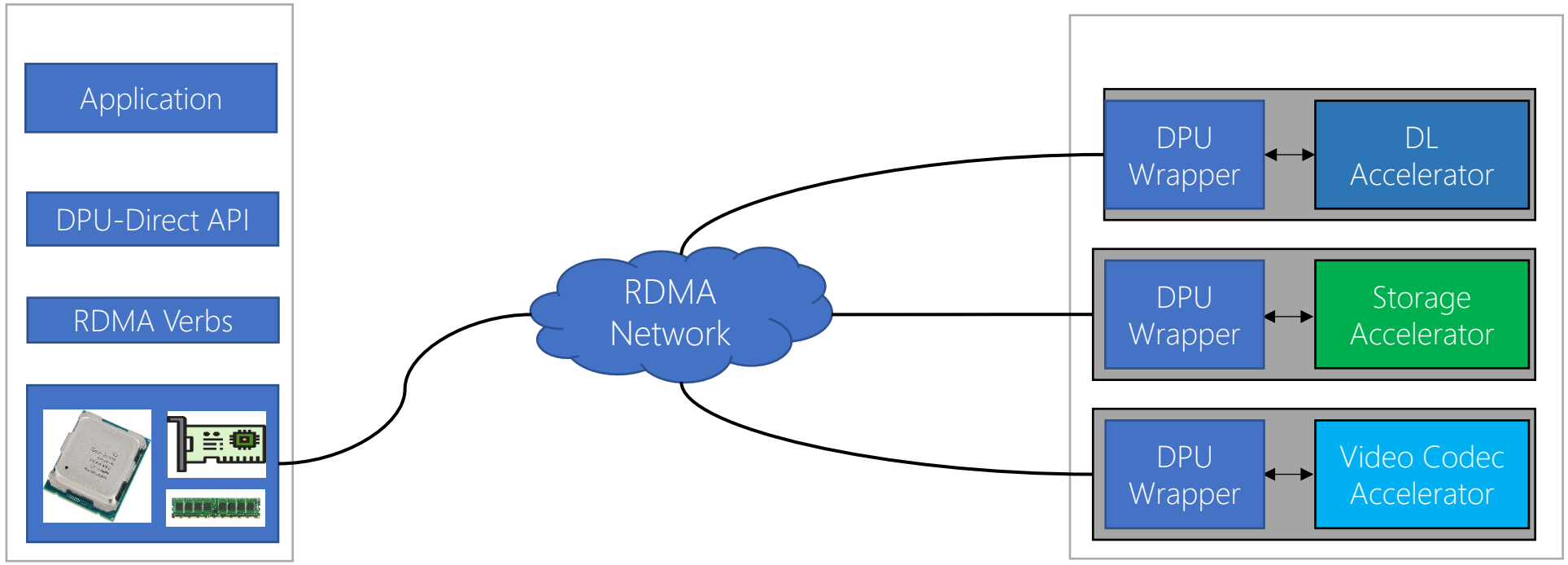
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



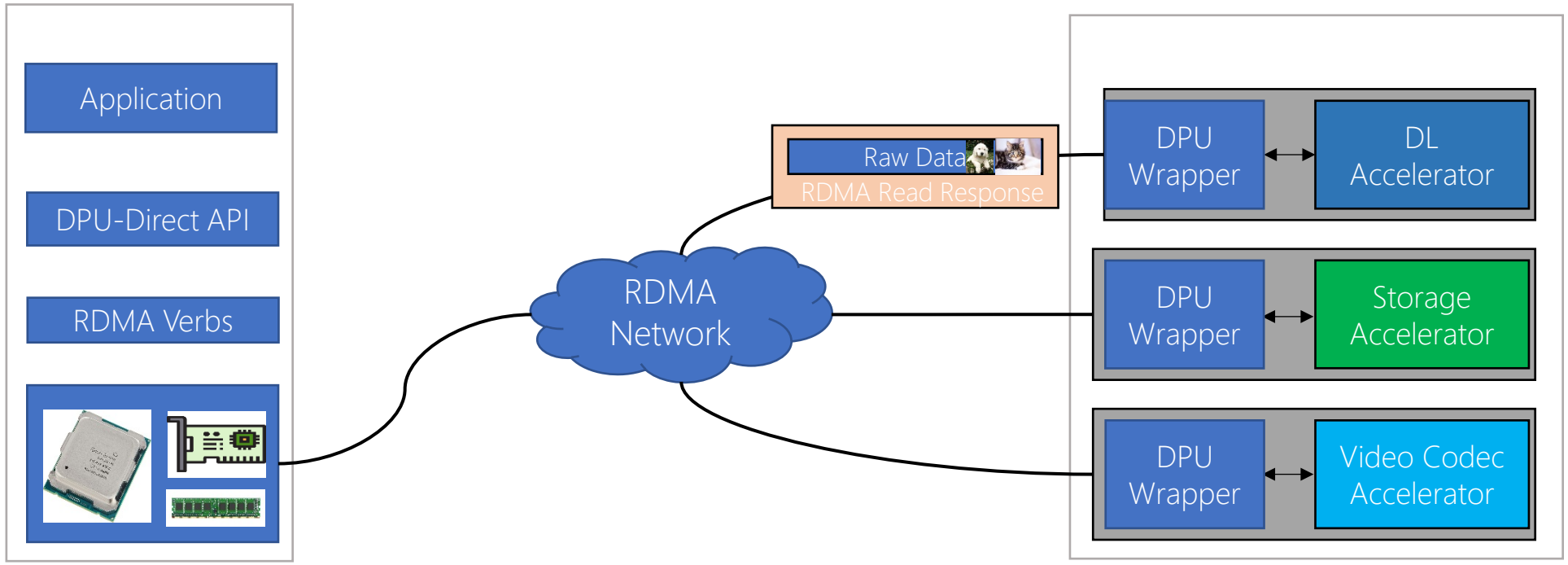
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



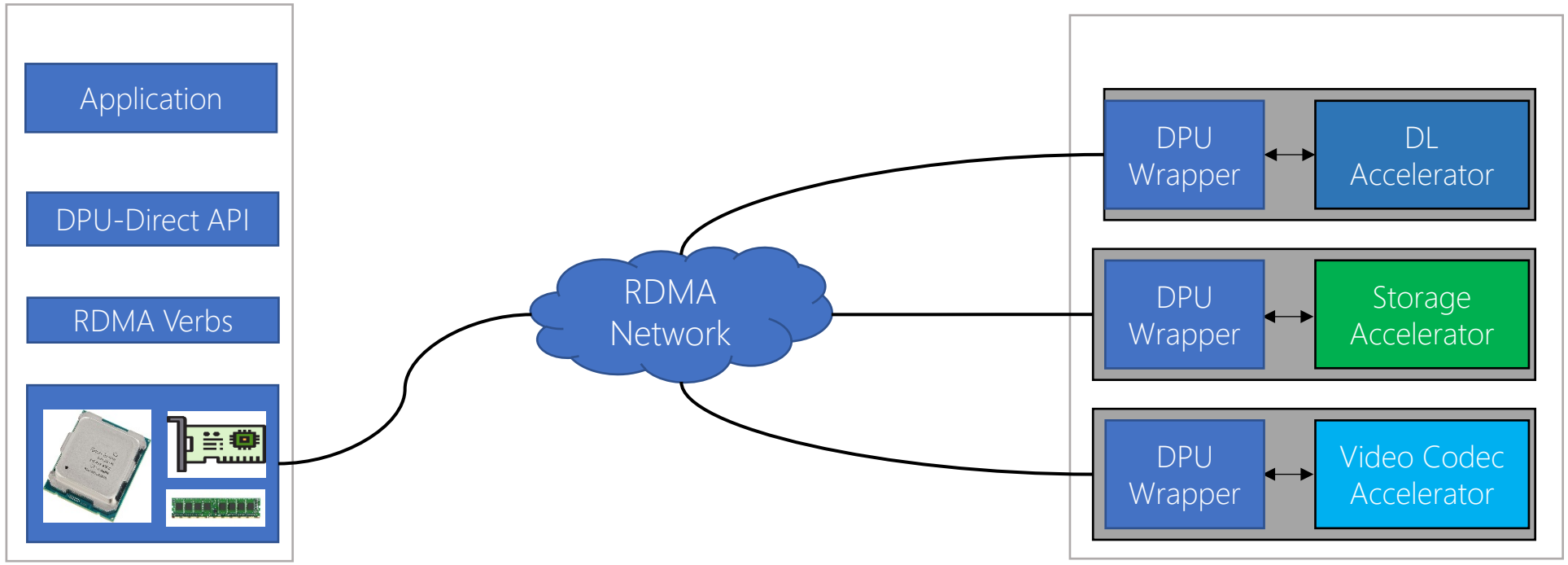
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



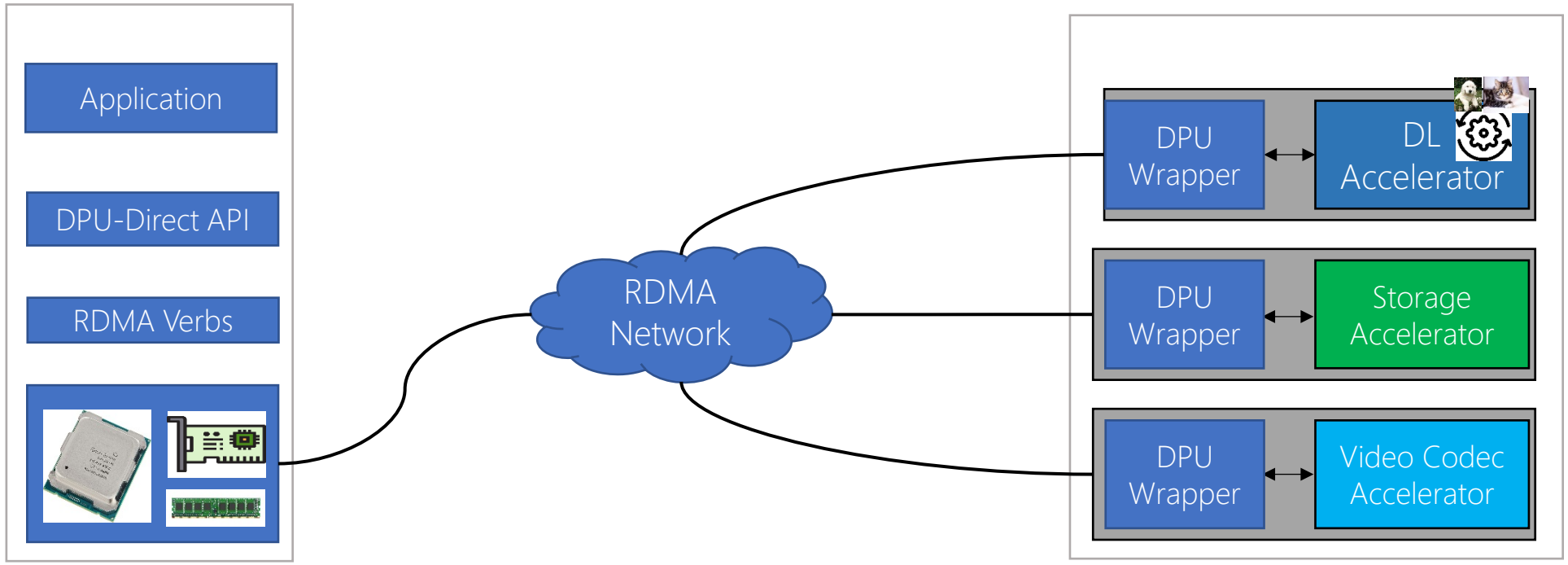
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



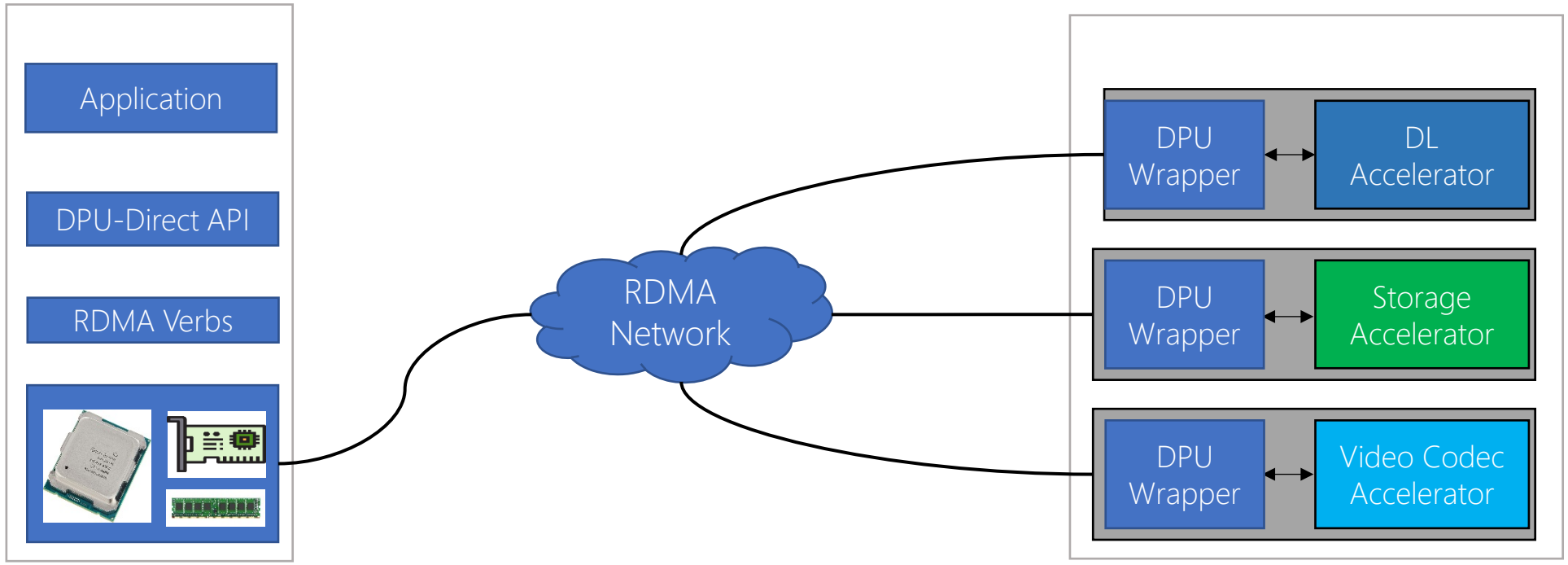
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



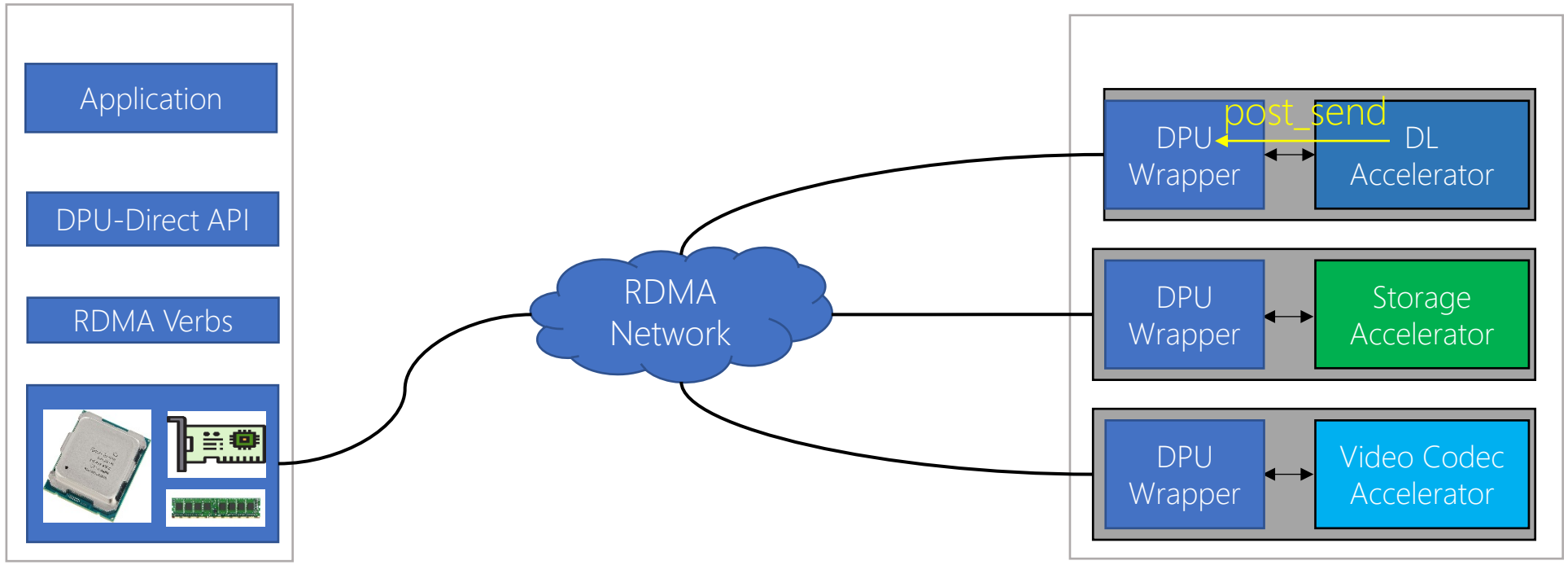
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



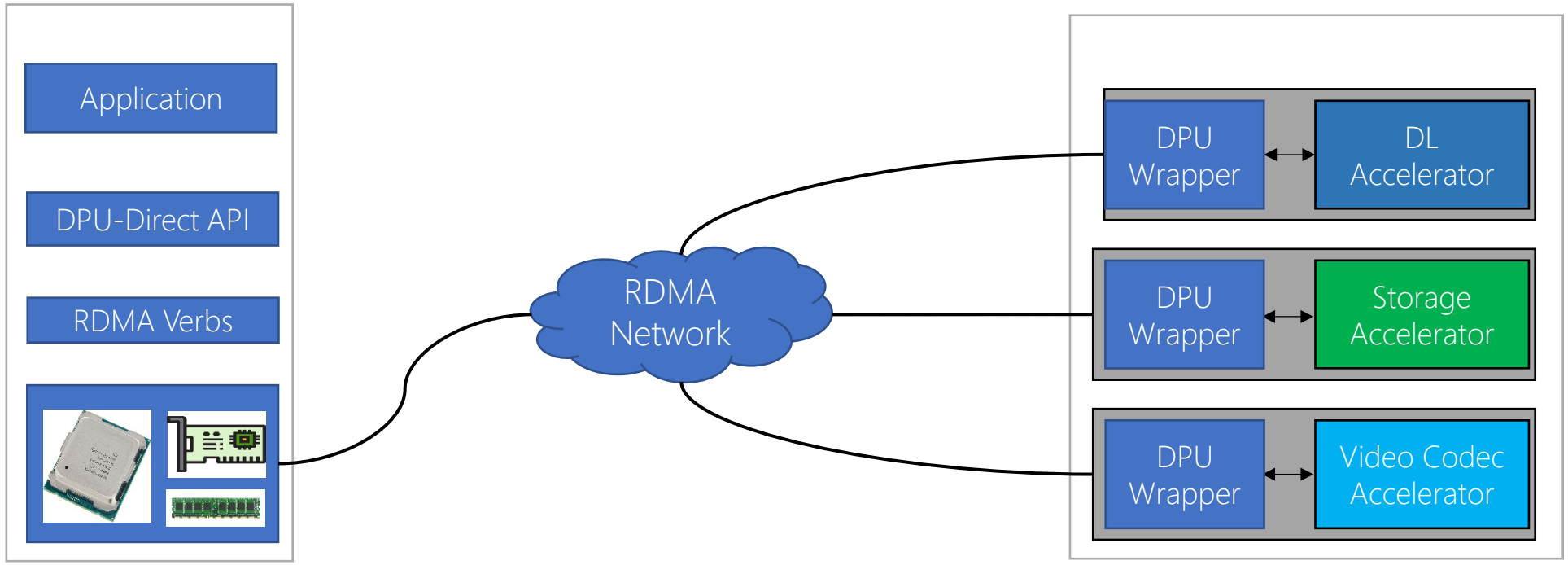
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



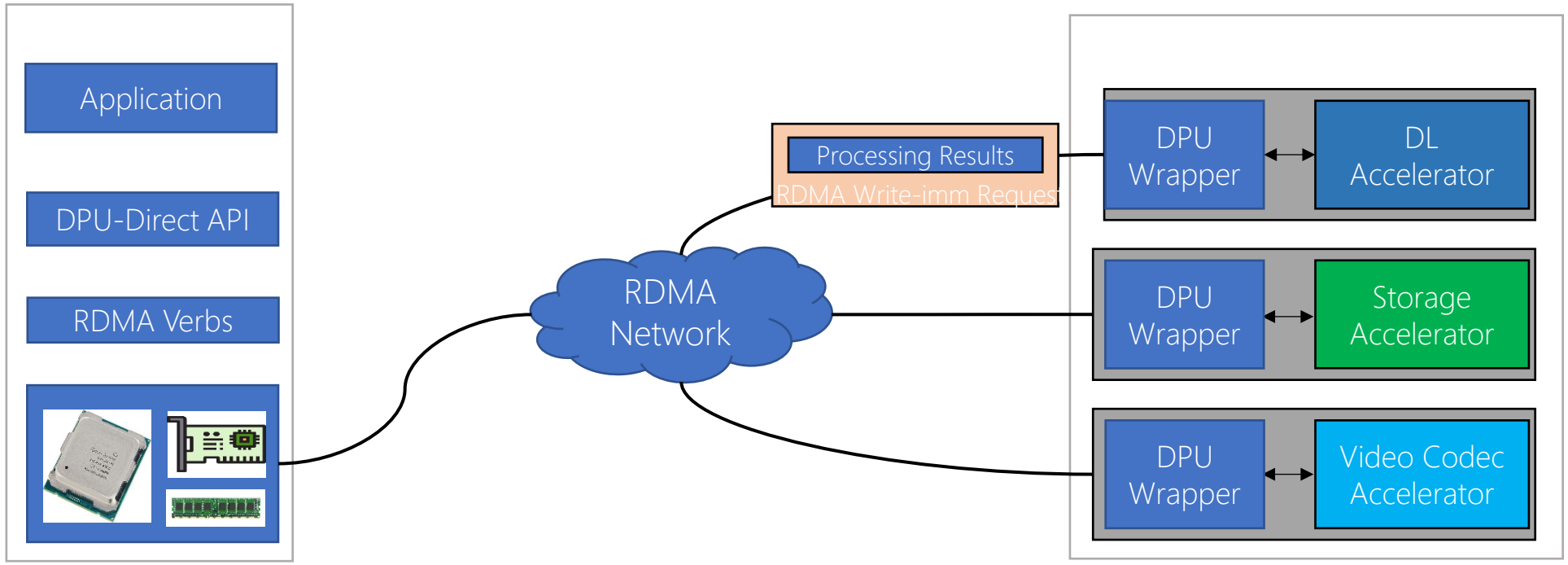
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



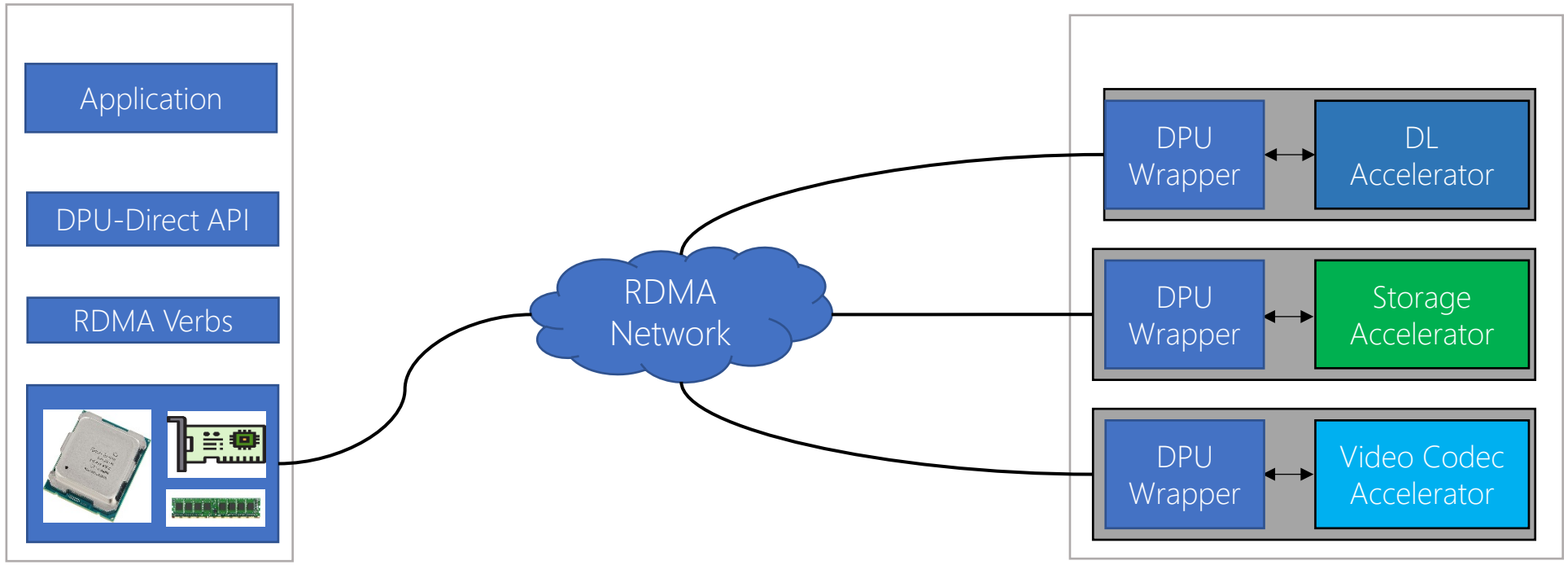
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



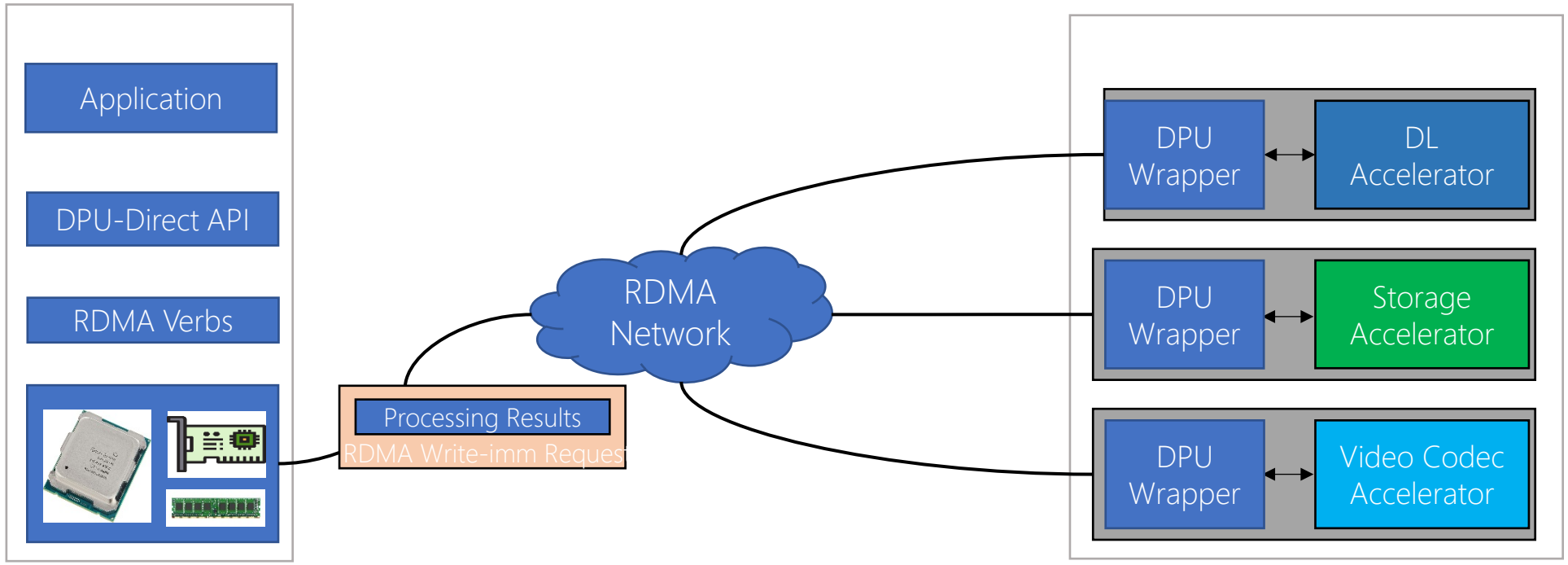
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



