

Using DPU-based SmartNICs for Database Management

Philip A. Bernstein
Microsoft Research

Jiasheng Hu, Qizhen Zhang
University of Toronto

Yuanfan Chen, Chihan Cui, Anna Li, Jialin Li (NUS),
Sidharth Sankhe, Raahil Vora, Kaiwen Zheng

21st HPTS
October 6, 2026

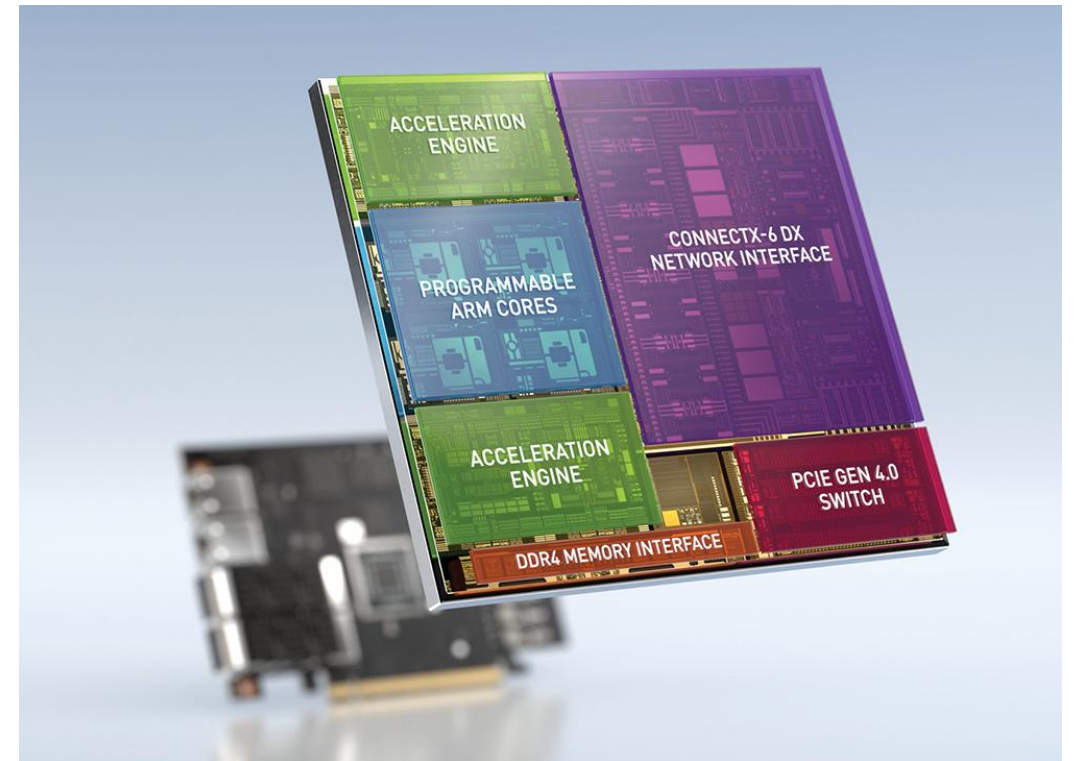
Smart Network Interface Card (SmartNIC)

- SmartNICs offload tasks that the system CPU would normally do
 - Use their own onboard processors to save host CPUs and improve performance
- Three types of SmartNICs: ASIC-based, FPGA-based, SoC-based (DPUs)



Data Processing Units (DPUs)

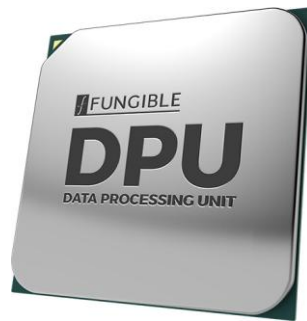
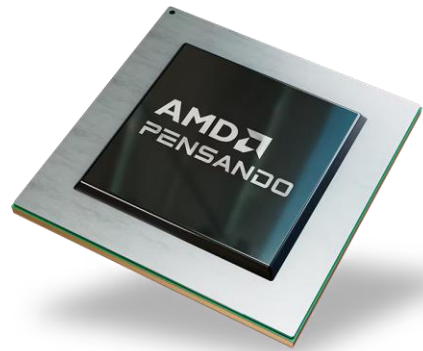
- Multi-core CPUs
 - Low latency, lower cost than host, but lower throughput than host
- ASIC accelerators for data path tasks
 - DMA, compression, encryption, CRC, secure hash, regex, ...
 - Low cost, high throughput, but high latency
- High-speed network connectivity
- Onboard memory, e.g., 8 GB - 32 GB
- PCIe connection to host resources