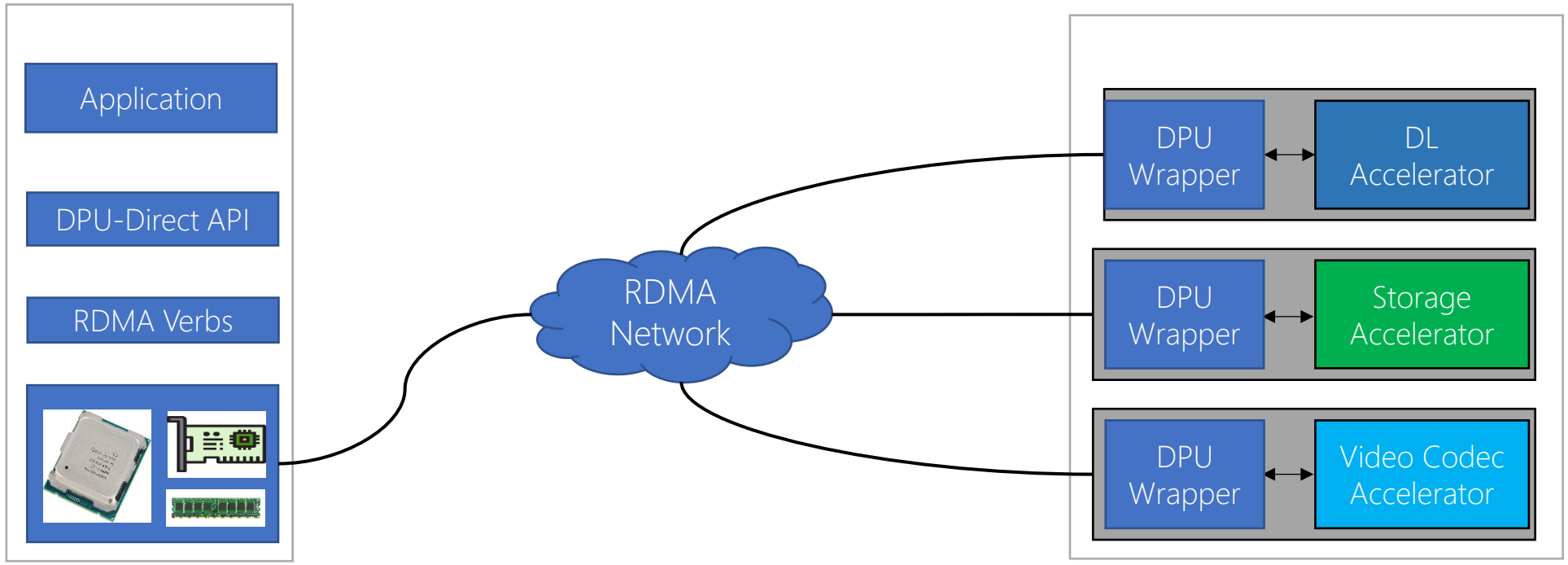
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



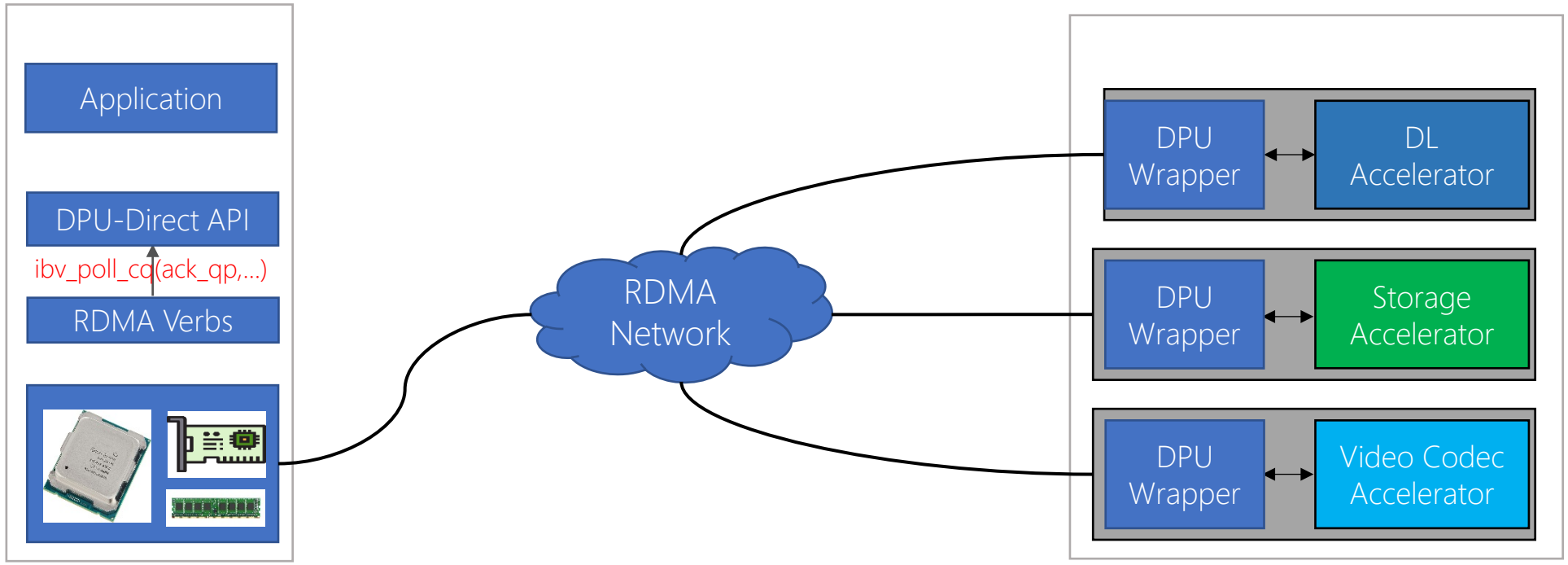
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



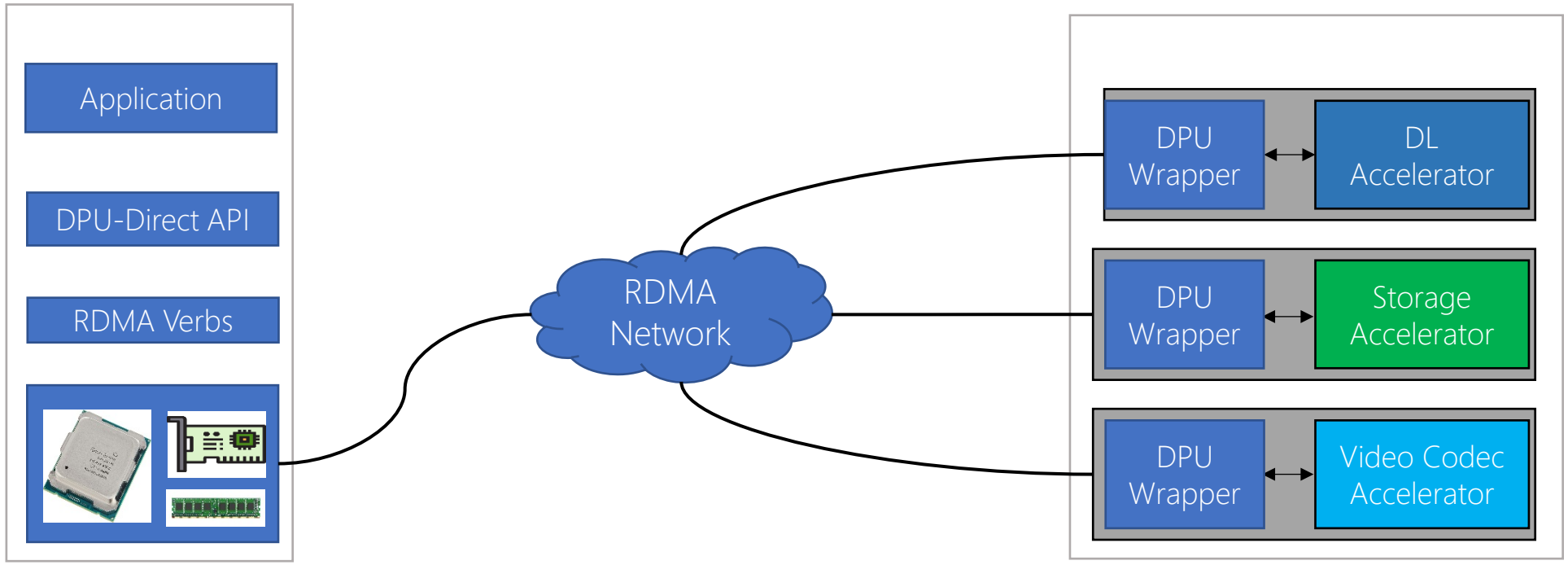
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



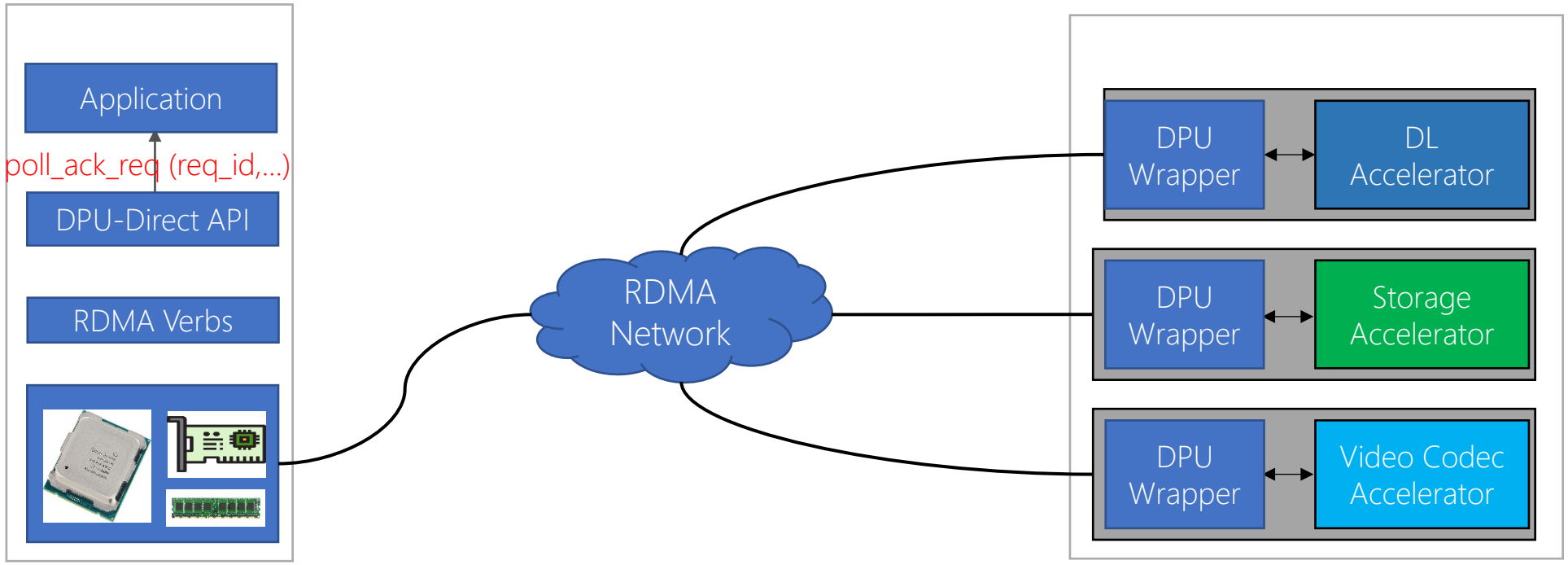
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



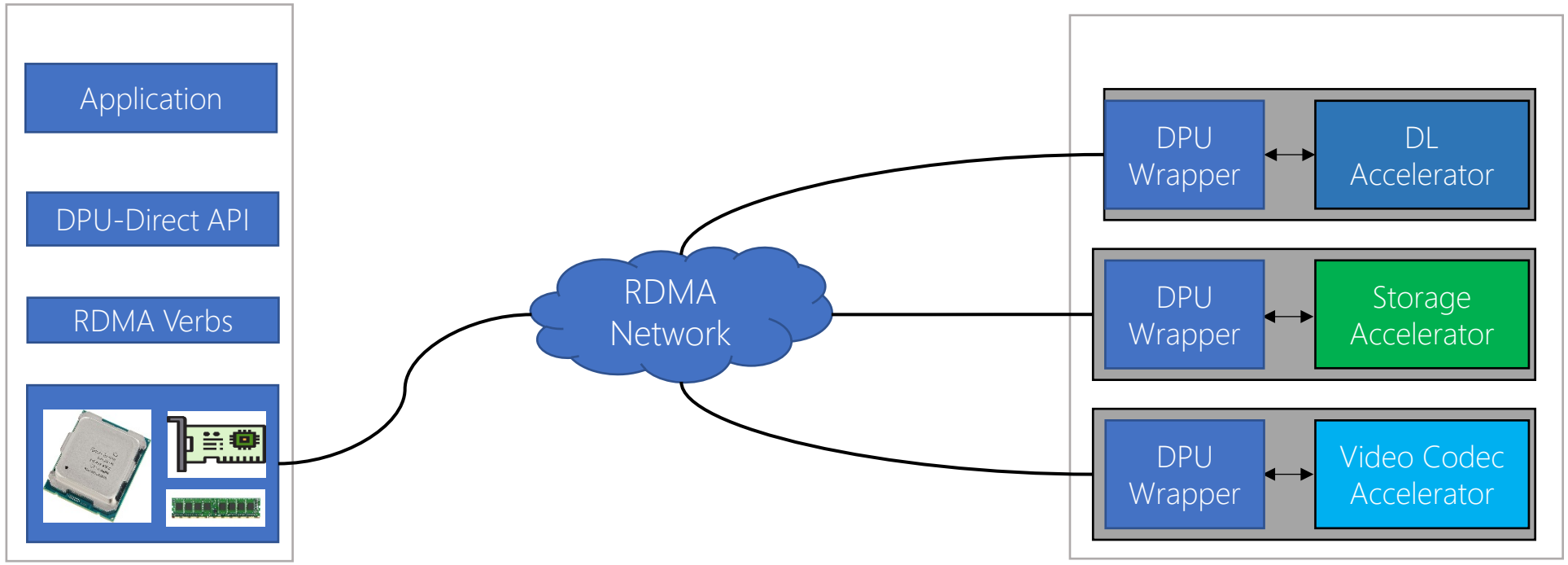
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



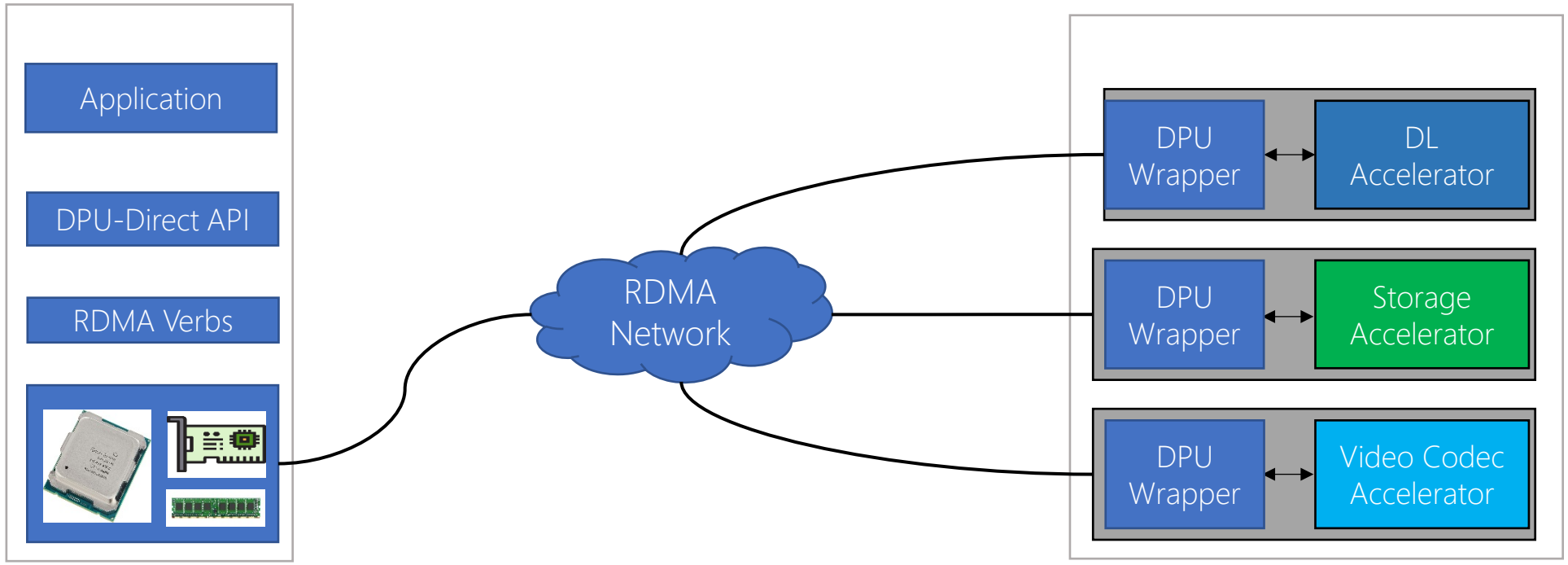
Overview

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



Overview

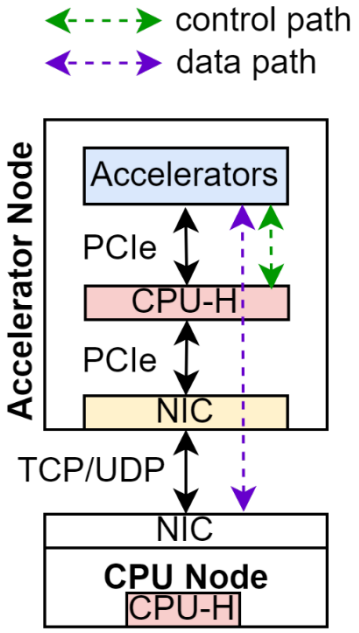
- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.



DPU-Direct connects accelerate nodes with CPU nodes via standard RDMA network.

Challenge 1: How to Architect an Accelerator Node

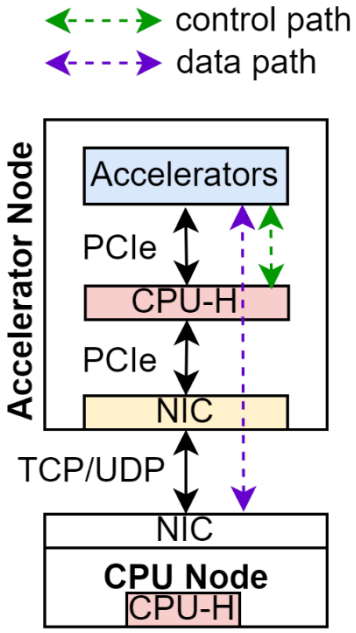
Challenge 1: How to Architect an Accelerator Node



NIC-Assisted

CPU-H: High-end CPUs

Challenge 1: How to Architect an Accelerator Node

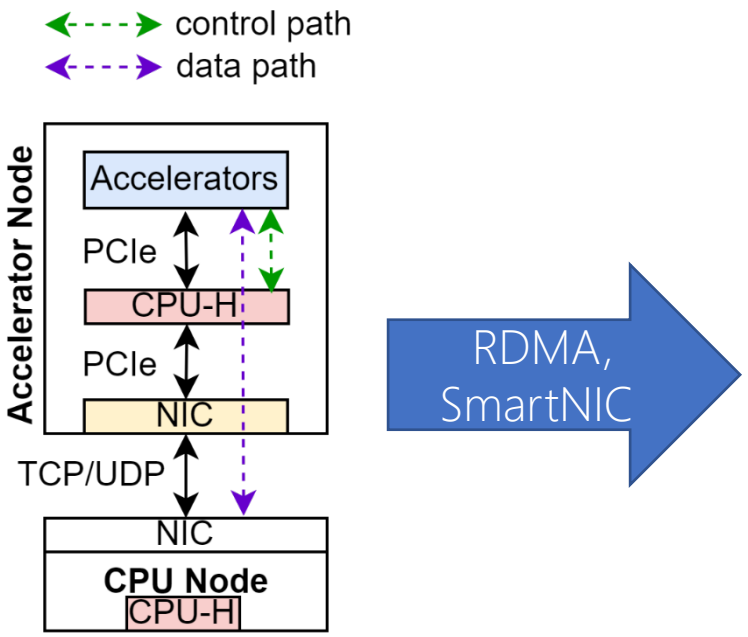


NIC-Assisted

- CPU bottlenecks the network data path

CPU-H: High-end CPUs

Challenge 1: How to Architect an Accelerator Node

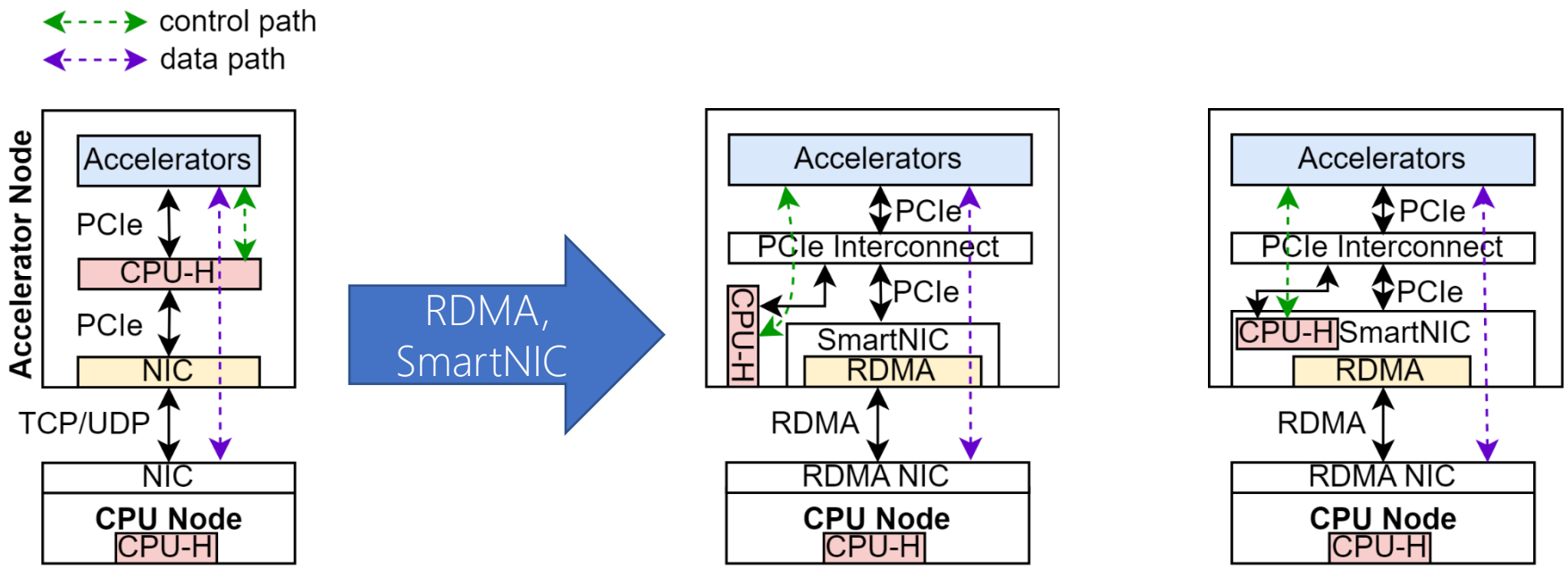


NIC-Assisted

- CPU bottlenecks the network data path

CPU-H: High-end CPUs

Challenge 1: How to Architect an Accelerator Node



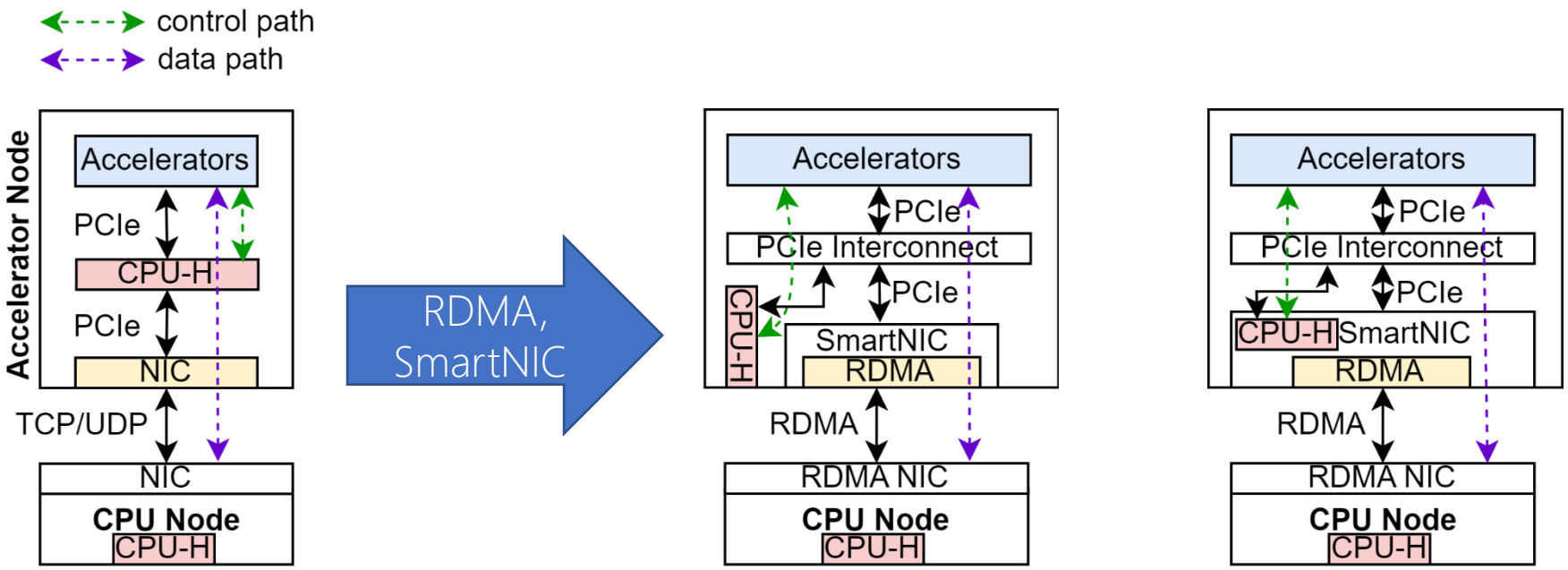
NIC-Assisted

SmartNIC-Assisted

- CPU bottlenecks the network data path

CPU-H: High-end CPUs

Challenge 1: How to Architect an Accelerator Node



NIC-Assisted

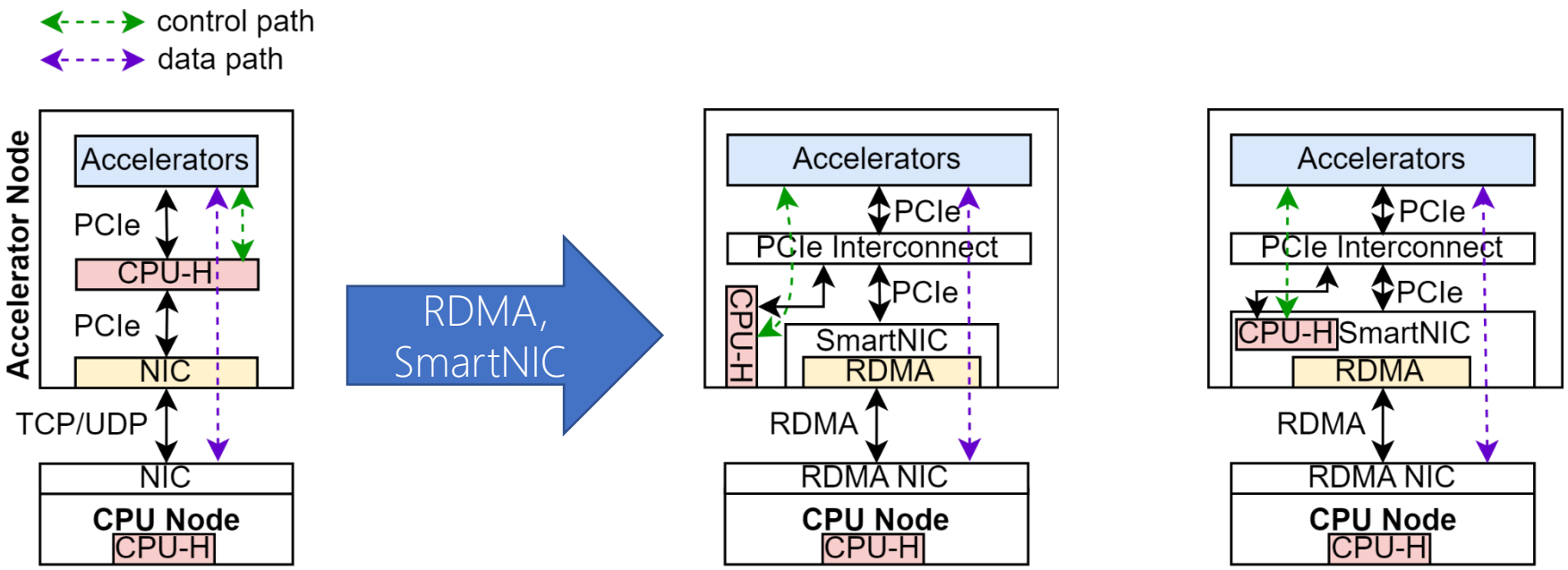
- CPU bottlenecks the network data path

SmartNIC-Assisted

- PCIe interface bottlenecks the network data path.

CPU-H: High-end CPUs

Challenge 1: How to Architect an Accelerator Node



NIC-Assisted

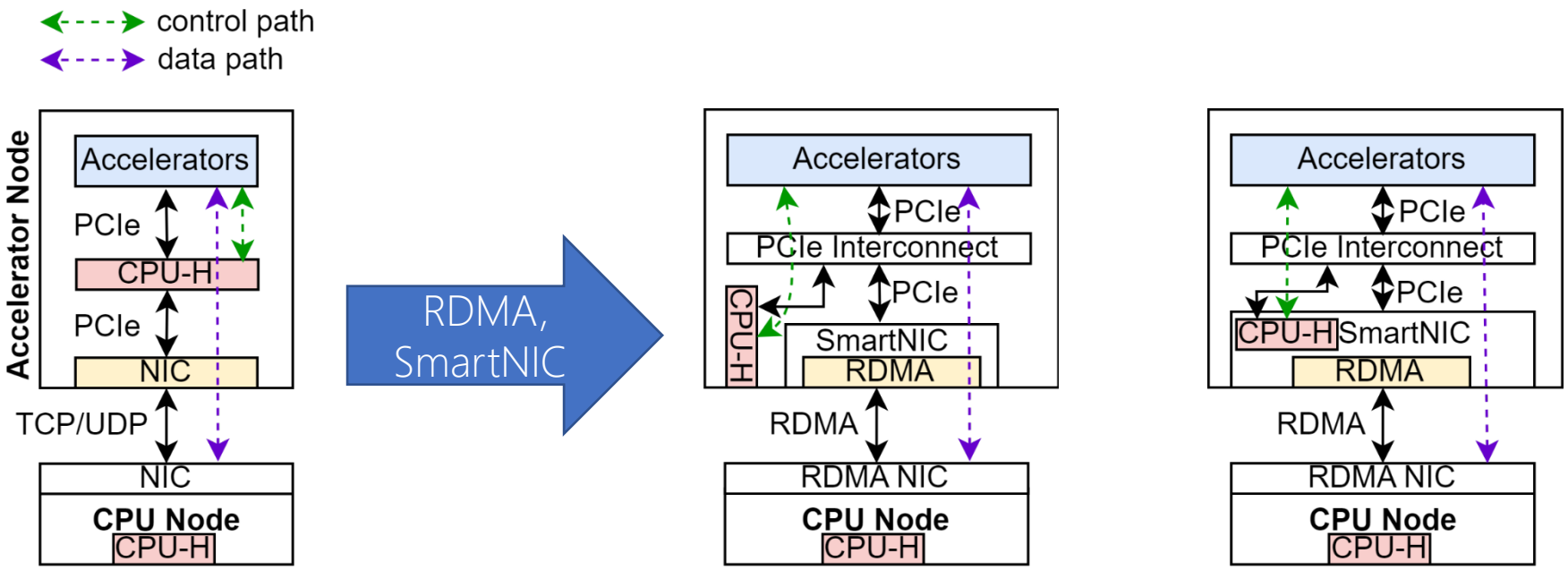
- CPU bottlenecks the network data path

SmartNIC-Assisted

- PCIe interface bottlenecks the network data path.
- CPU-H is overkill for the control path.

CPU-H: High-end CPUs

Challenge 1: How to Architect an Accelerator Node



NIC-Assisted

- CPU bottlenecks the network data path

CPU-H: High-end CPUs

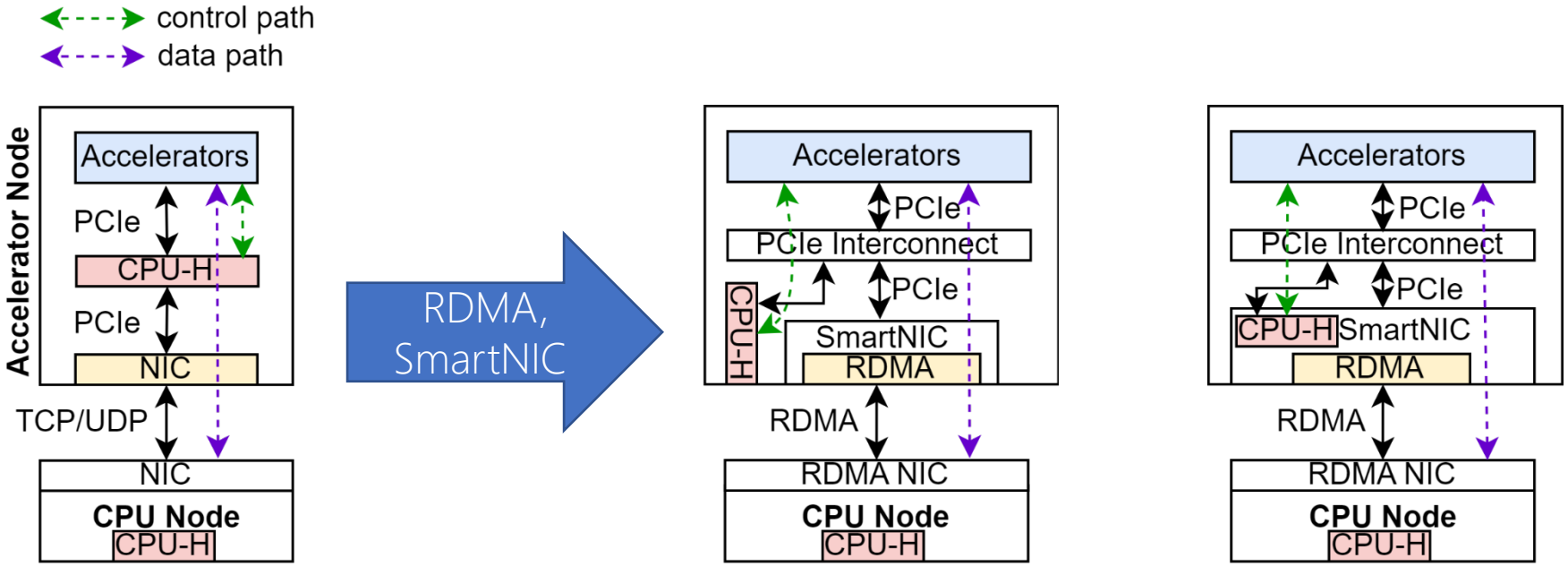
SmartNIC-Assisted

- PCIe interface bottlenecks the network data path.
- CPU-H is overkill for the control path.
- SmartNIC are over-design for accelerator disaggregation.

Challenge 1: How to Architect an Accelerator Node

Goal: The remote accelerator node should provide

compatible speedup & energy efficiency compared with local accelerator.



NIC-Assisted

- CPU bottlenecks the network data path

CPU-H: High-end CPUs

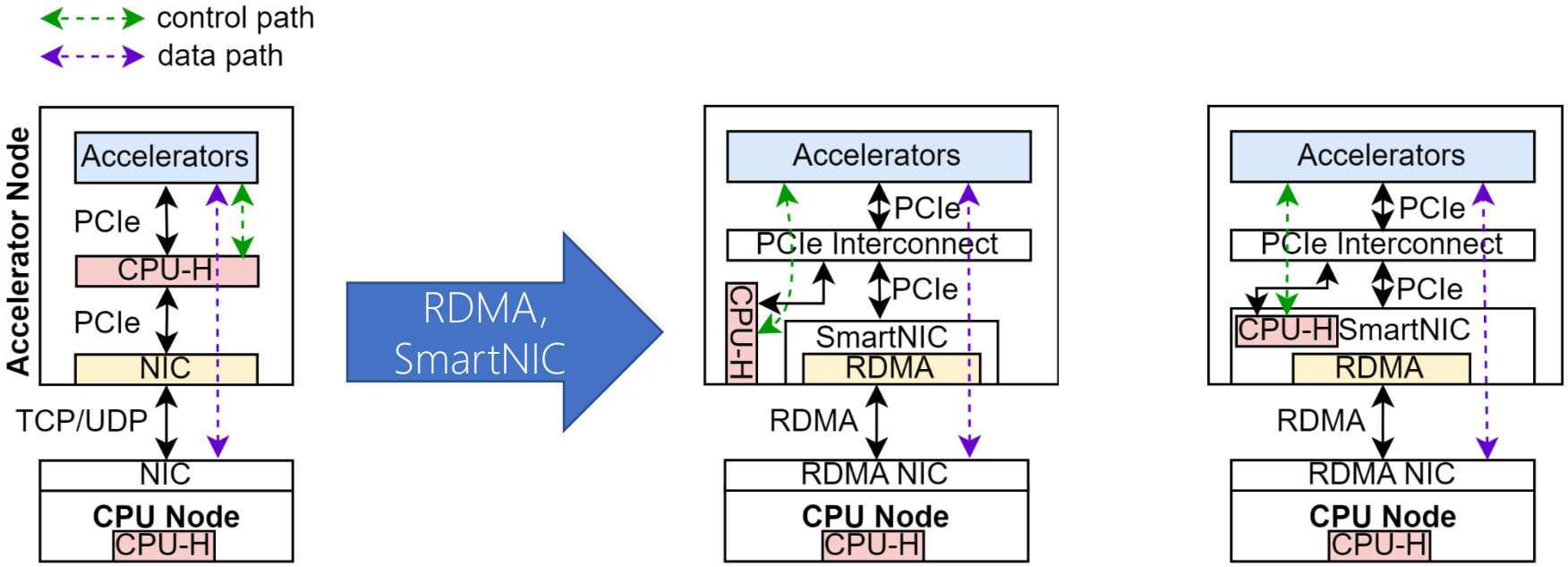
SmartNIC-Assisted

- PCIe interface bottlenecks the network data path.
- CPU-H is overkill for the control path.
- SmartNIC are over-design for accelerator disaggregation.

Challenge 1: How to Architect an Accelerator Node

Goal: The remote accelerator node should provide

compatible speedup & energy efficiency compared with local accelerator.



NIC-Assisted

- CPU bottlenecks the network data path

CPU-H: High-end CPUs

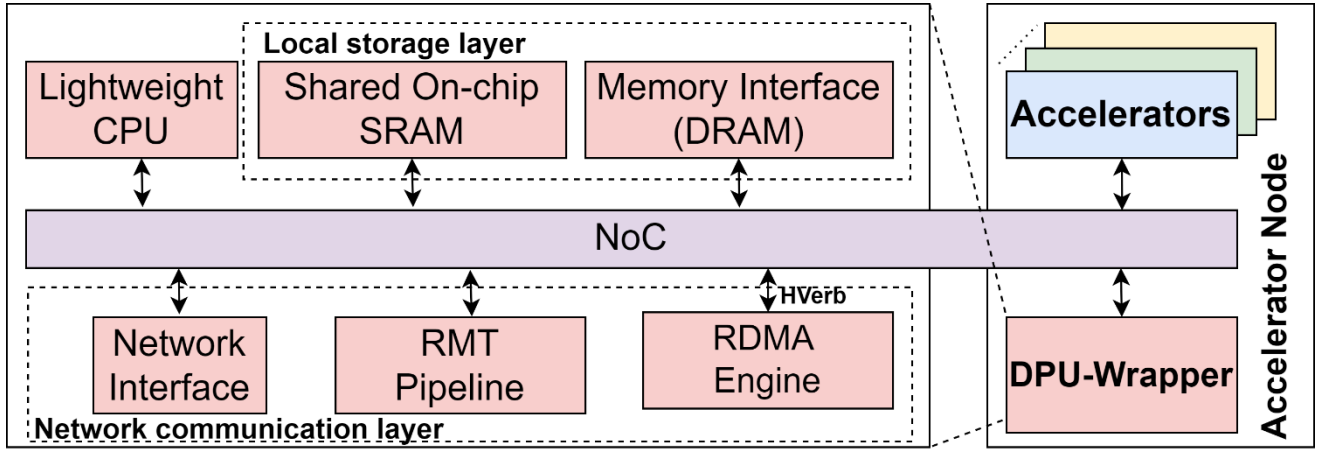
SmartNIC-Assisted

- PCIe interface bottlenecks the network data path.
- CPU-H is overkill for the control path.
- SmartNIC are over-design for accelerator disaggregation.

Accelerators node should be redesigned for DDC.

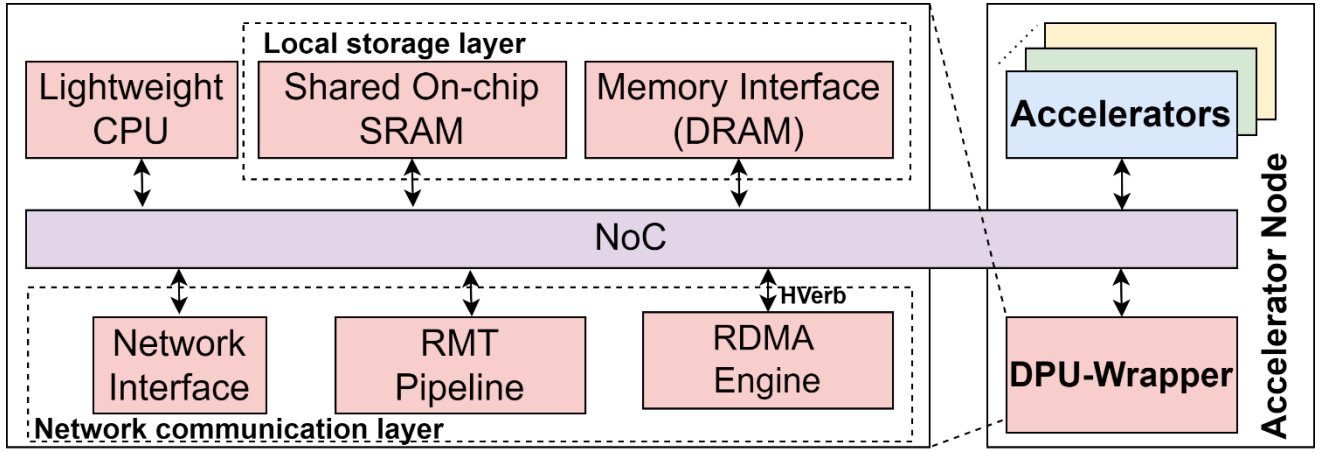
DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators



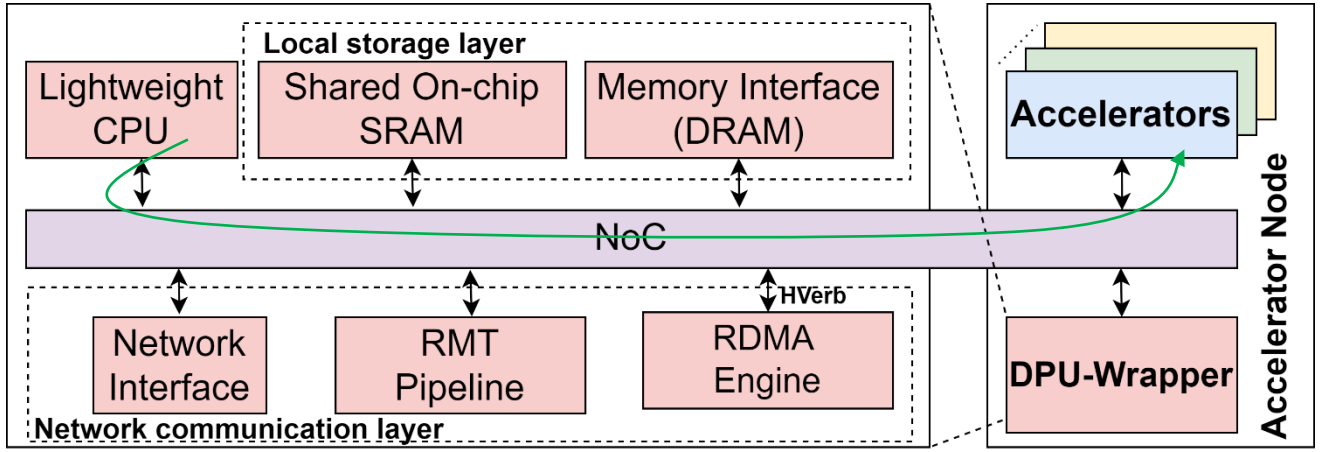
DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation



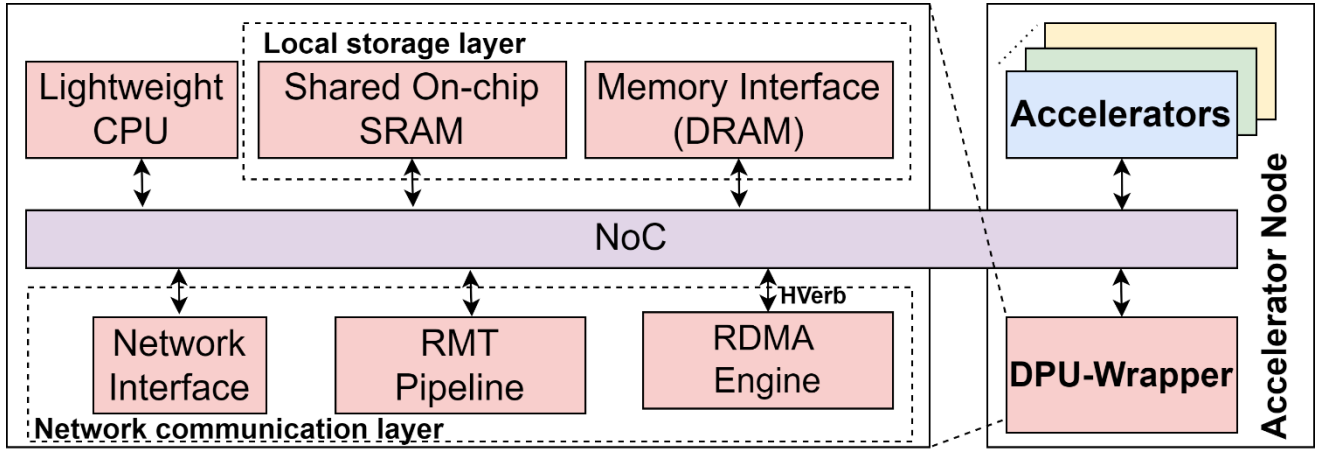
DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation



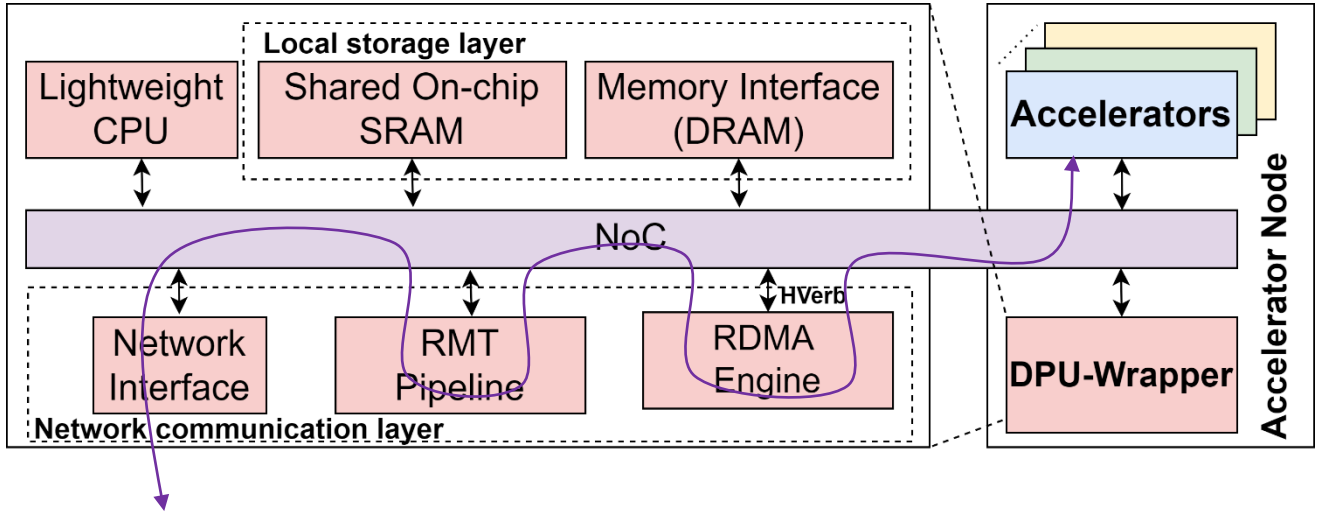
DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation



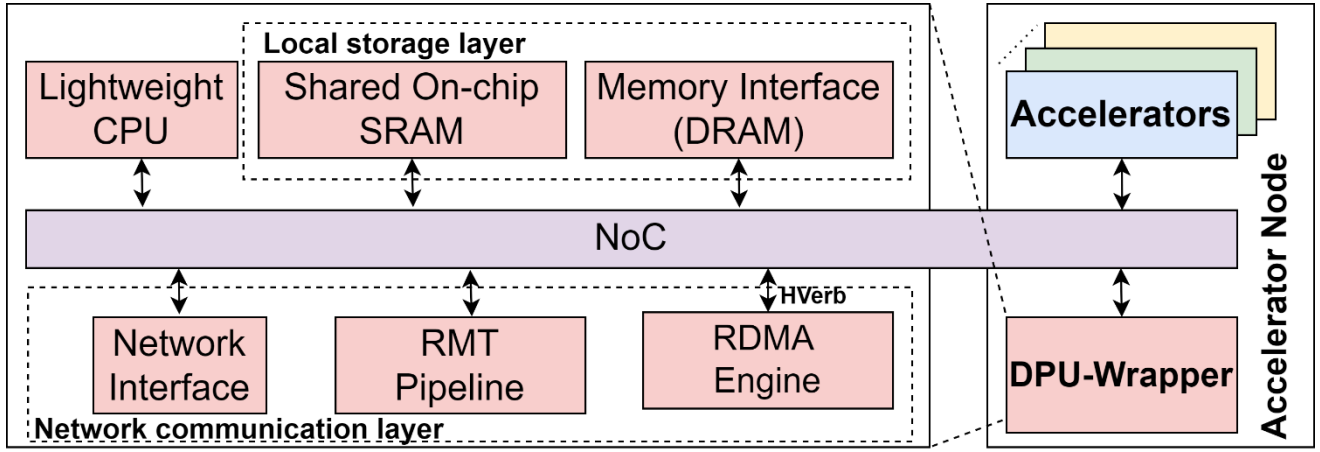
DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation



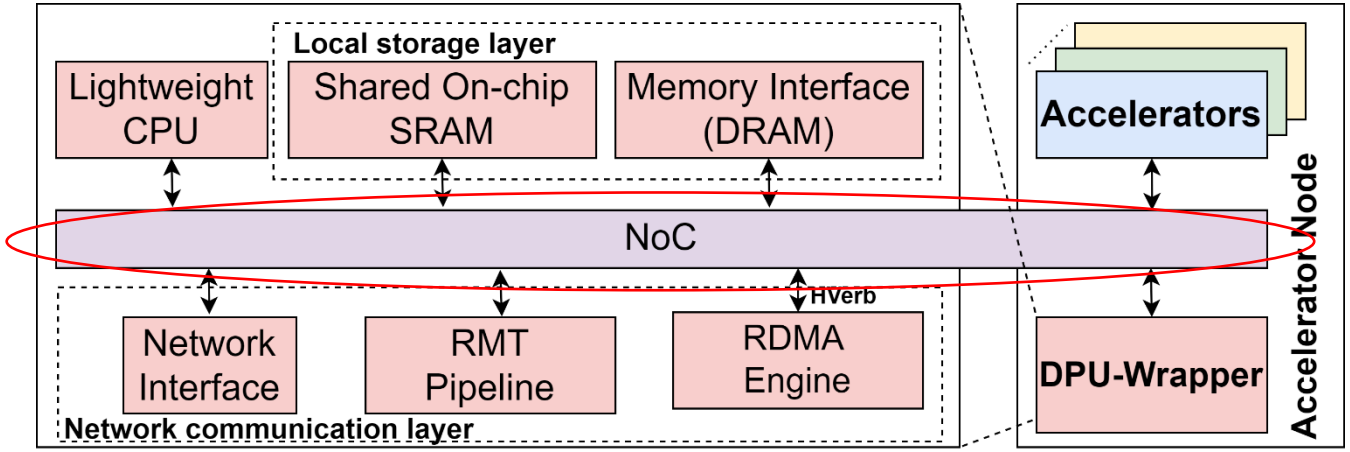
DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation



DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.

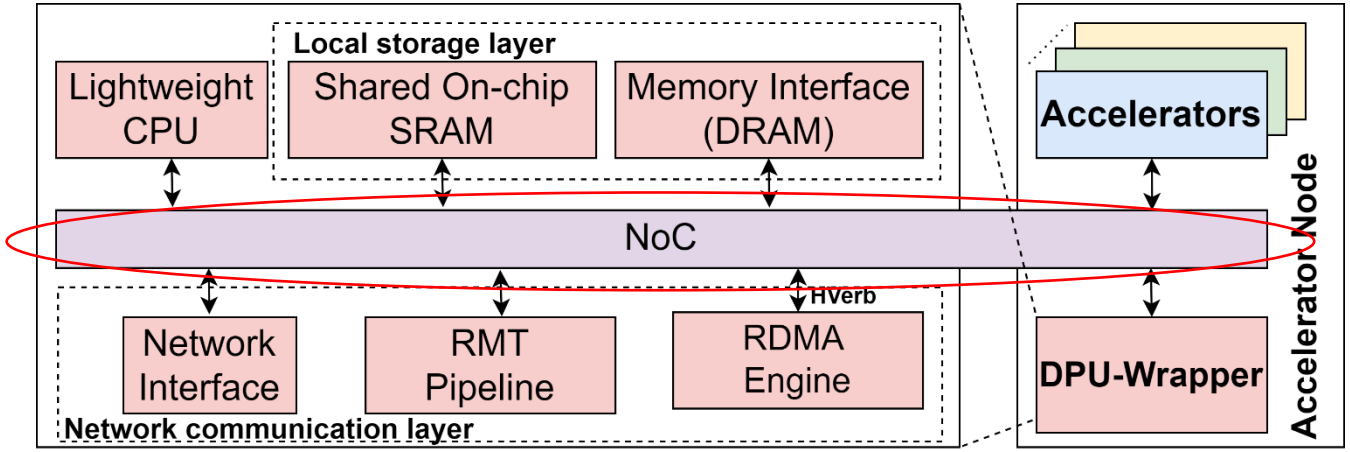


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

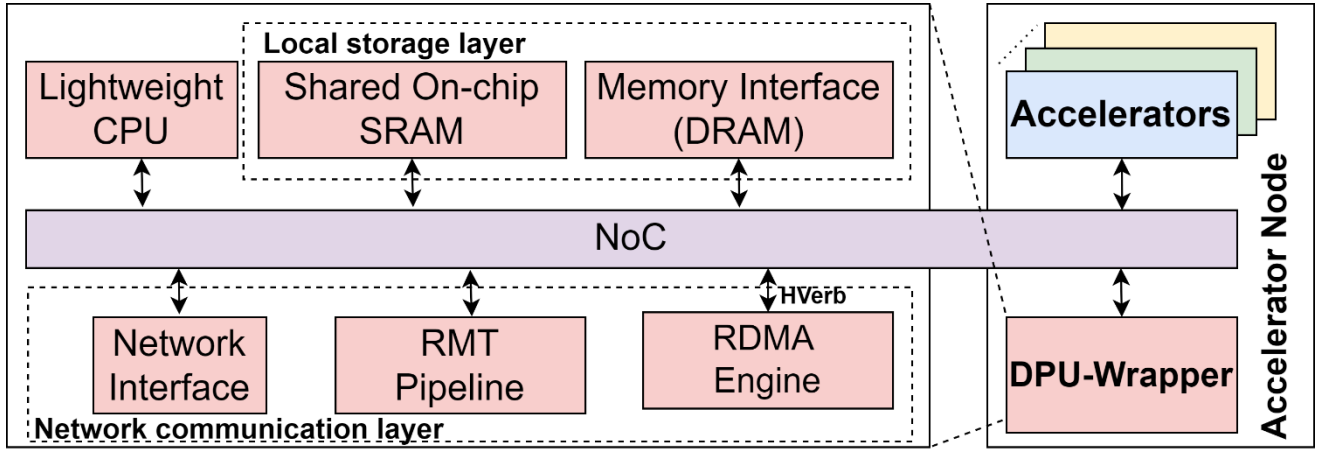


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

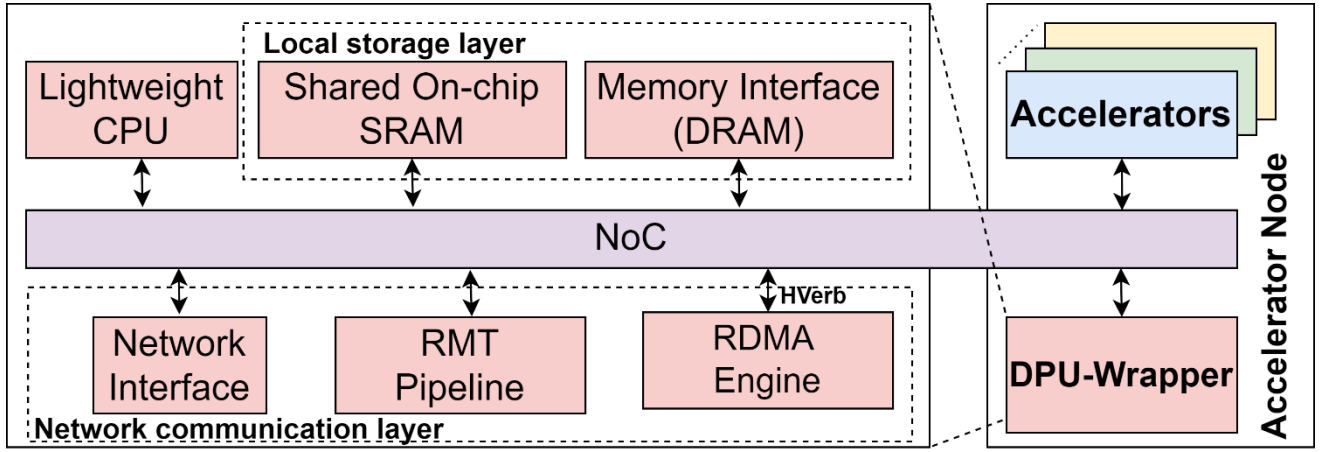


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

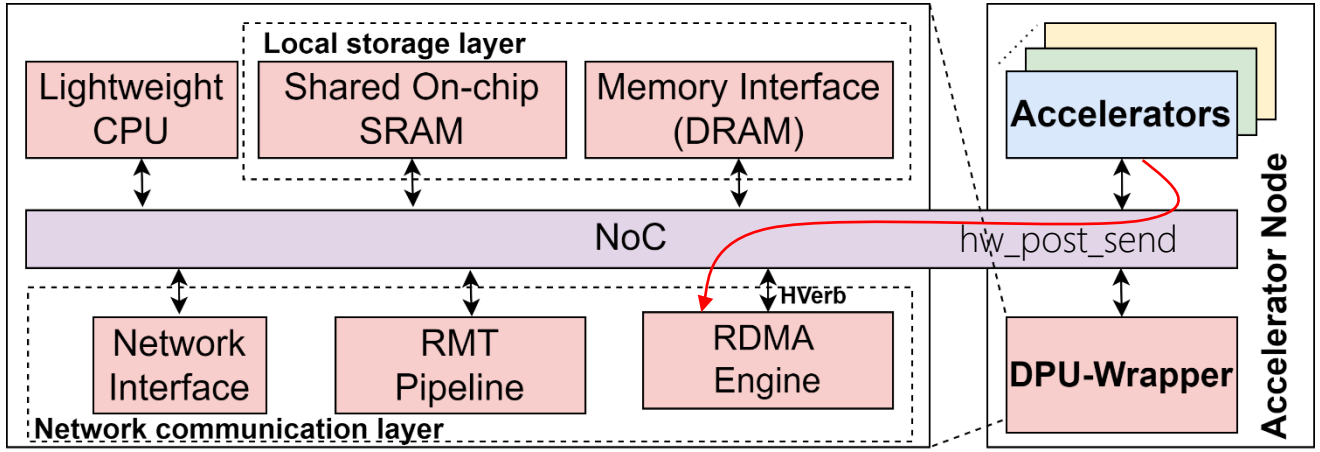


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

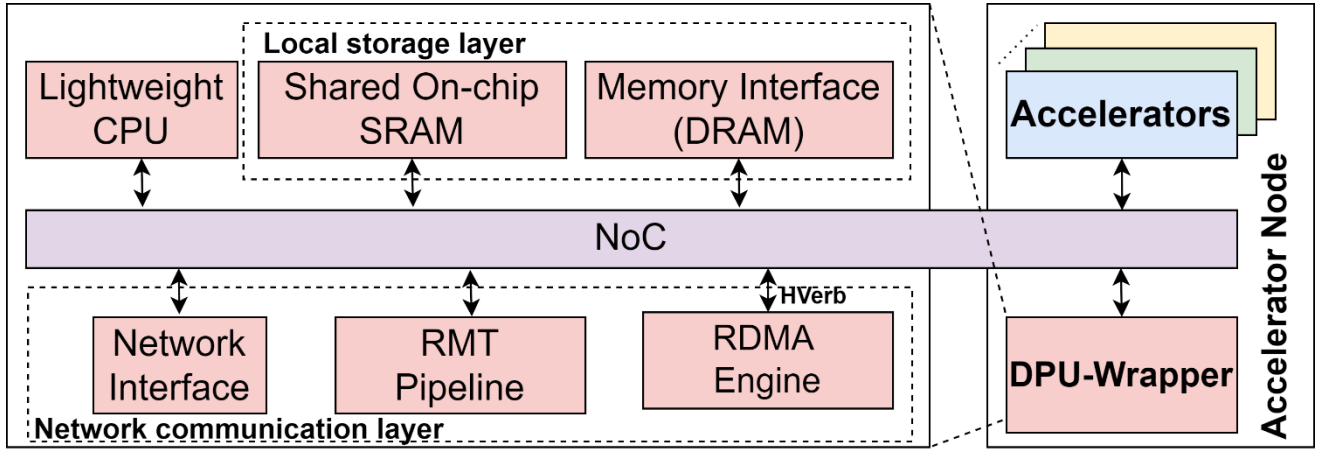


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

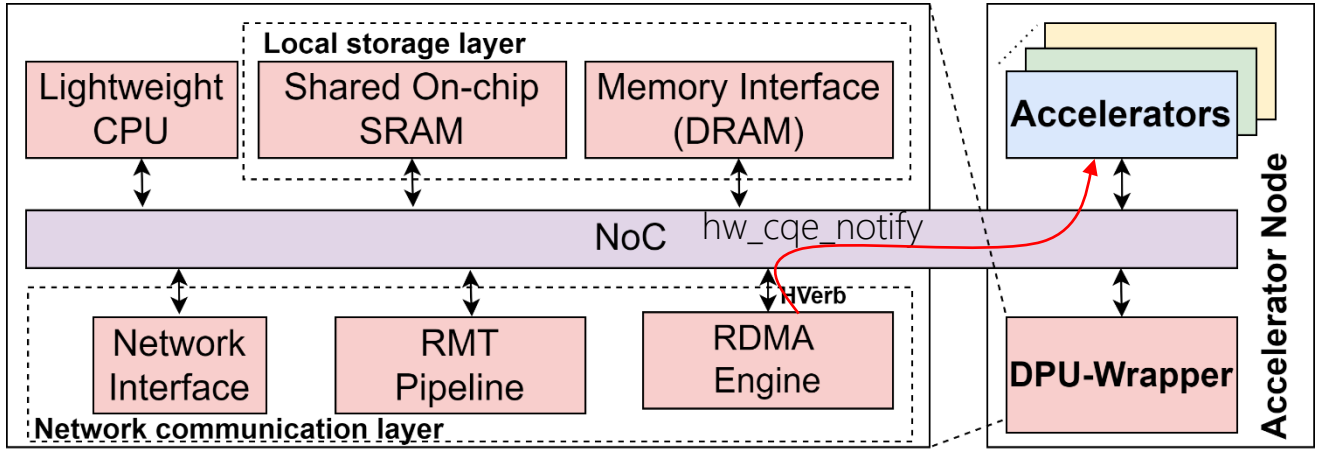


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

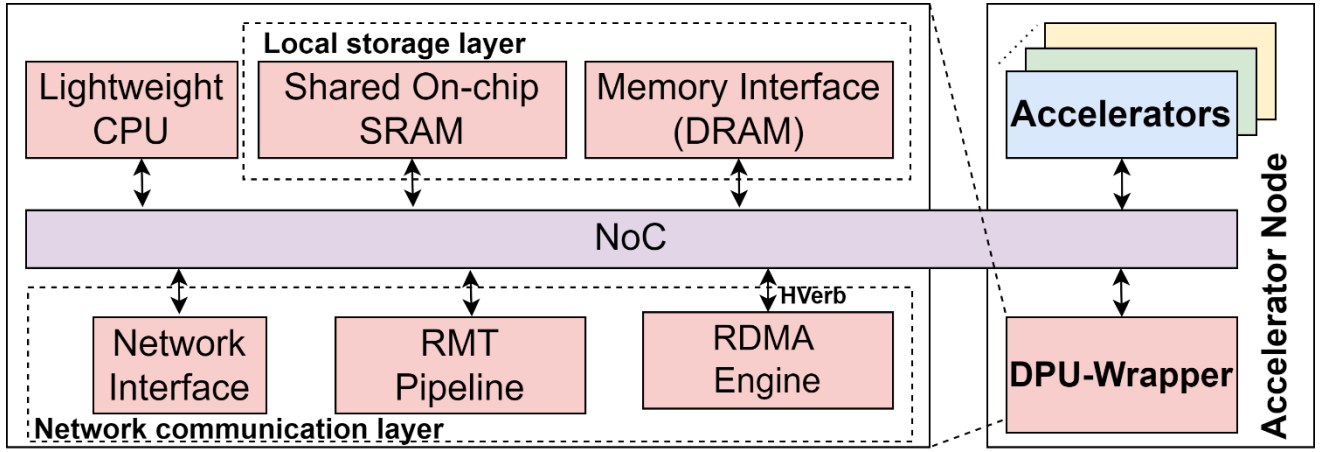


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

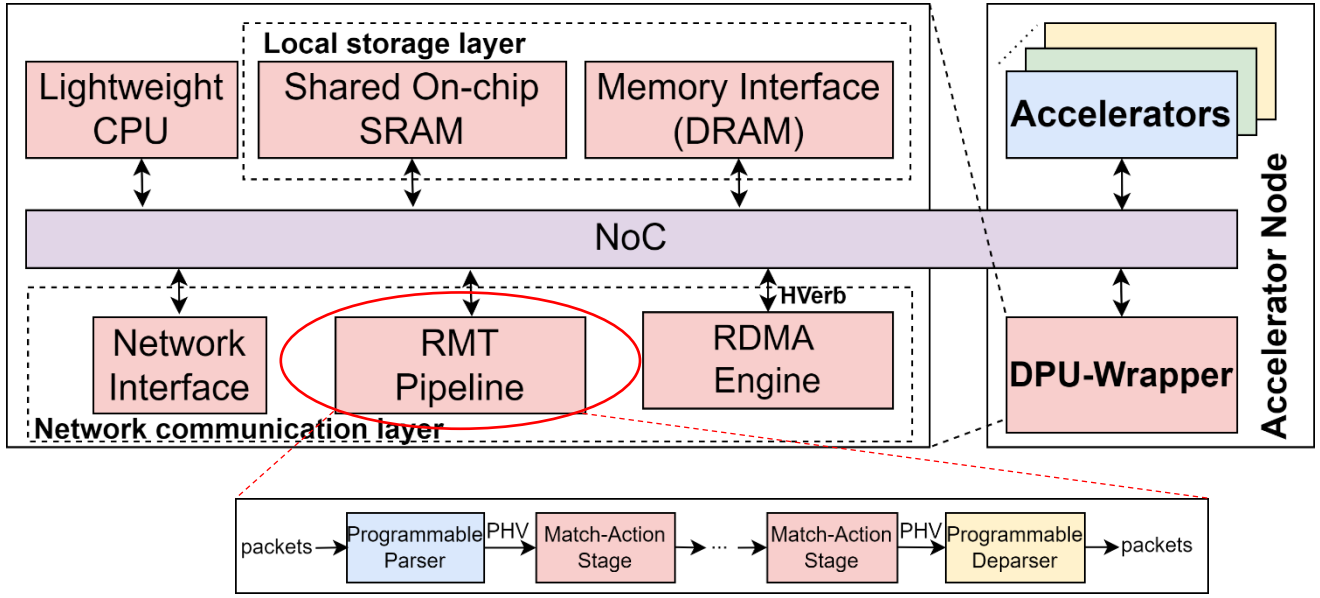


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).
  - RMT pipeline for layer-2 and layer-3 network functions, like VXLAN and NAT.

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

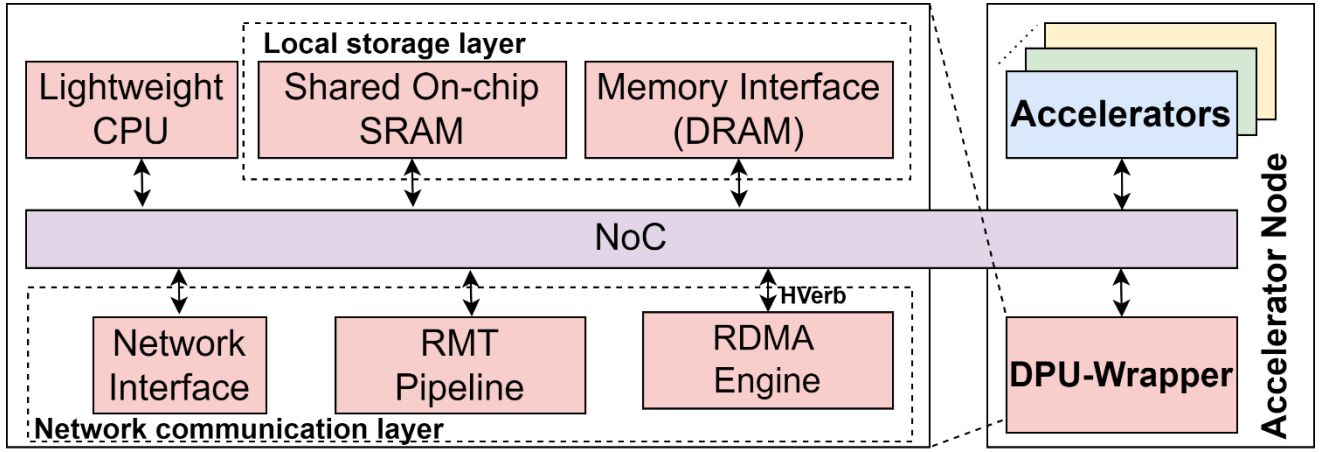


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).
  - RMT pipeline for layer-2 and layer-3 network functions, like VXLAN and NAT.

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

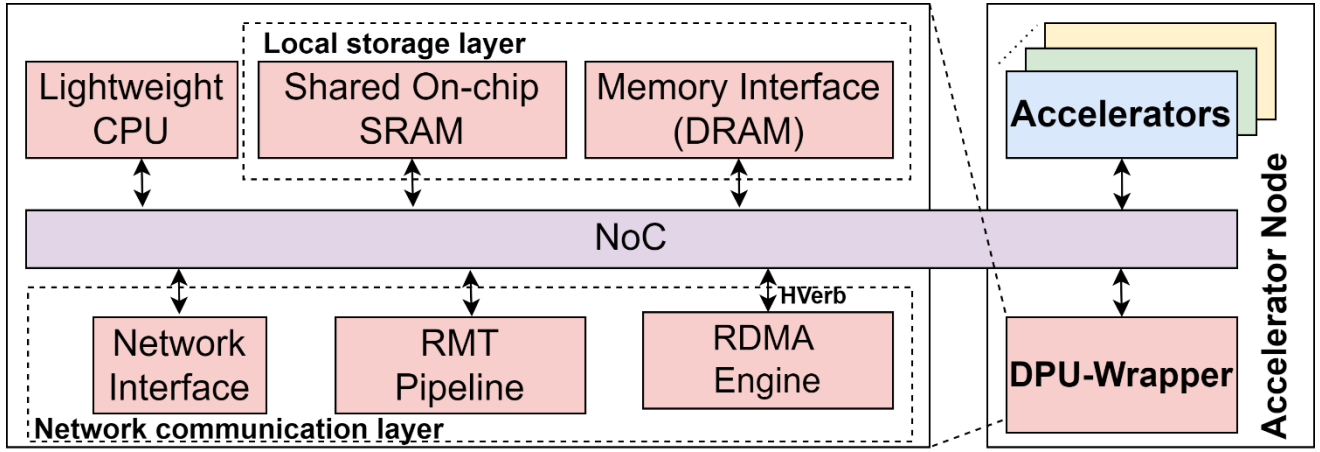


DPU Wrapper

- Accelerator Node = DPU Wrapper + Accelerators
  - Control-Data separation
  - Low latency: Accelerators is connected with the RDMA core by the on-chip network.
  - Accelerator interface with the accelerators via message passing (Hardware Verb Interface).
  - RMT pipeline for layer-2 and layer-3 network functions, like VXLAN and NAT.

	Interface Latency	Bandwidth
PCIe-based	100s ns	100s GB/s
Chiplet-based	10s ns	1s TB/s
NoC-based	1s ns	10s TB/s

Source: MCM-GPU[5]

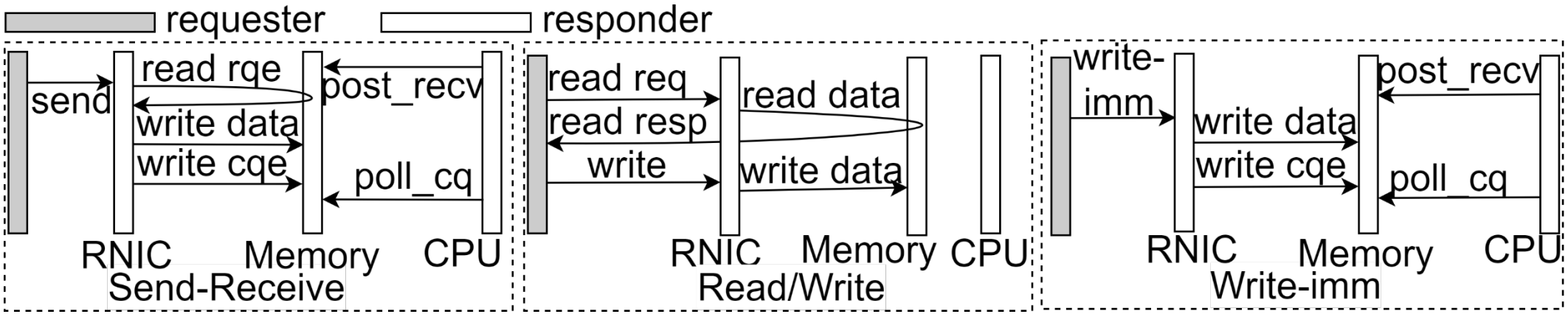


DPU-Direct draws clean-state design by treating accelerator as the first-class citizens.

Challenge 2: How to Architect the RDMA pattern

Challenge 2: How to Architect the RDMA pattern

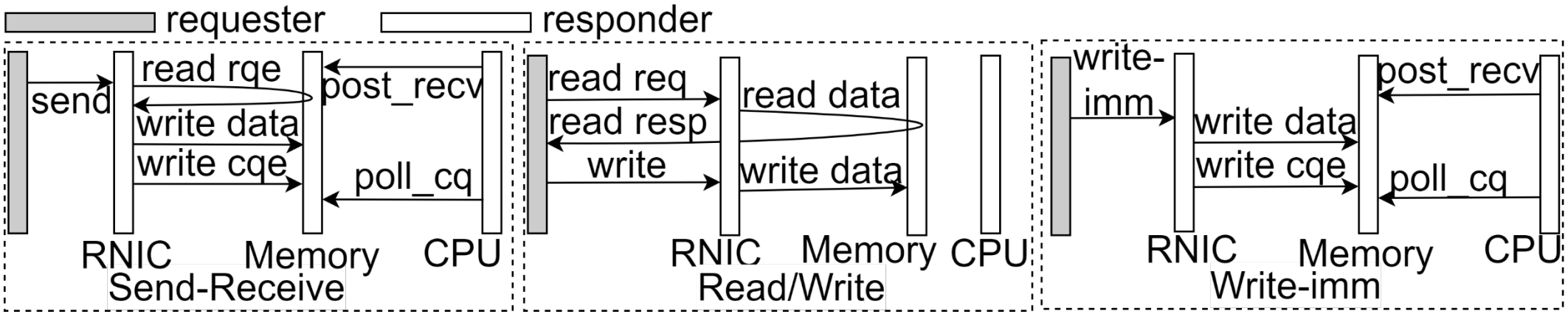
- RDMA lacks semantics for accelerator disaggregation.



The interaction process between requester and responder of the RDMA operations.

Challenge 2: How to Architect the RDMA pattern

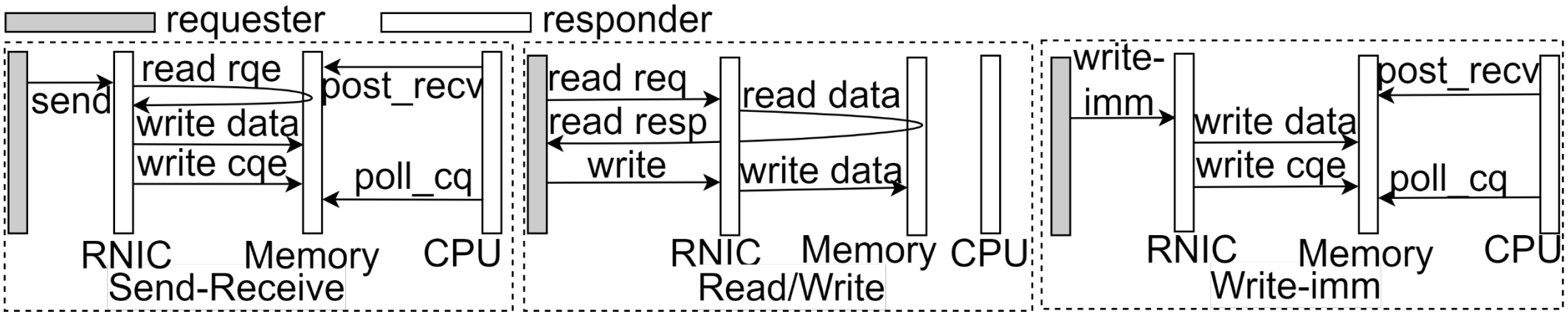
- RDMA lacks semantics for accelerator disaggregation.
- Existing method (StRoM[6]) introduce new RDMA operations, failing to interoperate with the commodity RNICs.



The interaction process between requester and responder of the RDMA operations.

Challenge 2: How to Architect the RDMA pattern

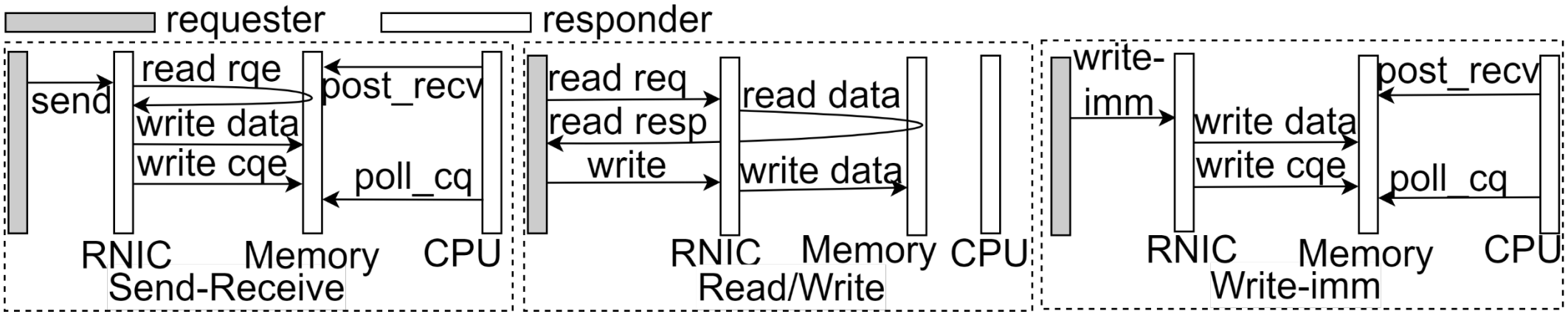
- RDMA lacks semantics for accelerator disaggregation.
- Existing method (StRoM[6]) introduce new RDMA operations, failing to interoperate with the commodity RNICs.
  - The commodity RNICs cannot generate RoCEv2 packets with the five newly added BTH opcodes for StRoM kernel interaction.



The interaction process between requester and responder of the RDMA operations.

Challenge 2: How to Architect the RDMA pattern

- RDMA lacks semantics for accelerator disaggregation.
- Existing method (StRoM[6]) introduce new RDMA operations, failing to interoperate with the commodity RNICs.
  - The commodity RNICs cannot generate RoCEv2 packets with the five newly added BTH opcodes for StRoM kernel interaction.

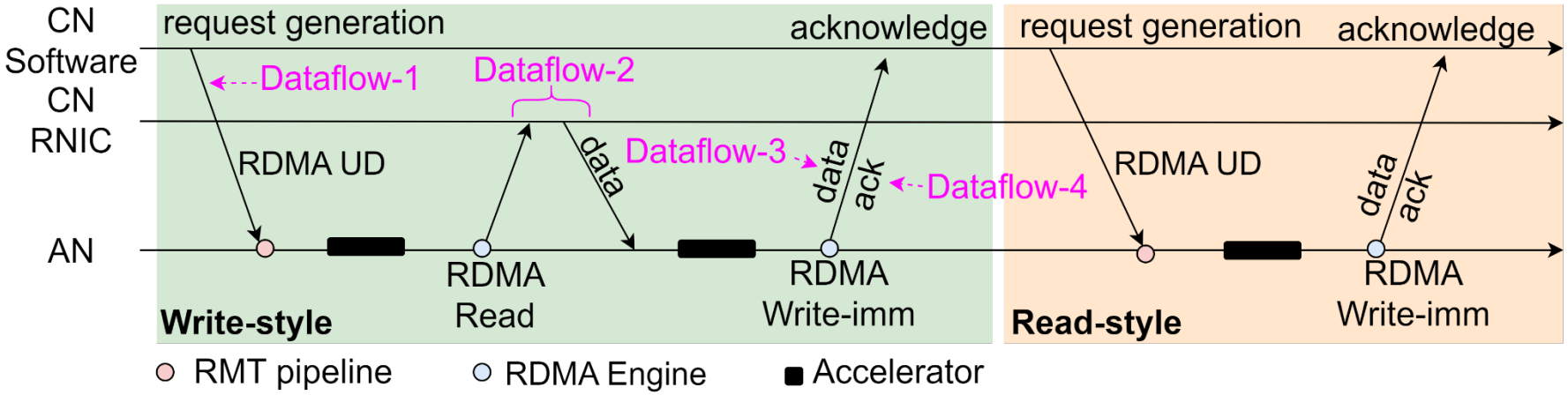


The interaction process between requester and responder of the RDMA operations.

The RDMA communication pattern should be redesigned for accelerator disaggregation?

RAAP

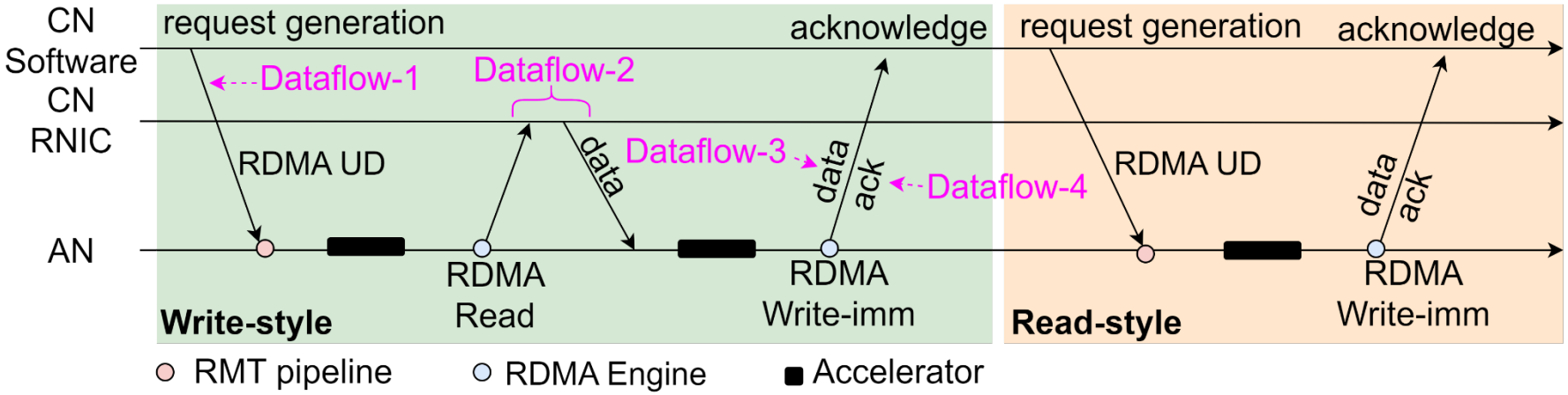
- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.



RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-1: transfer the request from CN to accelerator

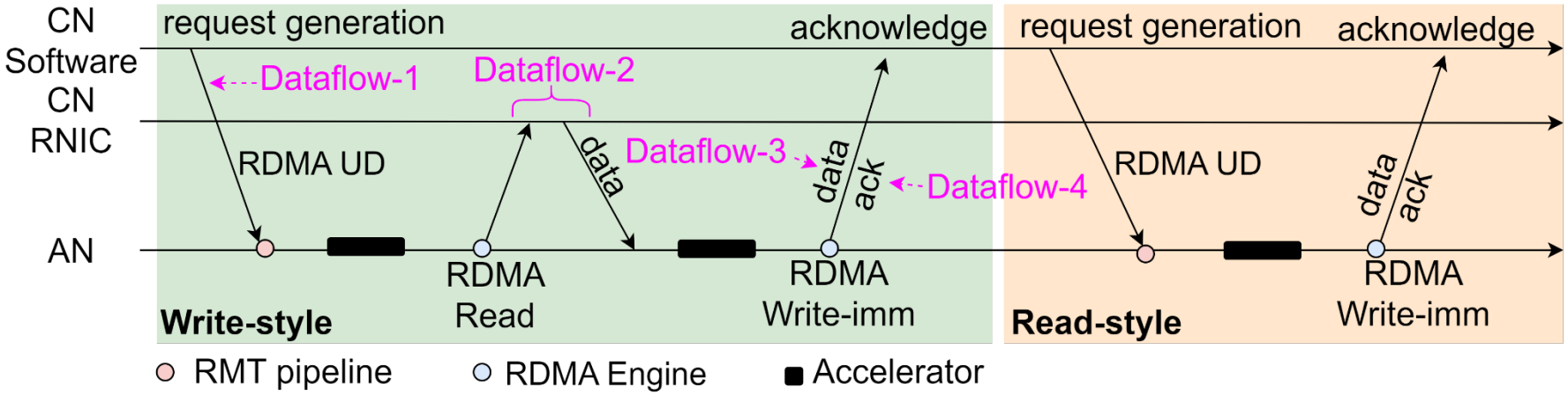


RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-1: transfer the request from CN to accelerator

RAAP: embed accelerator request in an RDMA unreliable datagram (UD) packet.



RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

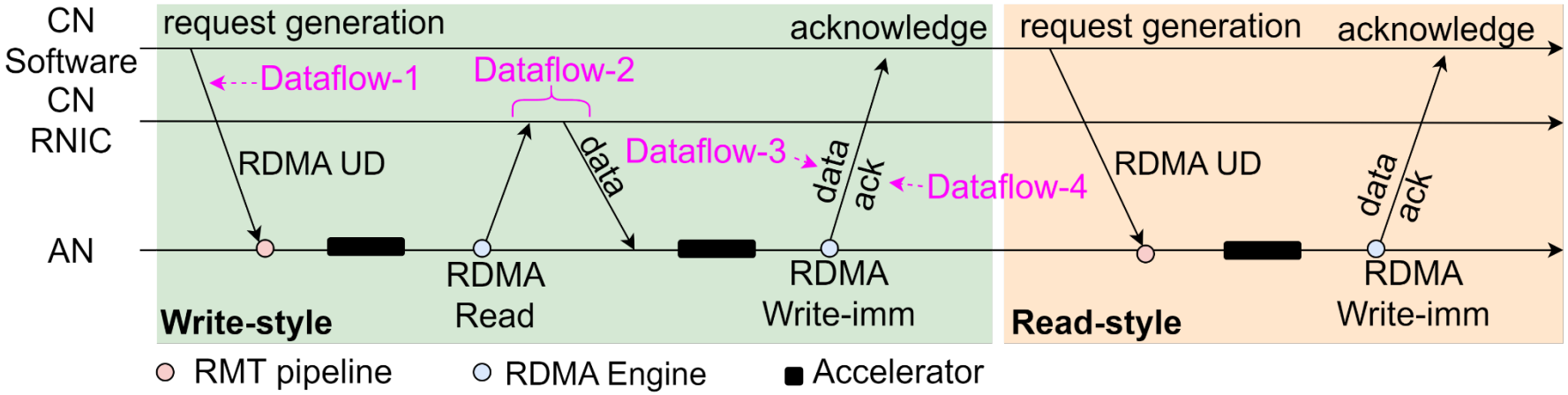
Dataflow-1: transfer the request from CN to accelerator

RAAP: embed accelerator request in an RDMA unreliable datagram (UD) packet.

Low-overhead request forwarding

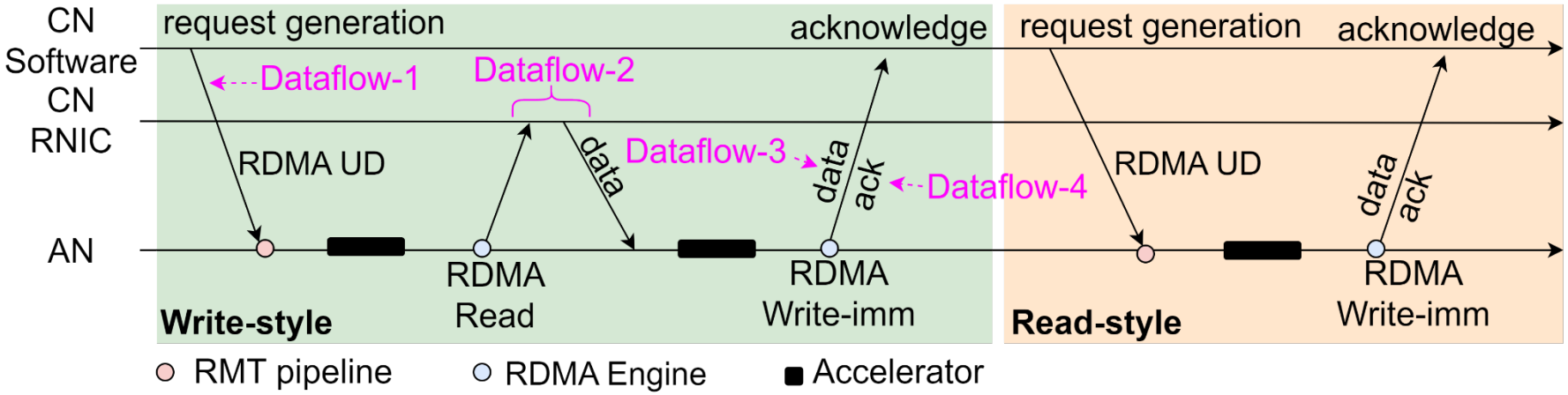
Enable out-of-order processing

Easy to scale the request generation



RAAP

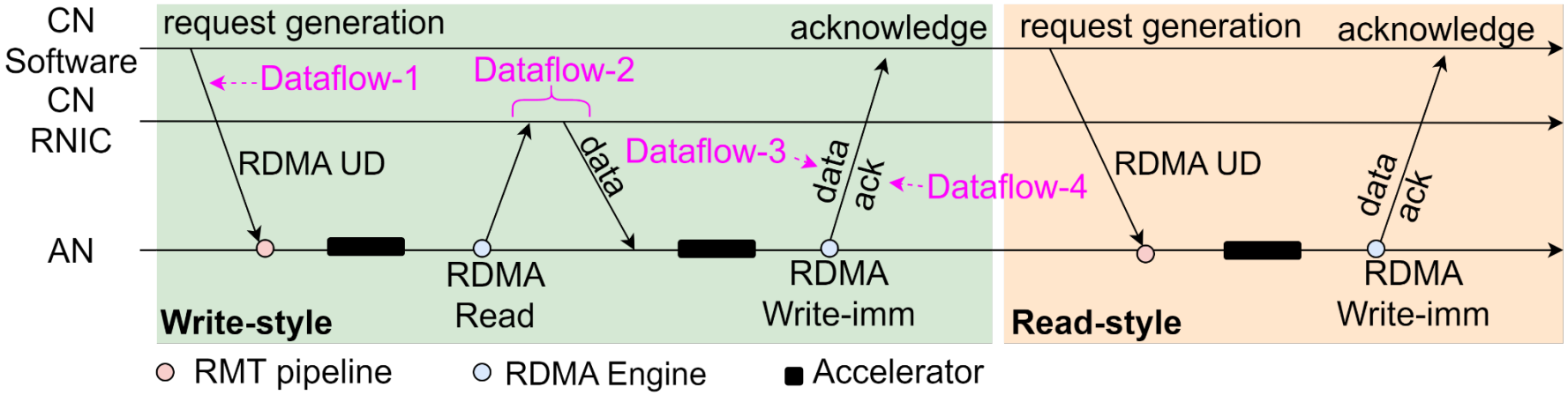
- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.



RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-2: transfer the bulk data from the CN to accelerator

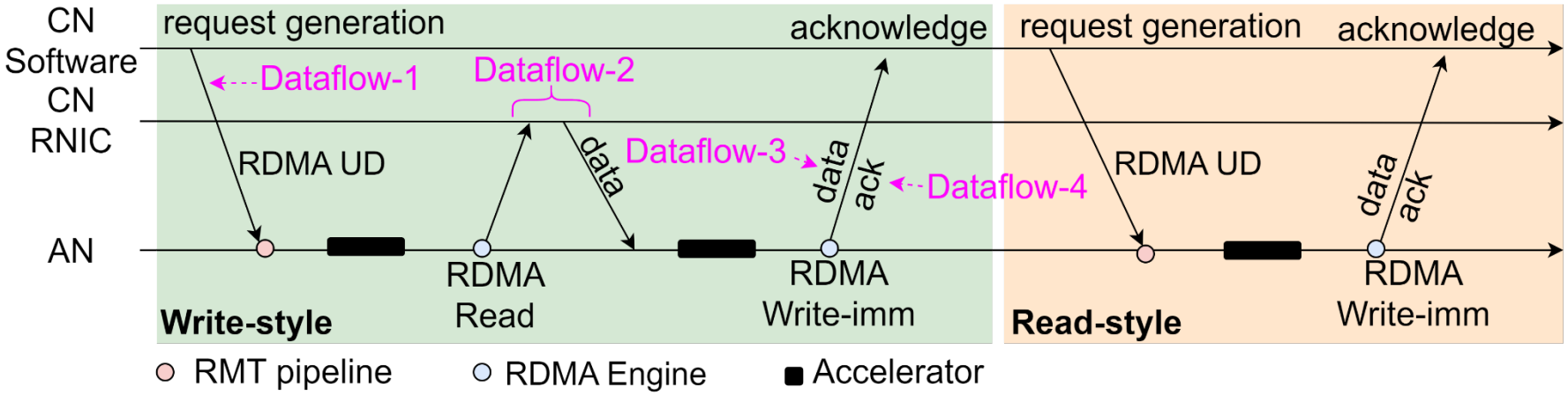


RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-2: transfer the bulk data from the CN to accelerator

RAAP: the accelerator proactively fetch the data to bypass the CN.



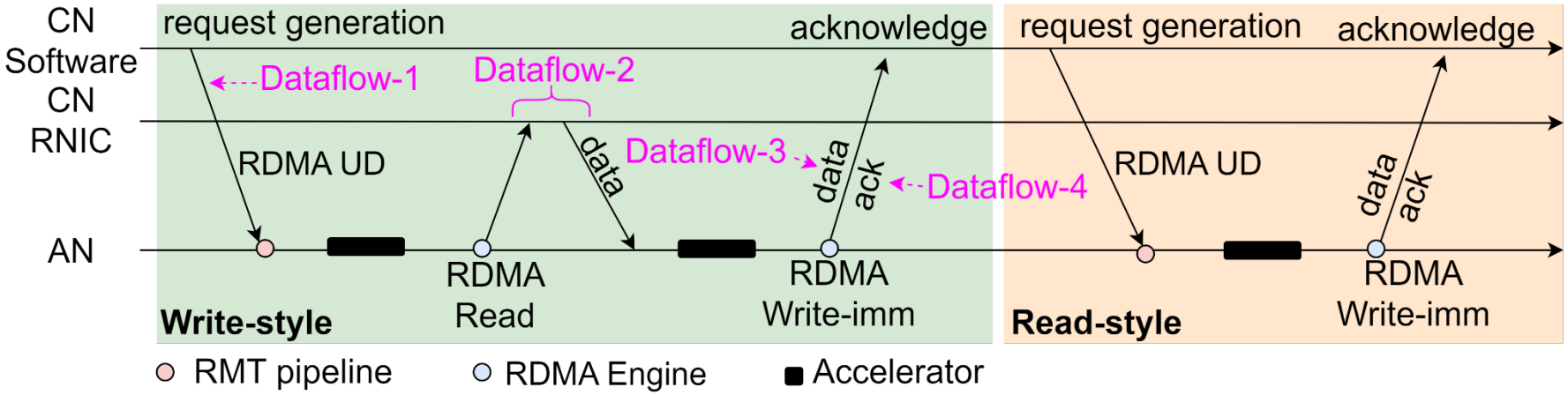
RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-2: transfer the bulk data from the CN to accelerator

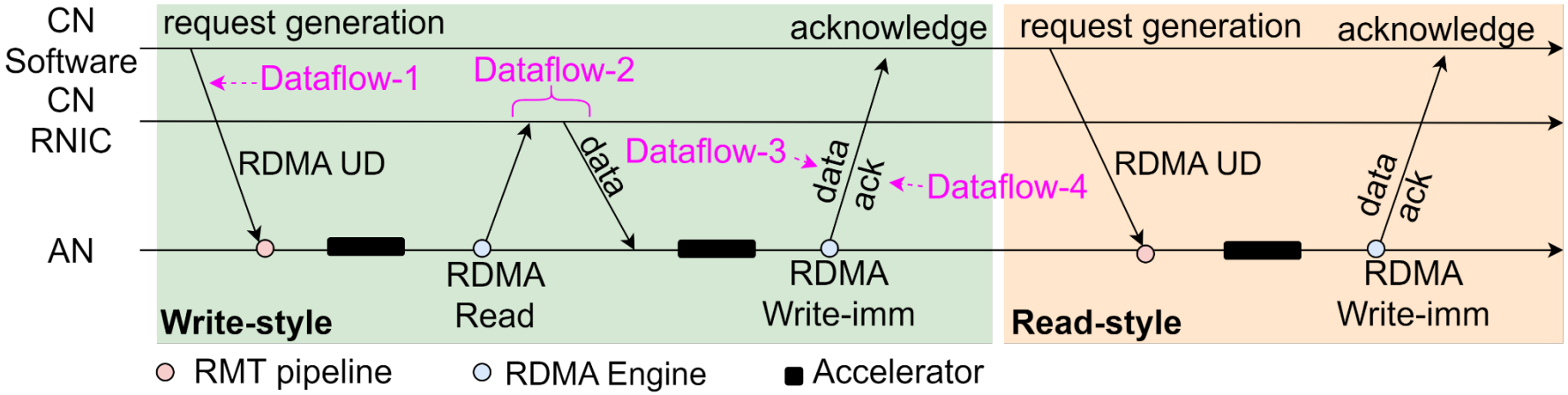
RAAP: the accelerator proactively fetch the data to bypass the CN.

Zero latency jitter introduced by the software.



RAAP

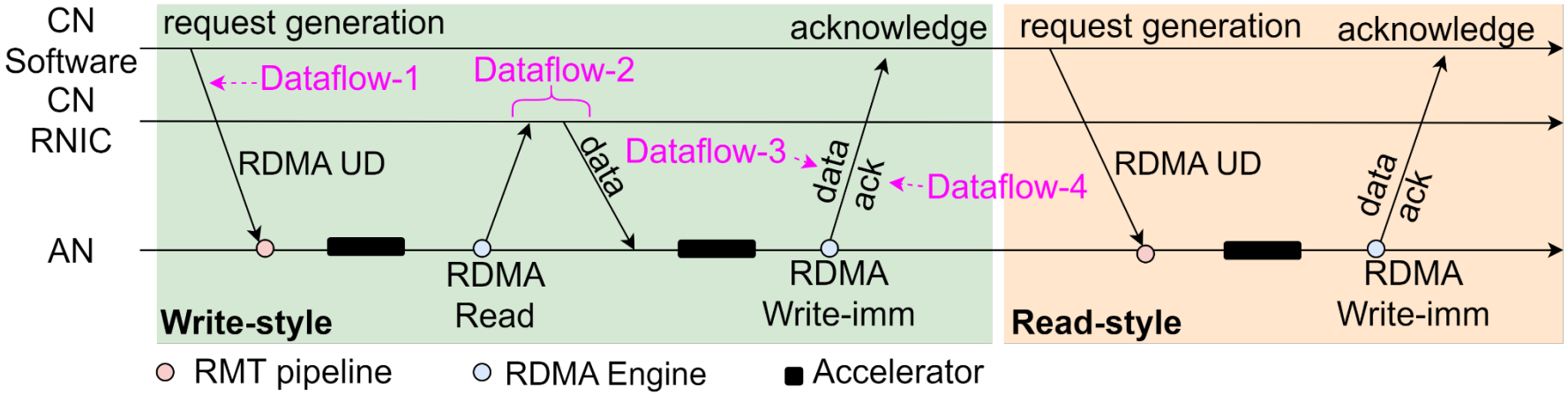
- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.



RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-3 and Dataflow-4: transfer bulk data and acknowledgement from accelerator to CN.

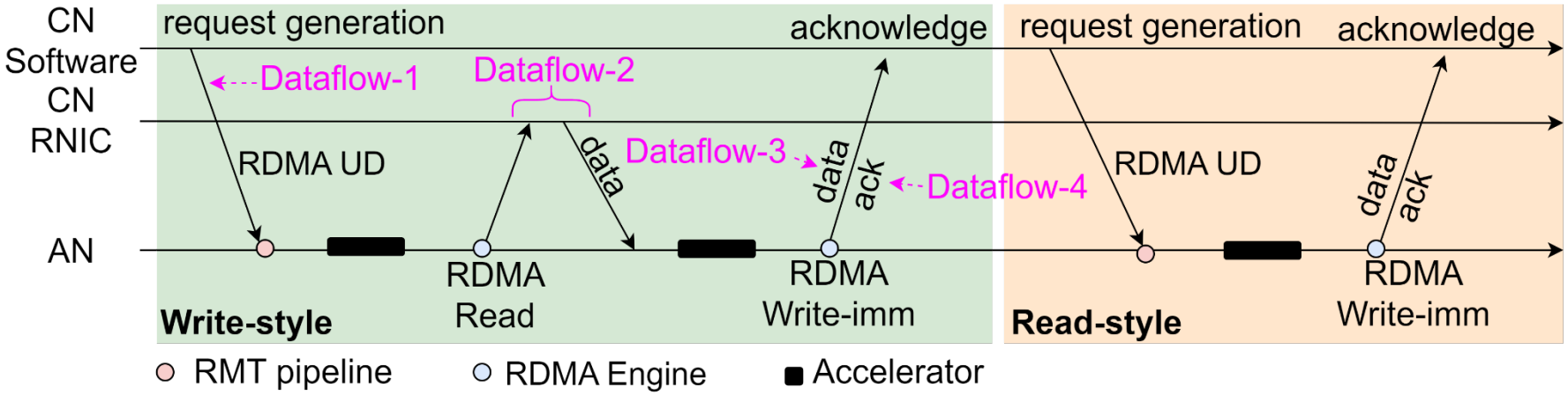


RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-3 and Dataflow-4: transfer bulk data and acknowledgement from accelerator to CN.

RAAP: combine Dataflow-3 and Dataflow-4 an RDMA Write-imm message.



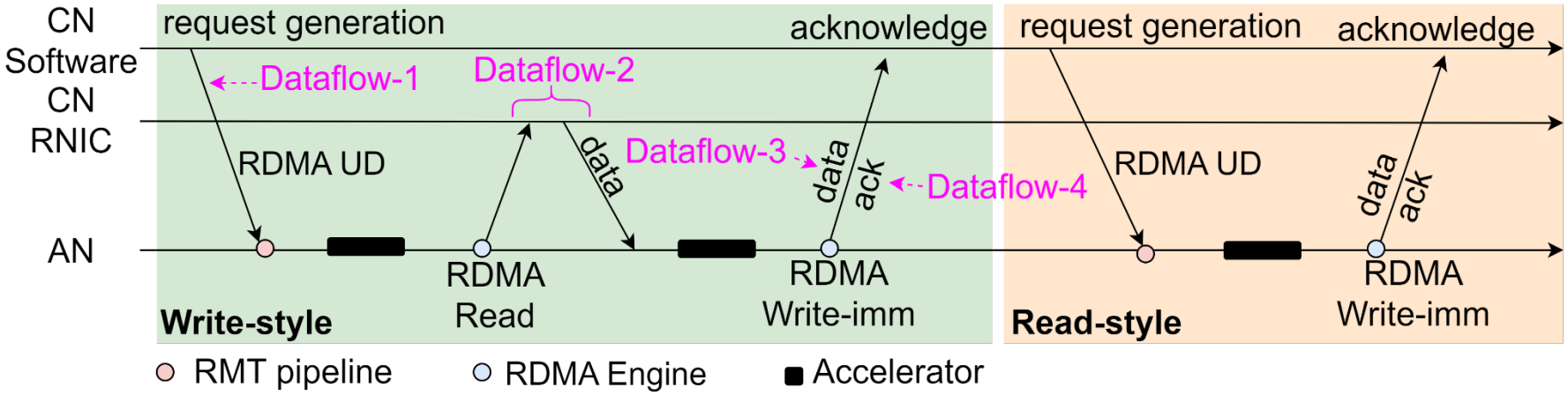
RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.

Dataflow-3 and Dataflow-4: transfer bulk data and acknowledgement from accelerator to CN.

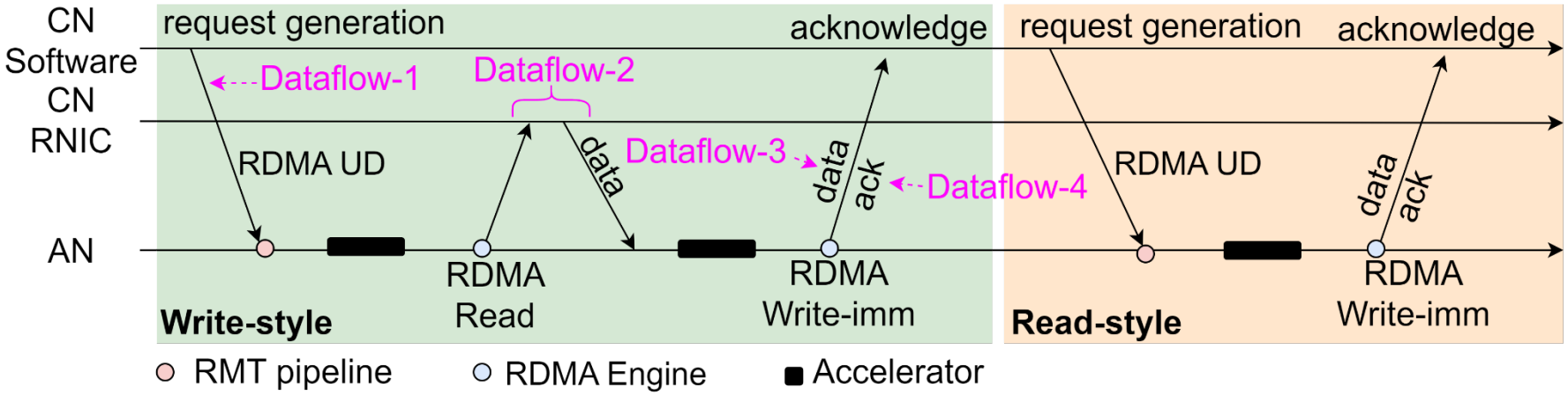
RAAP: combine Dataflow-3 and Dataflow-4 an RDMA Write-imm message.

One round-trip time



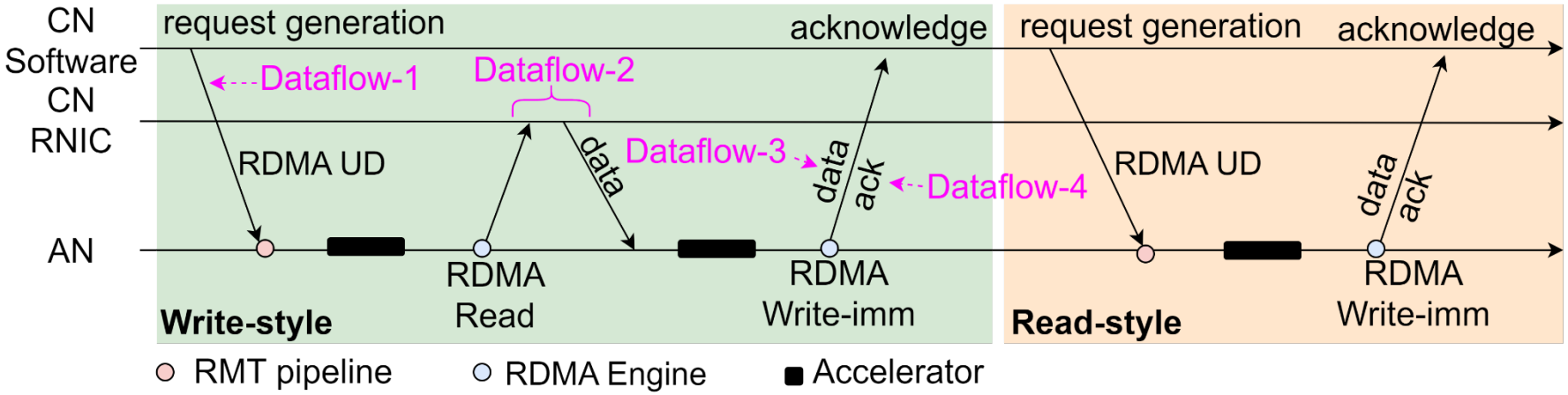
RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.



RAAP

- RAAP: RDMA-based Accelerator Access Pattern, constructing the accelerator semantics over standard RDMA operations for communication between the CN and the accelerator.



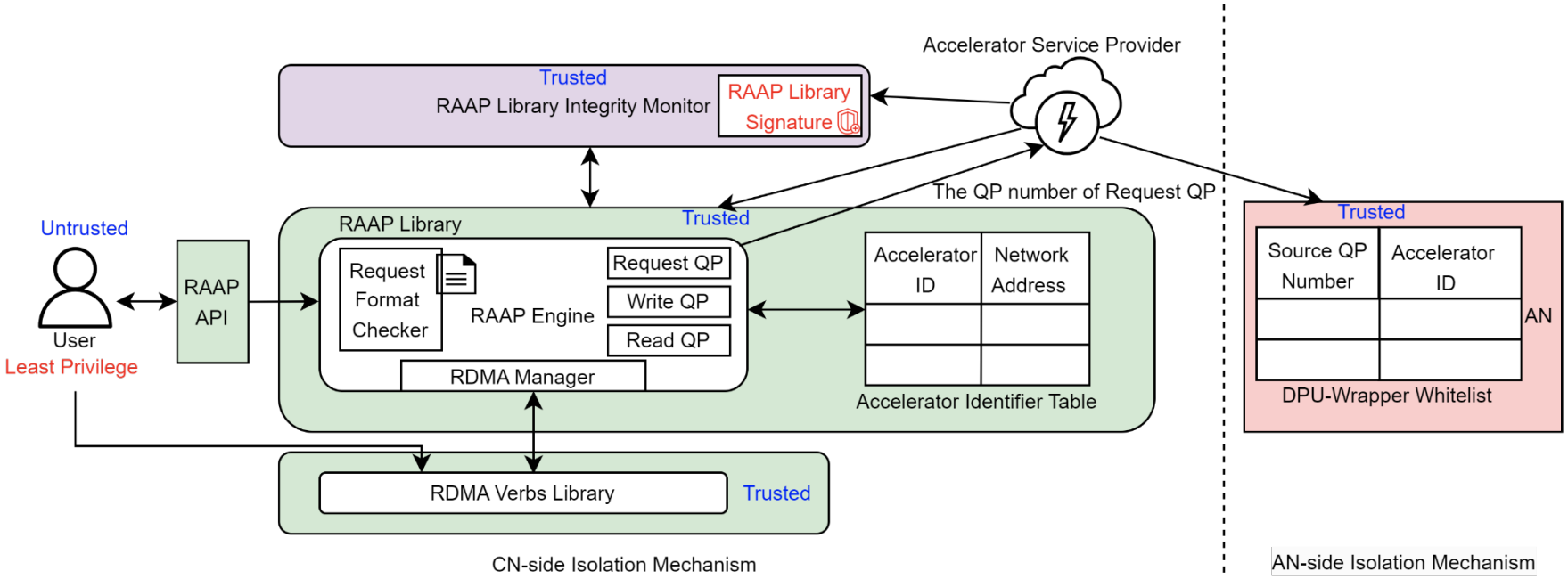
DPU-Direct considers the RDMA operations and the DPU Wrapper microarchitecture.

CN- and AN-side Isolation Mechanism

- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.

CN- and AN-side Isolation Mechanism

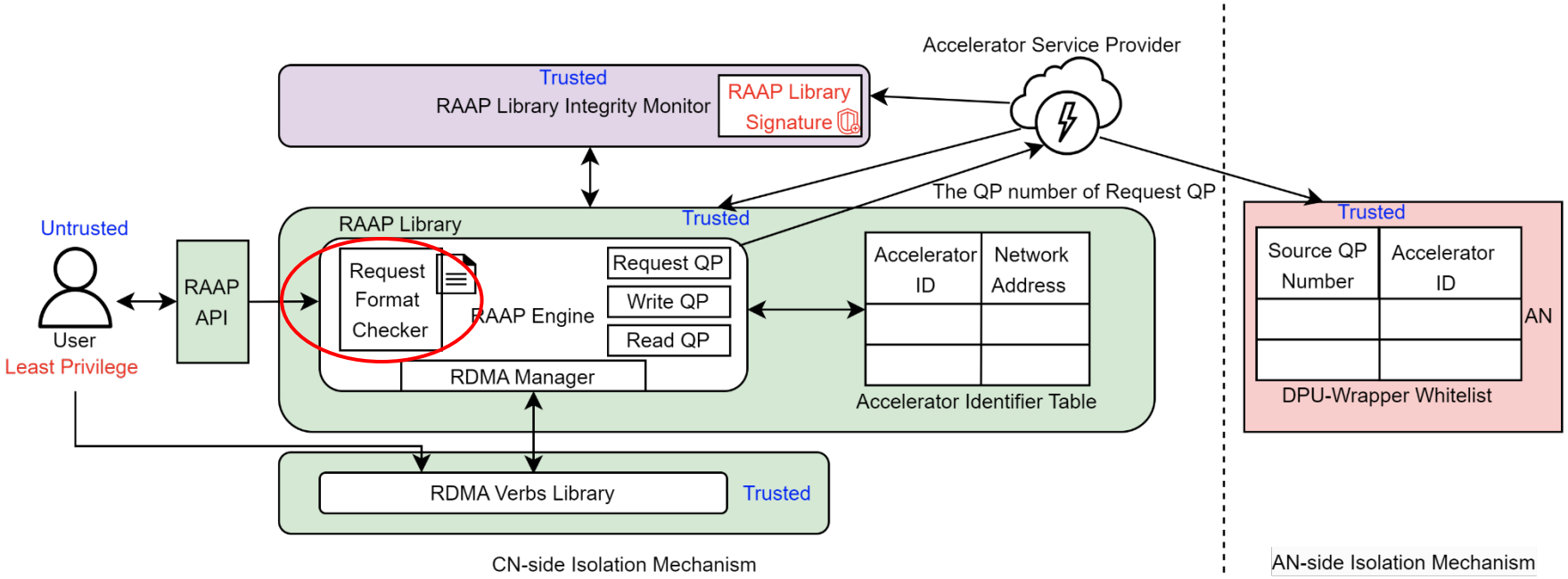
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.



CN- and AN-side Isolation Mechanism

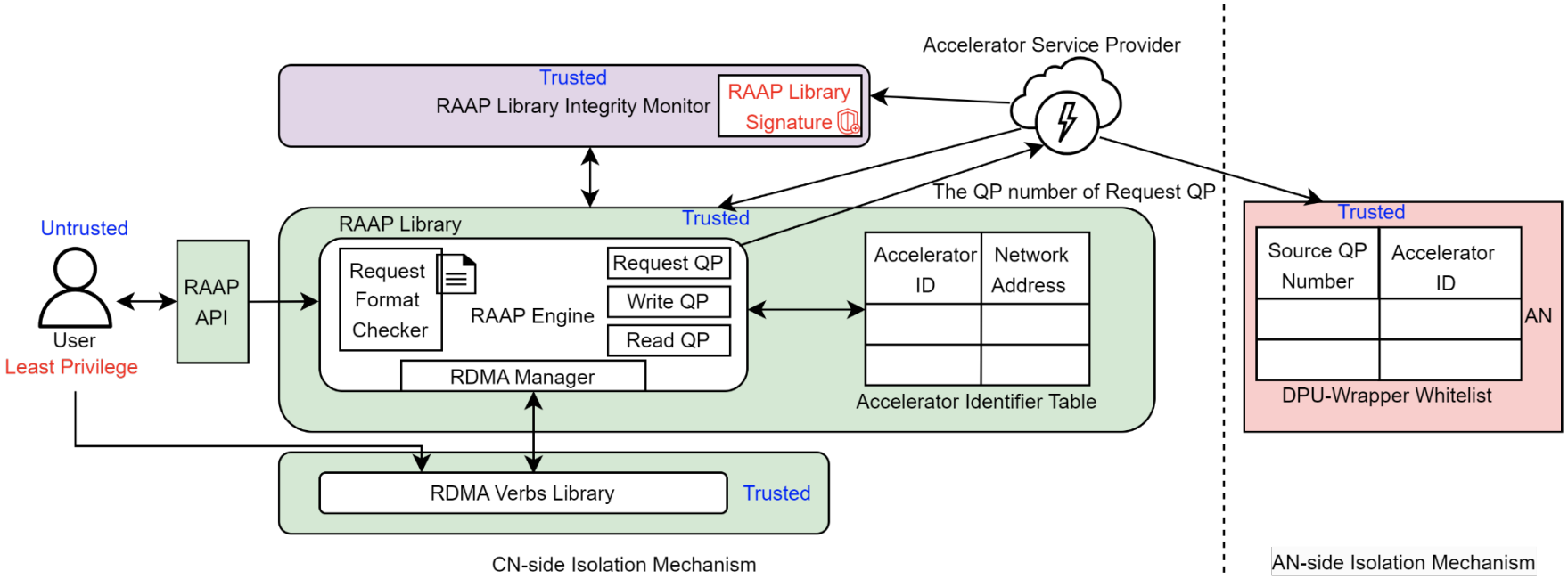
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.

Request Format Checker: guarantee the requests are valid.



CN- and AN-side Isolation Mechanism

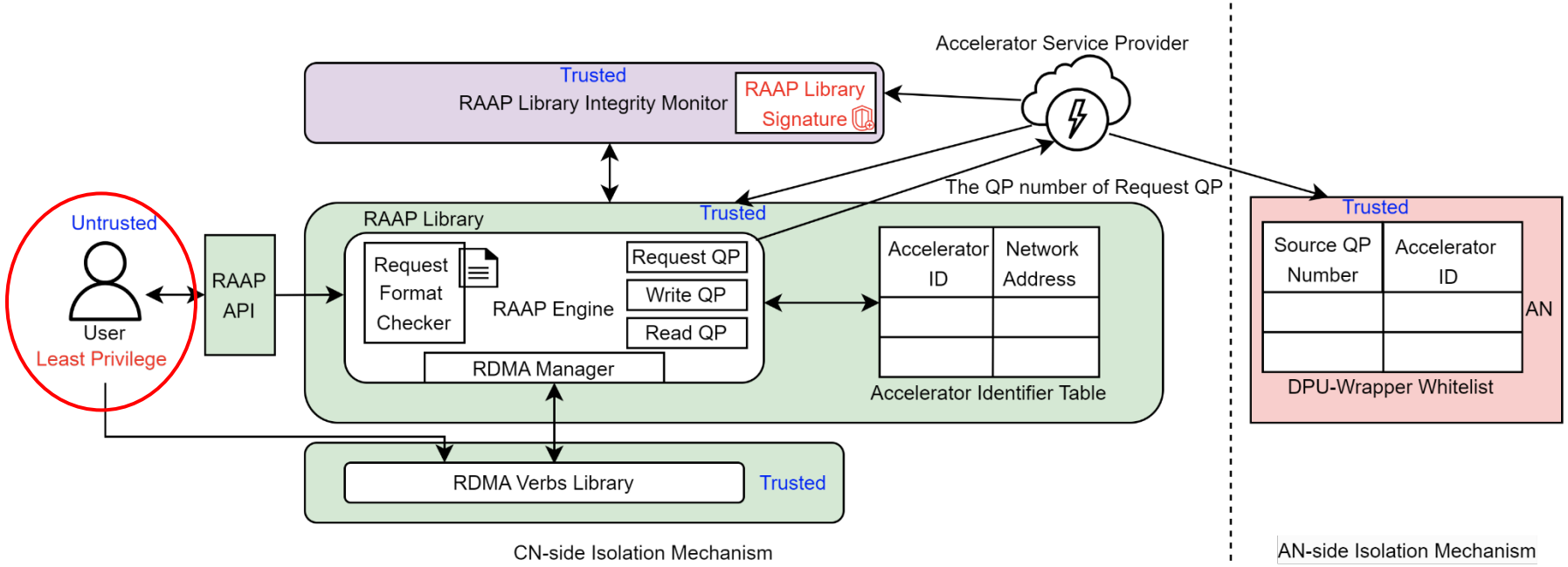
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.



CN- and AN-side Isolation Mechanism

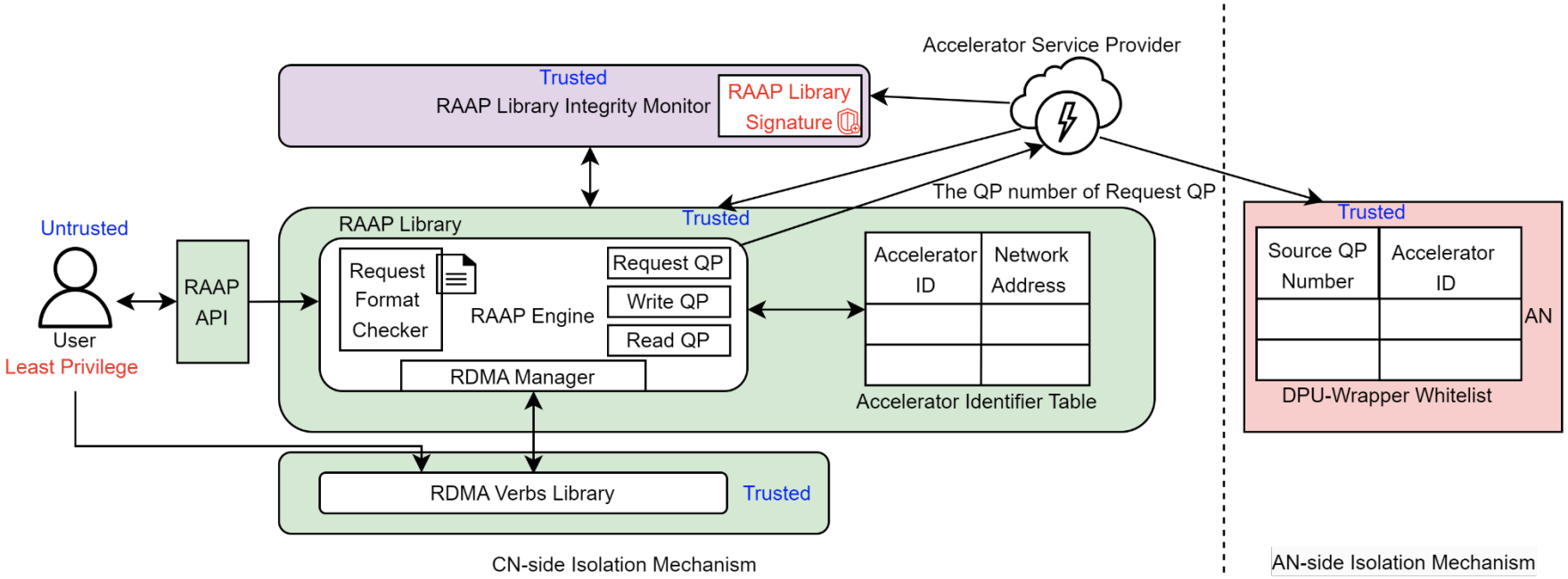
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.

Principle of Least Privilege: the user is given the minimum levels of permission needed to interact with RAAP library.



CN- and AN-side Isolation Mechanism

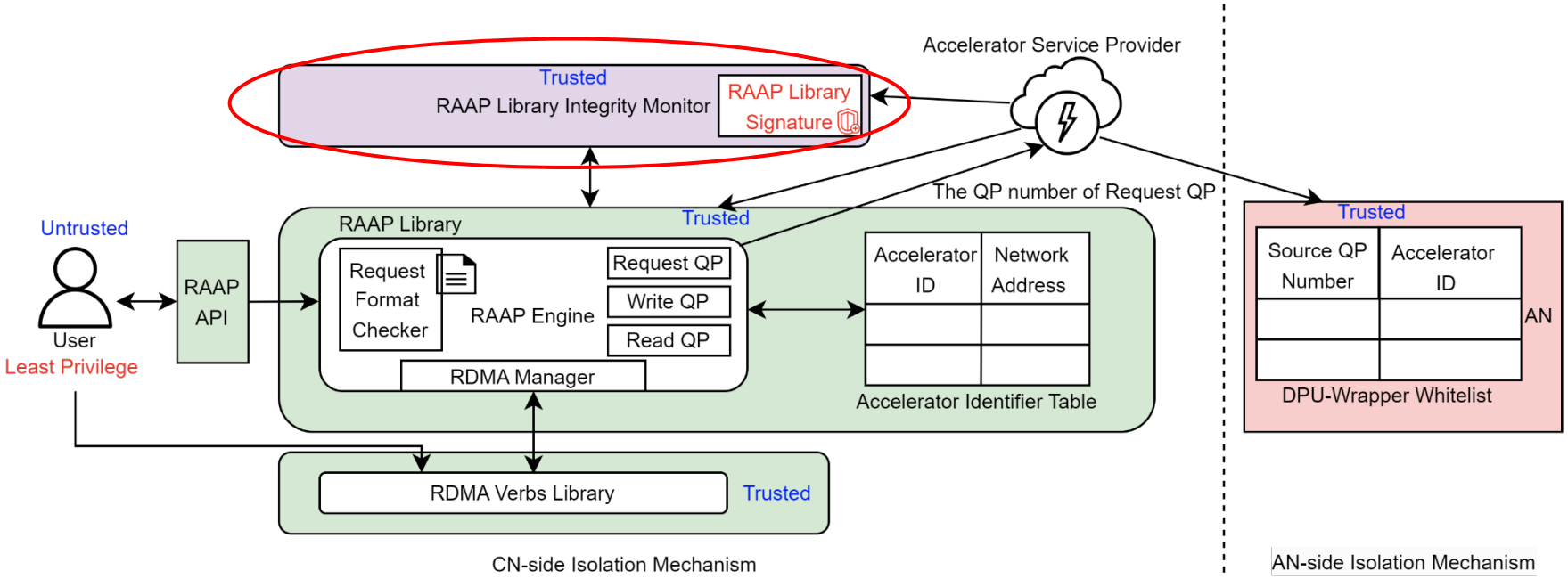
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.



CN- and AN-side Isolation Mechanism

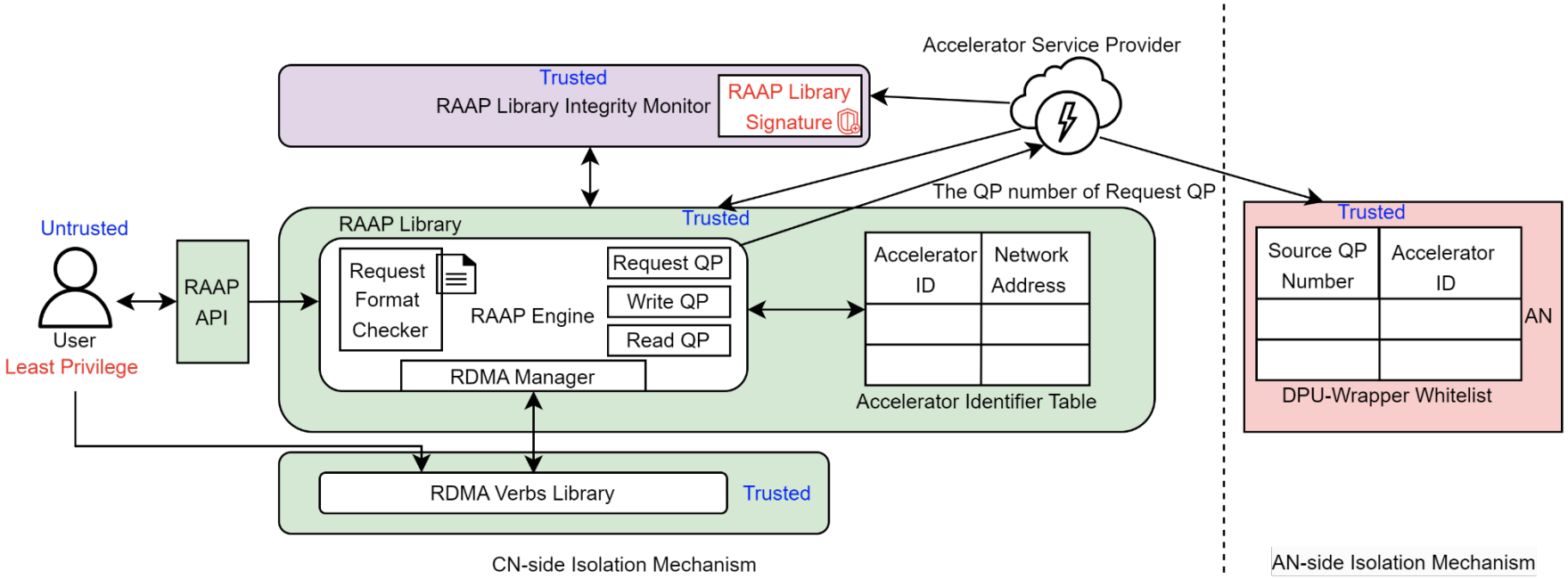
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.

RAAP library integrity monitor uses signatures to protect the RAAP library from being modified.



CN- and AN-side Isolation Mechanism

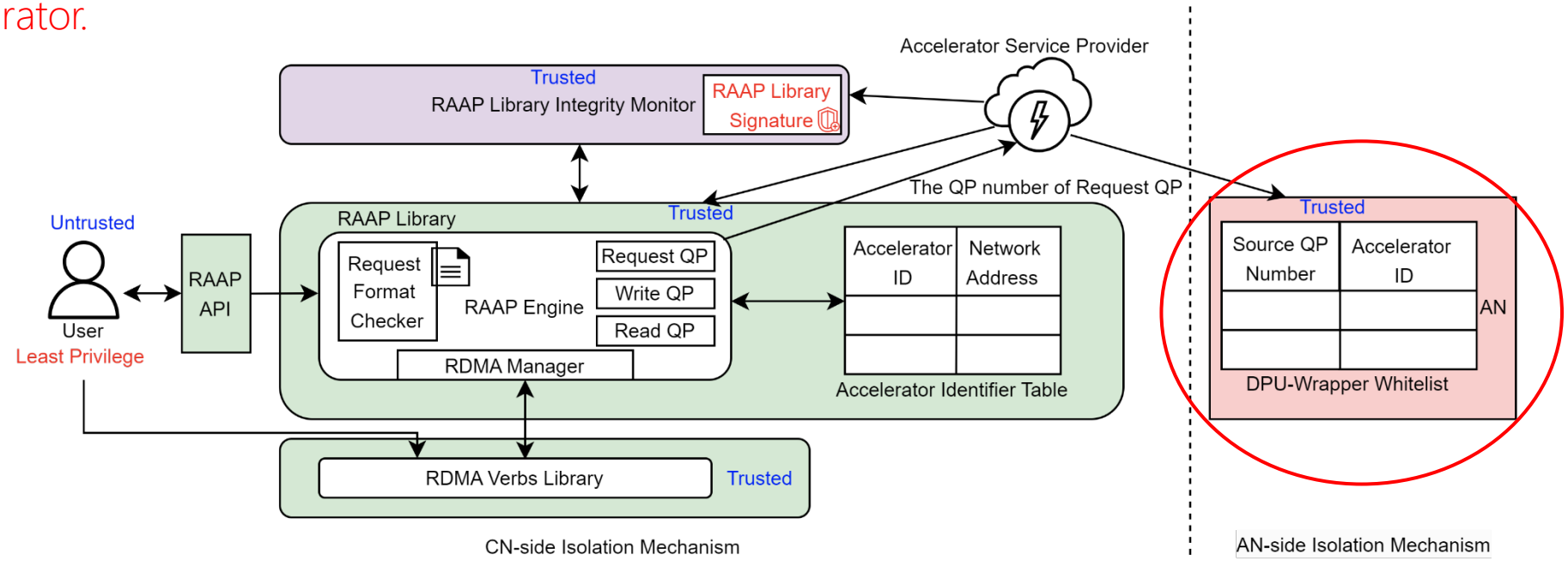
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.



CN- and AN-side Isolation Mechanism

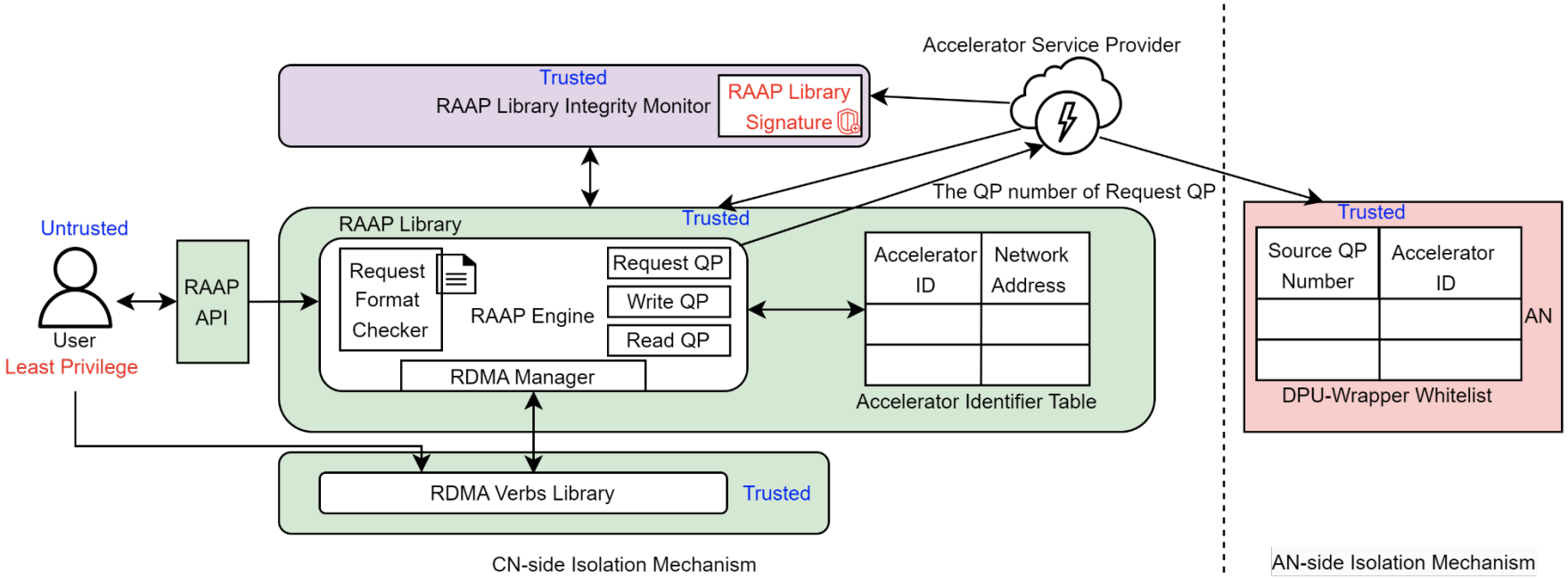
- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.

DPU Wrapper Whitelist: Only requests that has a matched whitelist entry are allowed to invoke the accelerator.



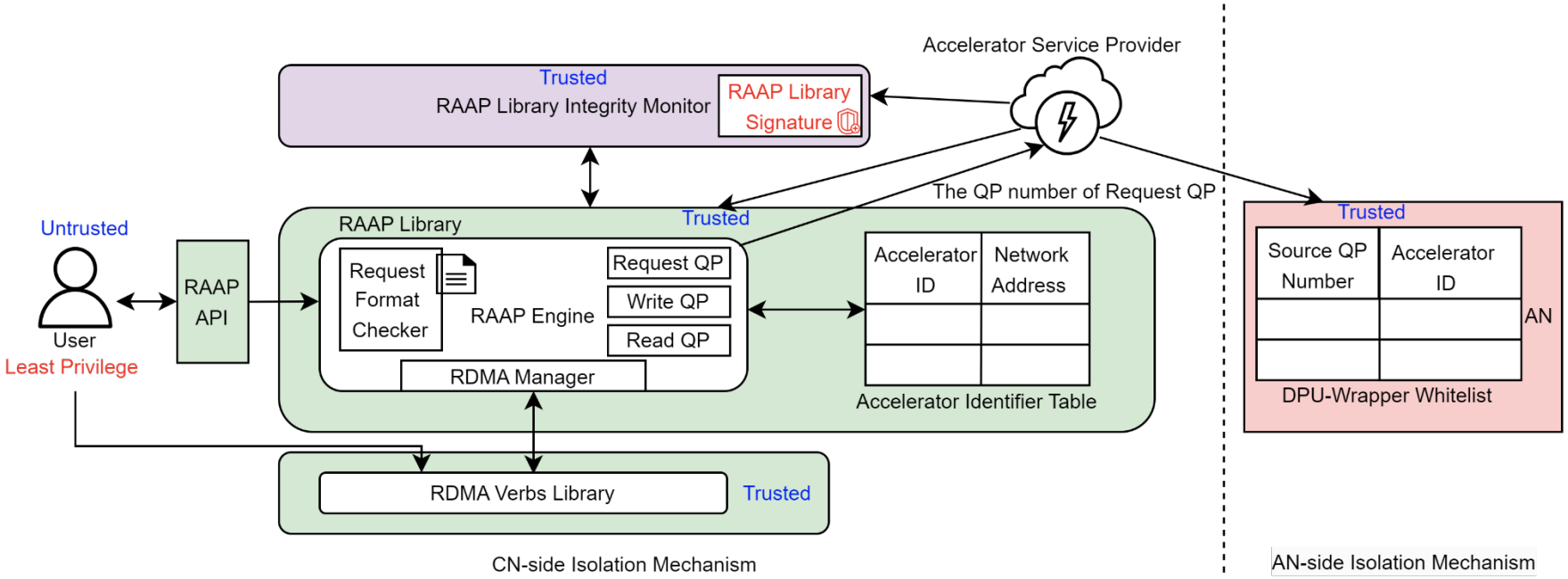
CN- and AN-side Isolation Mechanism

- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.



CN- and AN-side Isolation Mechanism

- Enabling the CN to build their applications upon native RDMA verbs API is dangerous because the malicious CNs may violate the process of RAAP.

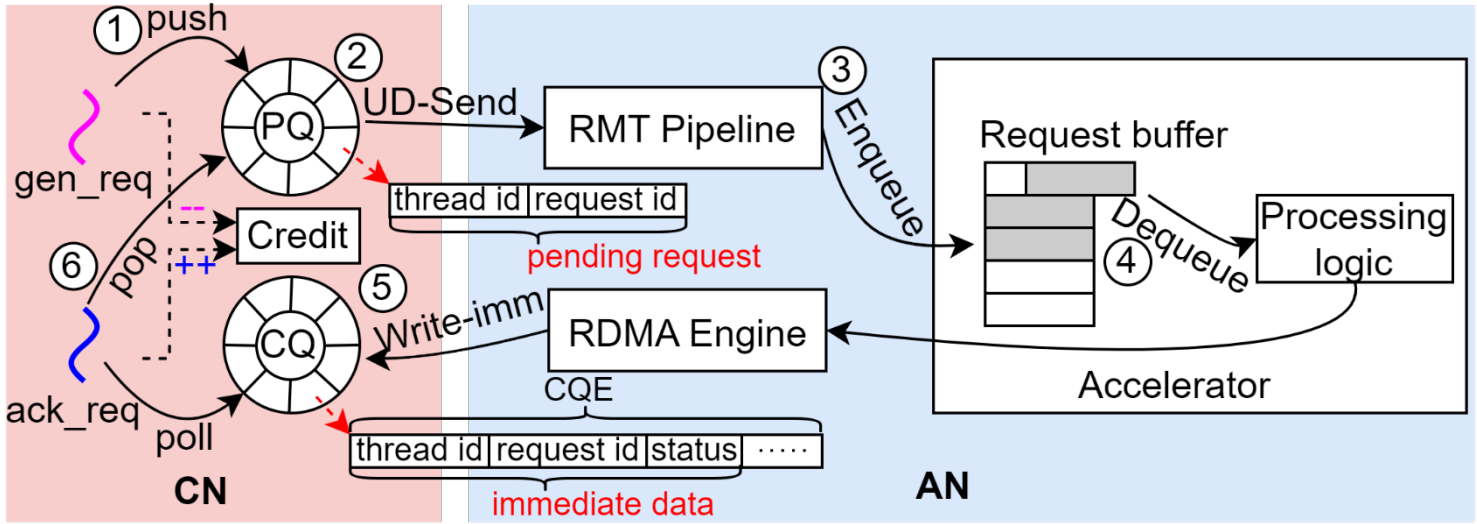


DPU-Direct provides strong security boundary between user application and RAAP library.

RAAP API and Programming Model

- Asynchronous I/O mechanism based on the pending queue (PQ) and completion queue (CQ).
- Credit-based flow control to avoid overwhelming the accelerator.

API	Description
async_post_req	Asynchronous call to post a accelerator request
poll_ack_req	Polling for the latest acknowledgement
async_ack_req	Asynchronous call to fetch the latest acknowledgement

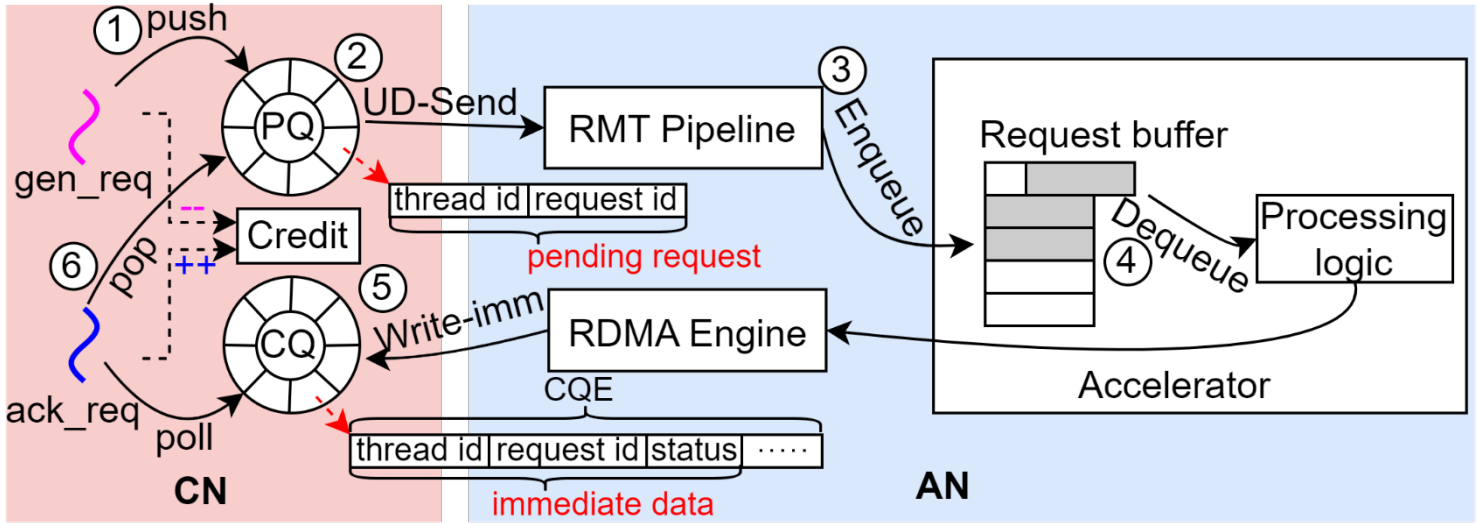


Threading model and credit-based flow control between CN and AN.

RAAP API and Programming Model

- Asynchronous I/O mechanism based on the pending queue (PQ) and completion queue (CQ).
- Credit-based flow control to avoid overwhelming the accelerator.

API	Description
async_post_req	Asynchronous call to post a accelerator request
poll_ack_req	Polling for the latest acknowledgement
async_ack_req	Asynchronous call to fetch the latest acknowledgement



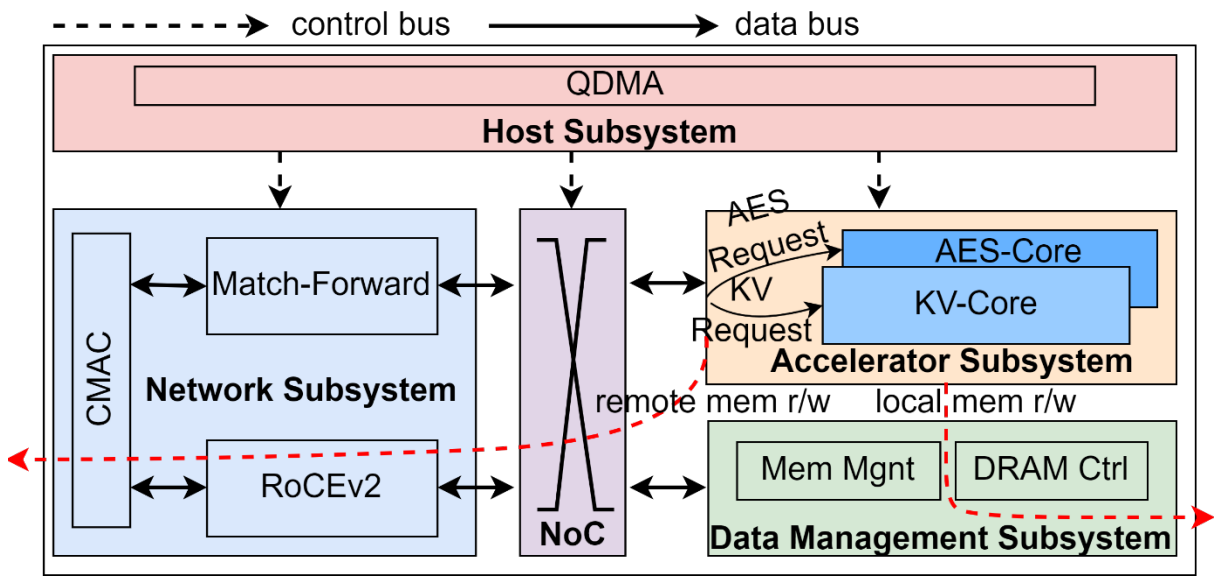
Threading model and credit-based flow control between CN and AN.

DPU-Direct provides well-defined API and programming model.

03 Evaluation

DPU-Direct Prototype

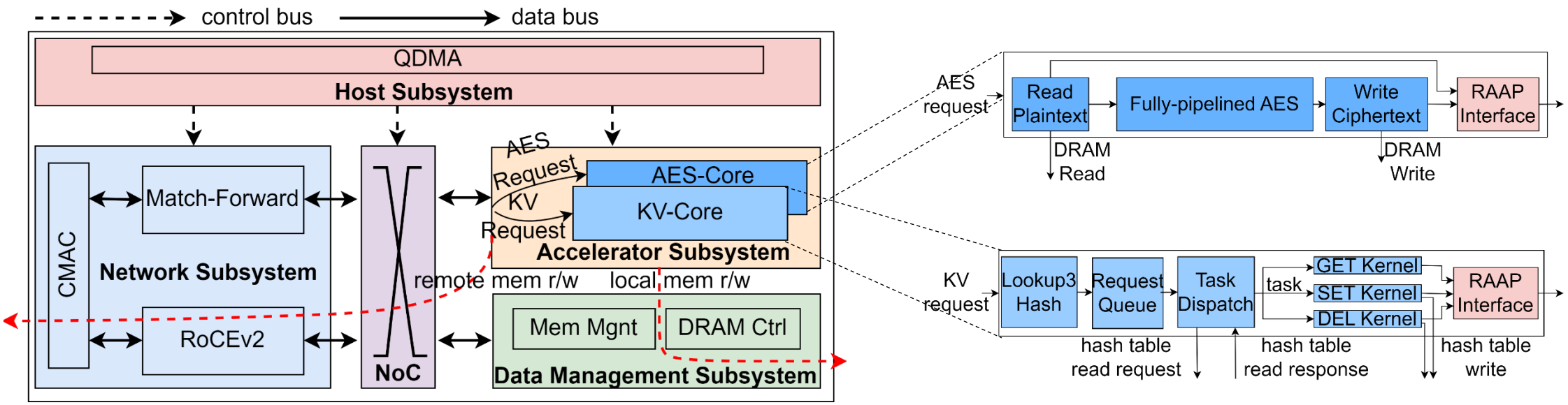
- The AN prototype is based on the open-source OpenNIC[1] project.
- Two **proof-of-concept** use cases.
  - Compute-intensive: AES encryption.
  - Latency-sensitive: key-value cache.



[1]https://github.com/Xilinx/open-nic

DPU-Direct Prototype

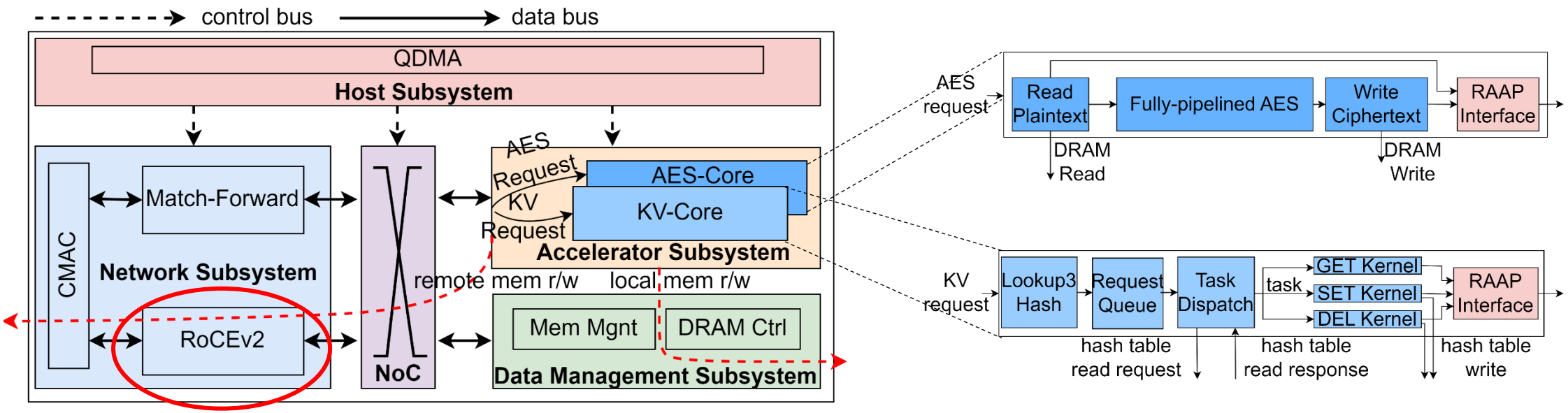
- The AN prototype is based on the open-source OpenNIC[1] project.
- Two **proof-of-concept** use cases.
  - Compute-intensive: AES encryption.
  - Latency-sensitive: key-value cache.



[1]https://github.com/Xilinx/open-nic

DPU-Direct Prototype

- The AN prototype is based on the open-source OpenNIC[1] project.
- Two **proof-of-concept** use cases.
  - Compute-intensive: AES encryption.
  - Latency-sensitive: key-value cache.

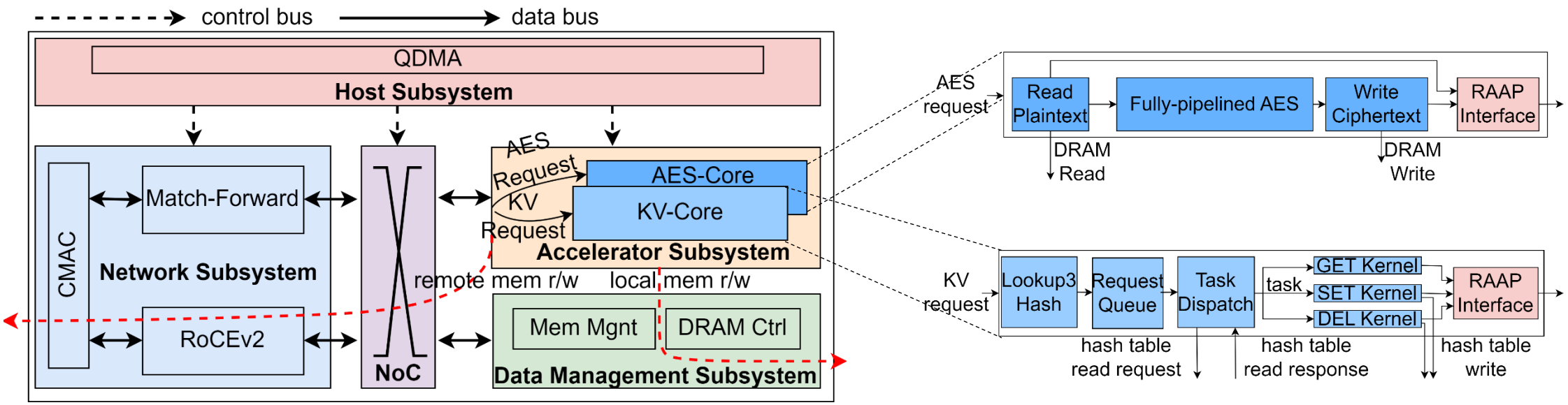


Supported by YUSUR Technology Co., Ltd.

[1]https://github.com/Xilinx/open-nic

DPU-Direct Prototype

- The AN prototype is based on the open-source OpenNIC[1] project.
- Two **proof-of-concept** use cases.
  - Compute-intensive: AES encryption.
  - Latency-sensitive: key-value cache.

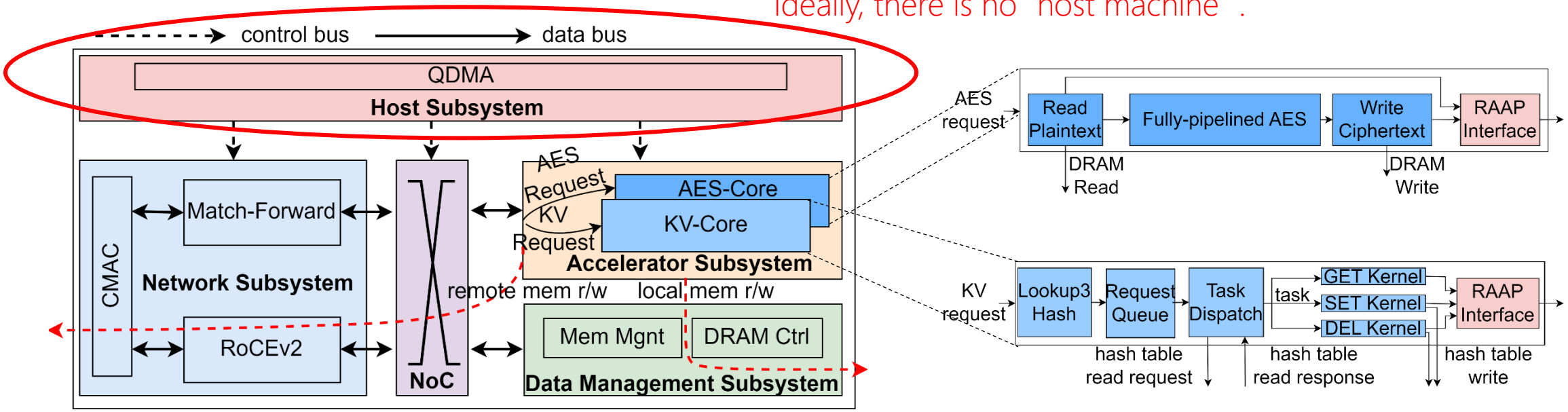


[1]https://github.com/Xilinx/open-nic

DPU-Direct Prototype

- The AN prototype is based on the open-source OpenNIC[1] project.
- Two **proof-of-concept** use cases.
  - Compute-intensive: AES encryption.
  - Latency-sensitive: key-value cache.

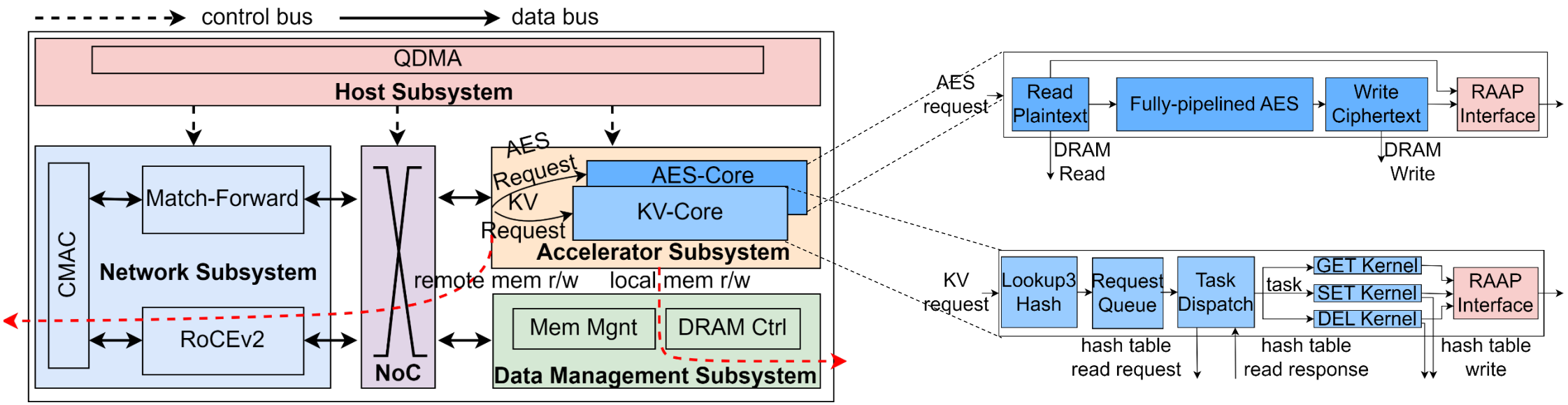
Ideally, there is no "host machine" .



[1]https://github.com/Xilinx/open-nic

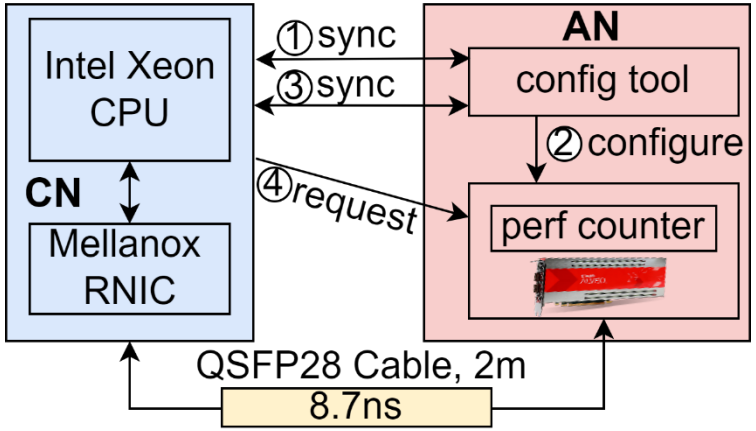
DPU-Direct Prototype

- The AN prototype is based on the open-source OpenNIC[1] project.
- Two **proof-of-concept** use cases.
  - Compute-intensive: AES encryption.
  - Latency-sensitive: key-value cache.

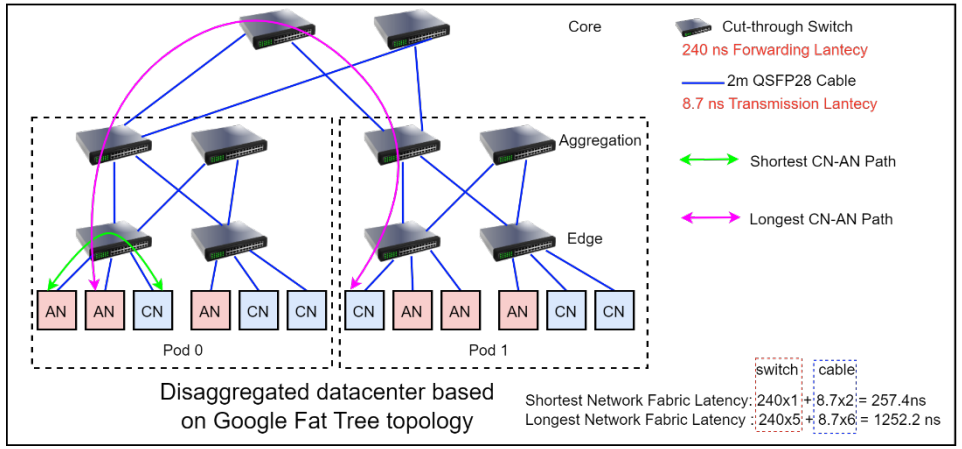


[1]https://github.com/Xilinx/open-nic

Evaluation Setup



Testbed

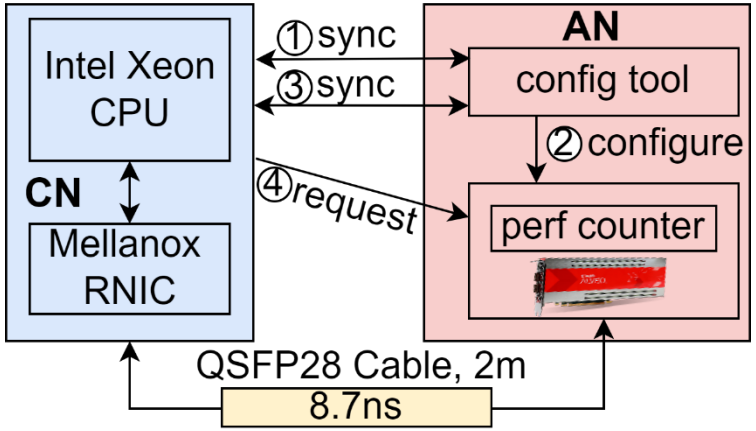


Emulated topology

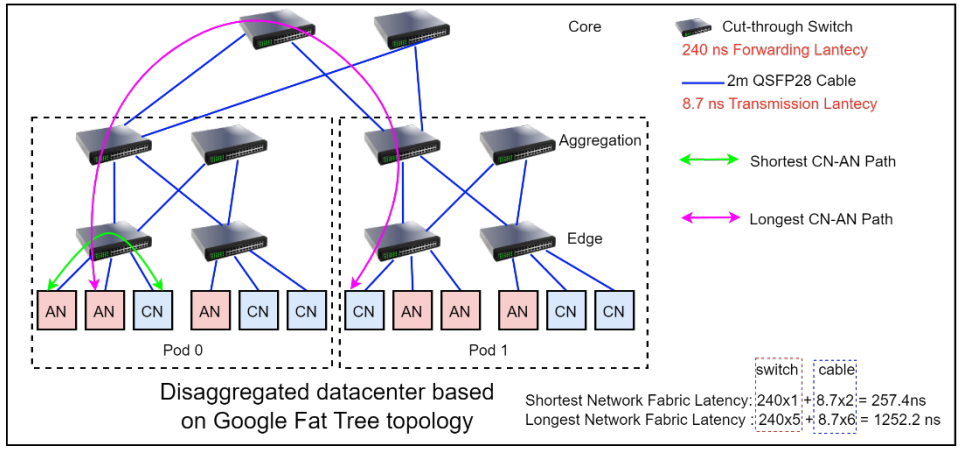
Evaluation Setup

Baselines:

- AES encryption: (1) AES encryption using native CPU instructions; (2) AES encryption based on Intel's AES-NI instruction set.
- Key-value cache: key-value cache server based on RDMA and polling



Testbed



Emulated topology

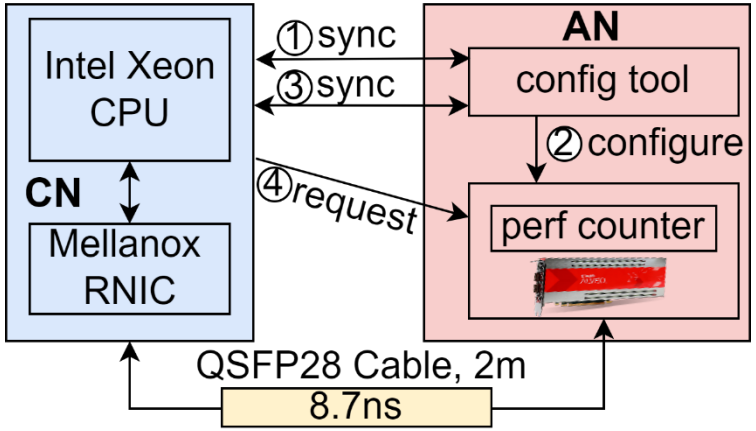
Evaluation Setup

Baselines:

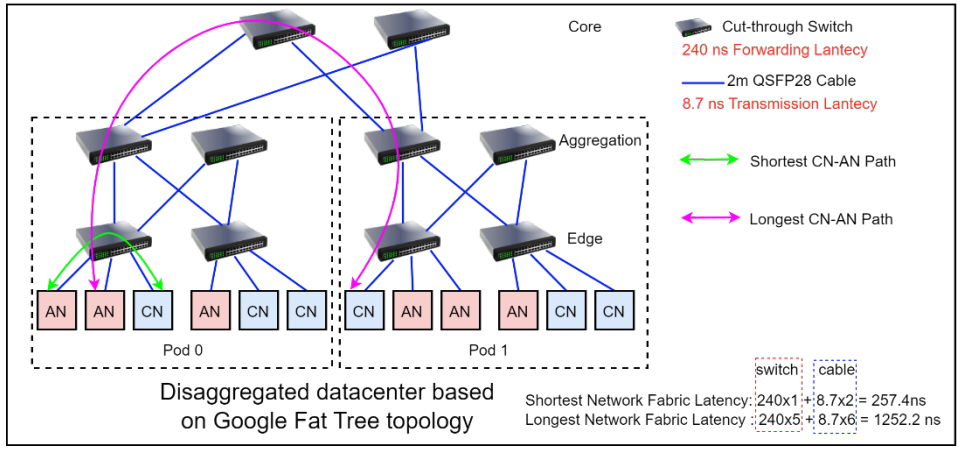
- AES encryption: (1) AES encryption using native CPU instructions; (2) AES encryption based on Intel's AES-NI instruction set.
- Key-value cache: key-value cache server based on RDMA and polling

Workloads:

- AES encryption: random plaintext.
- Key-value cache: YCSB traces.



Testbed



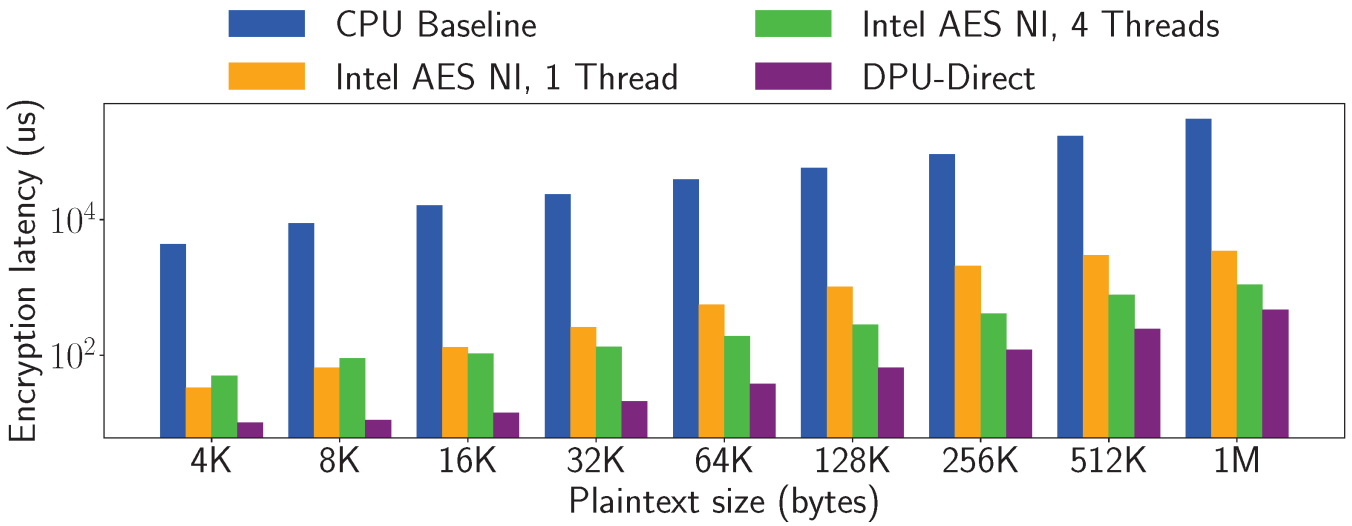
Emulated topology

AES Encryption

When accelerators are disaggregated through DPU-Direct, will they still provide the expected performance improvement?

AES Encryption

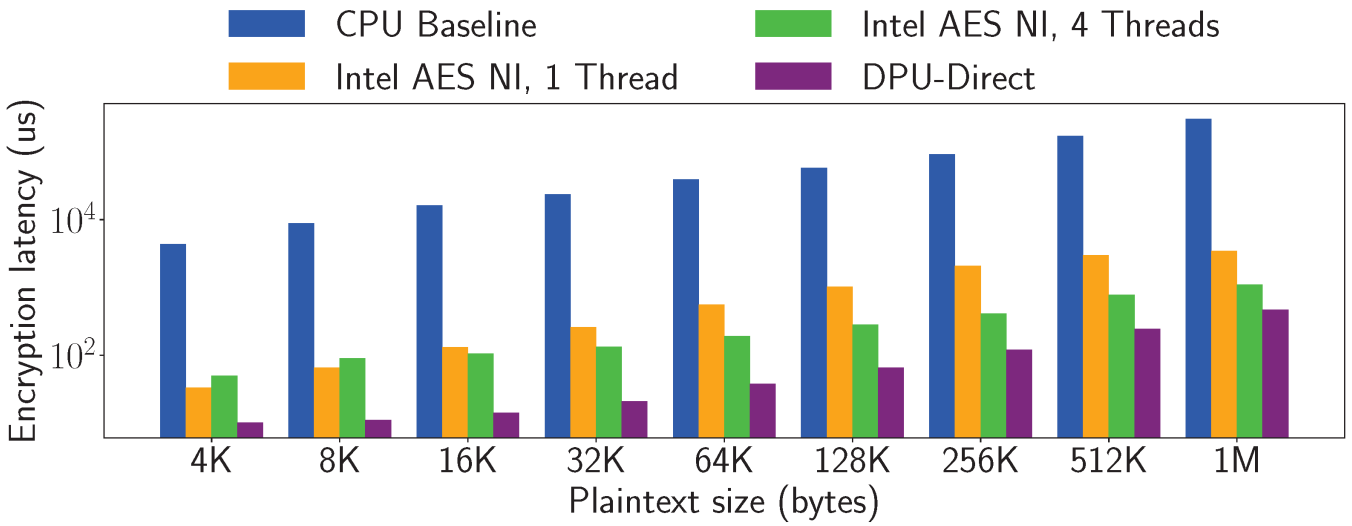
When accelerators are disaggregated through DPU-Direct, will they still provide the expected performance improvement?

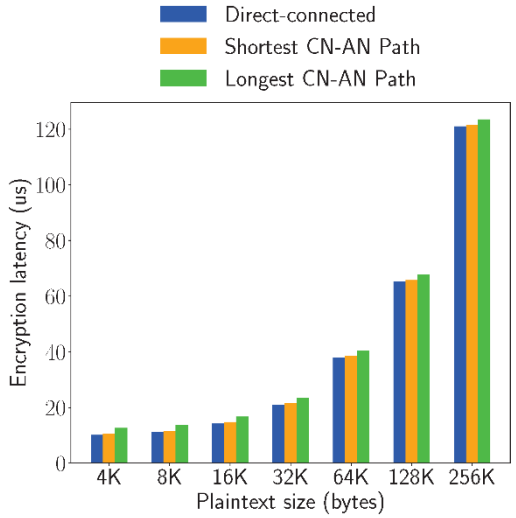


- Compared with the CPU baseline, DPU-Direct can achieve a speedup of at least 400x and at most 1000x.
- Compared four-thread Intel AES NI, DPU-Direct can still achieve about 2~7x speedup.

AES Encryption

When accelerators are disaggregated through DPU-Direct, will they still provide the expected performance improvement?



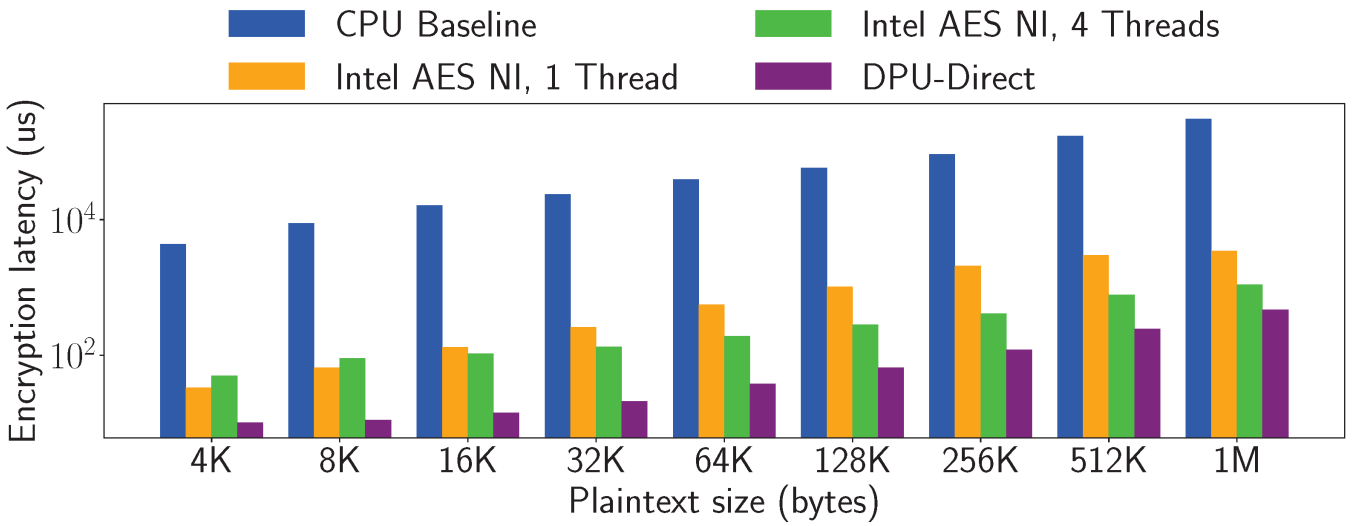


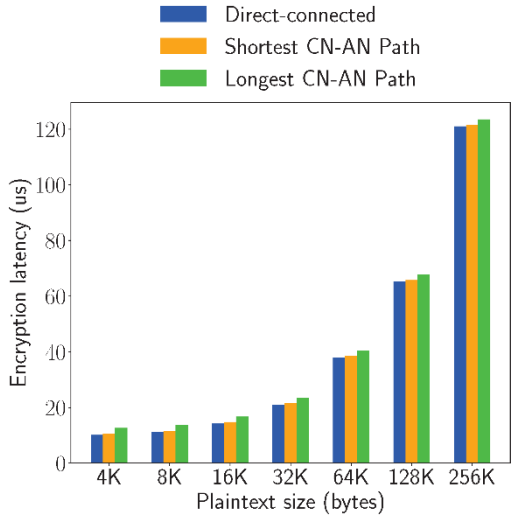
- Compared with the CPU baseline, DPU-Direct can achieve a speedup of at least 400x and at most 1000x.
- Compared four-thread Intel AES NI, DPU-Direct can still achieve about 2~7x speedup.

- When the computation-intensive accelerator processes a large volume of data, the interference of network fabric latency on the performance of DPU-Direct can be negligible.

AES Encryption

When accelerators are disaggregated through DPU-Direct, will they still provide the expected performance improvement?





- Compared with the CPU baseline, DPU-Direct can achieve a speedup of at least 400x and at most 1000x.
- Compared four-thread Intel AES NI, DPU-Direct can still achieve about 2~7x speedup.

- When the computation-intensive accelerator processes a large volume of data, the interference of network fabric latency on the performance of DPU-Direct can be negligible.

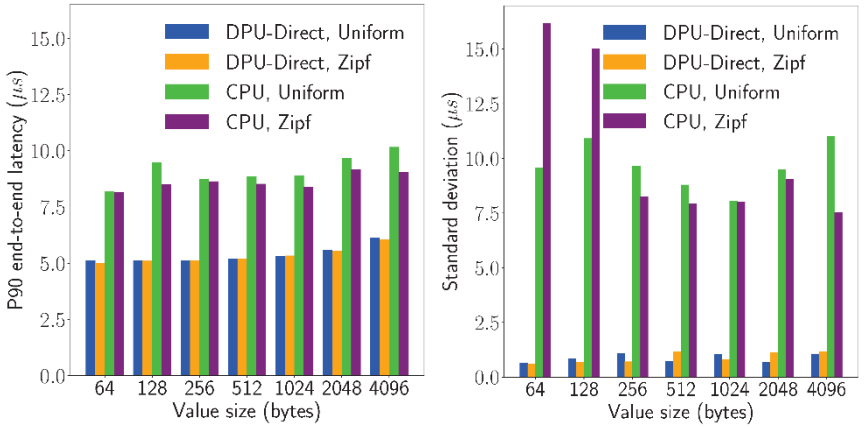
Computation-intensive accelerators on DPU-Direct is comparable with the local accelerator.

Key-value Cache

How DPU-Direct benefits latency-sensitive applications?

Key-value Cache

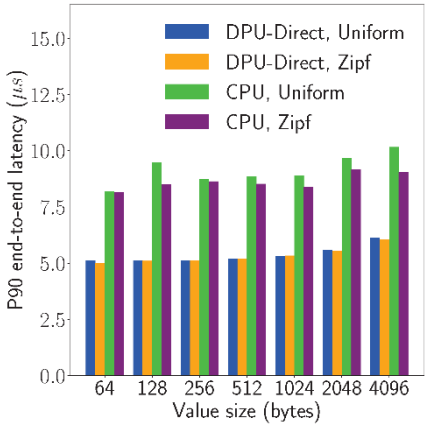
How DPU-Direct benefits latency-sensitive applications?

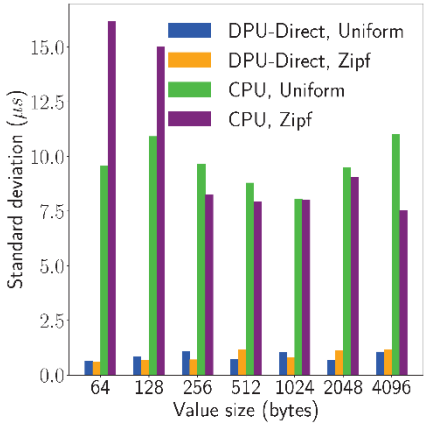


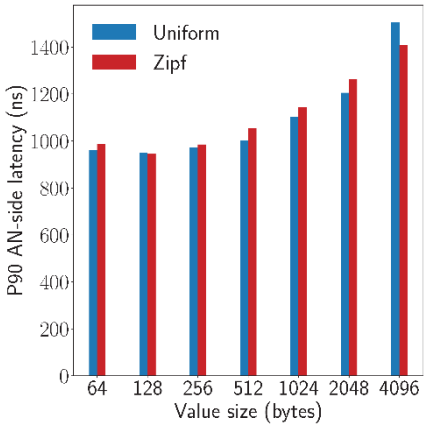
- The DPU-Direct prototype outperforms the CPU-RDMA-Polling baseline regarding absolute latency and jitter.
- Direct achieves microsecond-level jitter.

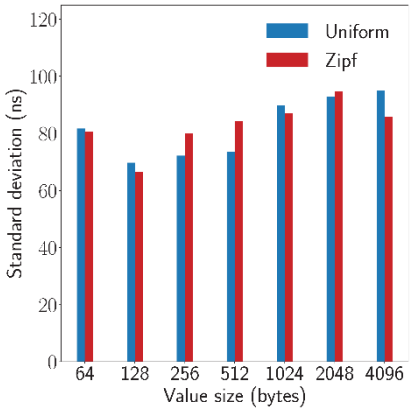
Key-value Cache

How DPU-Direct benefits latency-sensitive applications?







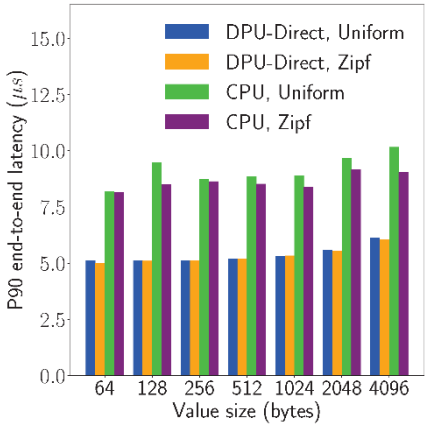


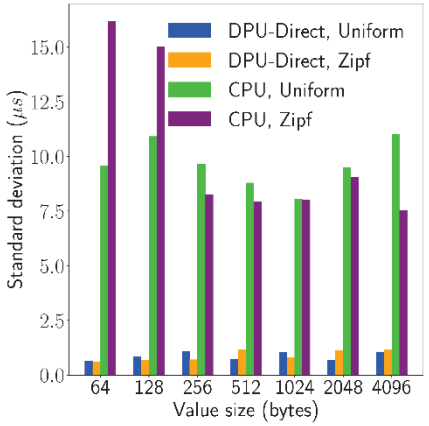
- The DPU-Direct prototype outperforms the CPU-RDMA-Polling baseline regarding absolute latency and jitter.
- Direct achieves microsecond-level jitter.

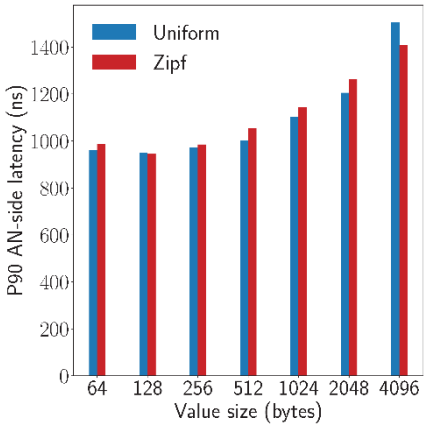
- Latency and jitter within the AN is extreme low.
- Most latency and jitter come from the CN.

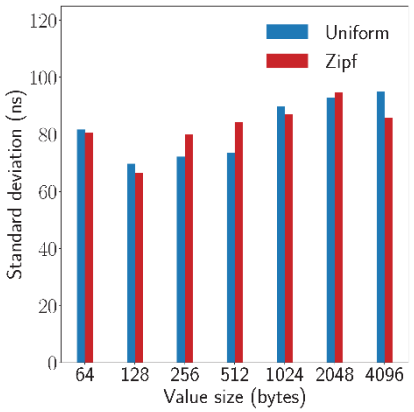
Key-value Cache

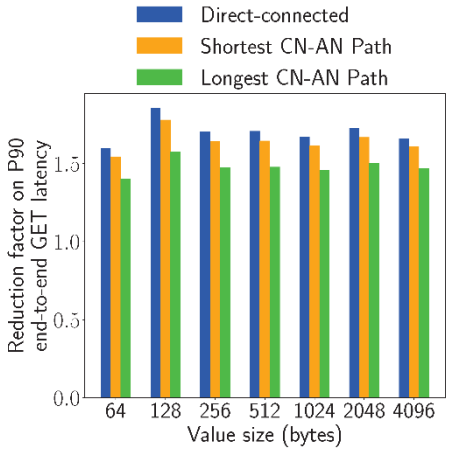
How DPU-Direct benefits latency-sensitive applications?











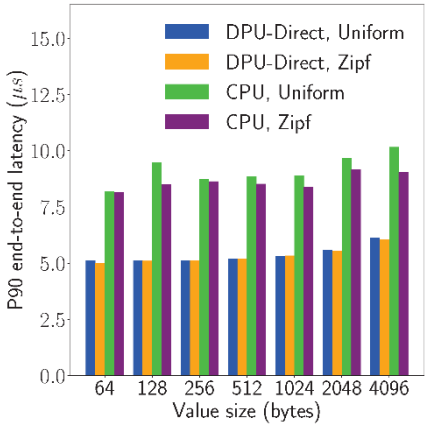
- The DPU-Direct prototype outperforms the CPU-RDMA-Polling baseline regarding absolute latency and jitter.
- Direct achieves microsecond-level jitter.

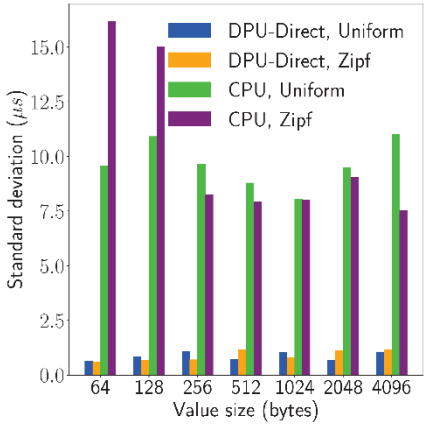
- Latency and jitter within the AN is extreme low.
- Most latency and jitter come from the CN.

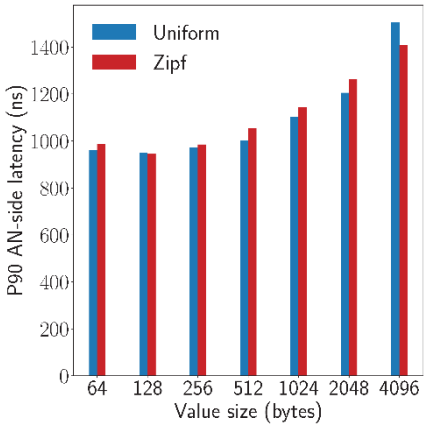
- When latency-sensitive accelerators are disaggregated through DPU-Direct, the reduction factor of end-to-end latency will be reduced to a certain extent.

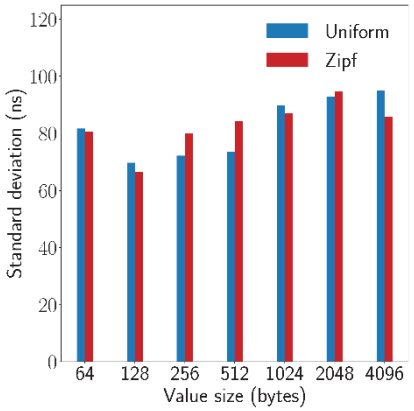
Key-value Cache

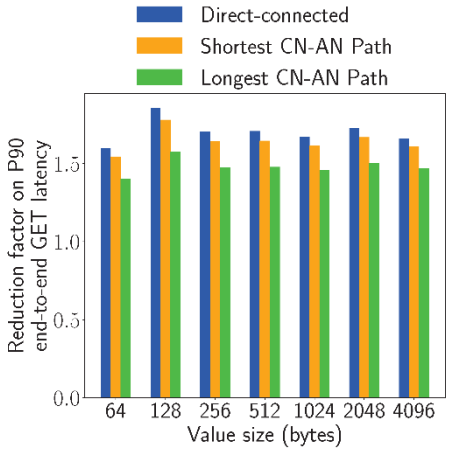
How DPU-Direct benefits latency-sensitive applications?











- The DPU-Direct prototype outperforms the CPU-RDMA-Polling baseline regarding absolute latency and jitter.
- Direct achieves microsecond-level jitter.

- Latency and jitter within the AN is extreme low.
- Most latency and jitter come from the CN.

- When latency-sensitive accelerators are disaggregated through DPU-Direct, the reduction factor of end-to-end latency will be reduced to a certain extent.

DPU-Direct provides promising latency and jitter reduction to latency-sensitive applications.

04 Summary

Summary

- DPU-Direct: a holistic solution for disaggregated accelerator node.
  - DPU Wrapper: turn accelerators into disaggregation-native device.
  - RAAP: overlay accelerator semantics based on standard RDMA network.
  - DPU-Direct AAPI: provide accelerator interface for applications.
- For compute-intensive accelerators, DPU-Direct provides close-to-local performance.
- Latency-sensitive applications built on DPU-Direct can obtain extreme low latency and low jitter.
  - Accelerator reduce the application-logic latency.
  - Hardware-based network stack (RDMA) reduce the communication latency.

Reference

[1] Gan, Yu, et al. "An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems." Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems. 2019.
[2] Cheng, Yue, Ali Anwar, and Xuejing Duan. "Analyzing alibaba's co-located datacenter workloads." 2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018.
[3] Zhang, Yingqiang, et al. "Towards cost-effective and elastic cloud database deployment via memory disaggregation." Proceedings of the VLDB Endowment 14.10 (2021): 1900-1912.
[4] Hennessy, John L., and David A. Patterson. Computer architecture: a quantitative approach, Sixth Edition. Elsevier, 2019.
[5] Arunkumar, Akhil, et al. "MCM-GPU: Multi-chip-module GPUs for continued performance scalability." ACM SIGARCH Computer Architecture News 45.2 (2017): 320-332.
[6] Sidler, David, et al. "StRoM: smart remote memory." Proceedings of the Fifteenth European Conference on Computer Systems. 2020.