NVIDIA BlueField Networking Platform, nvidia.com

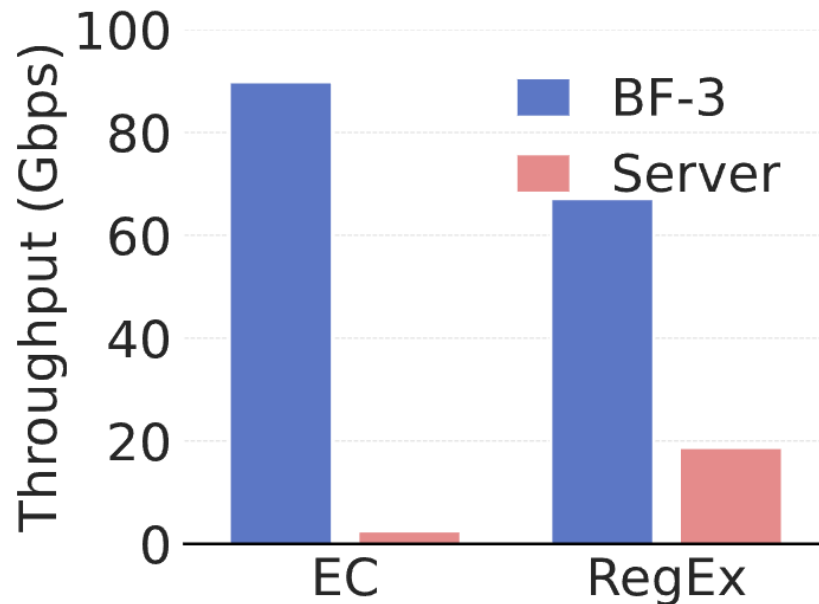
DPU's are Popular

Major silicon vendors and cloud providers have DPU products



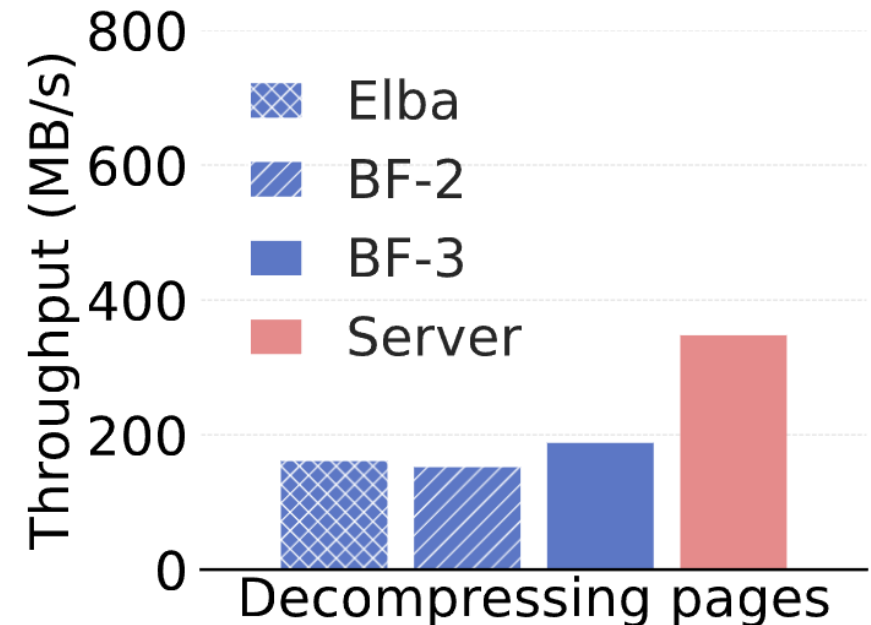
Compute Efficiency of DPUs

Hardware accelerators (erasure coding & regex)



BF (Bluefield) – NVIDIA Mellanox
Elba – AMD Pensando

SoC cores on the DPU



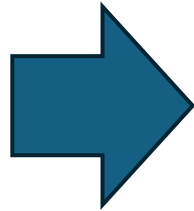
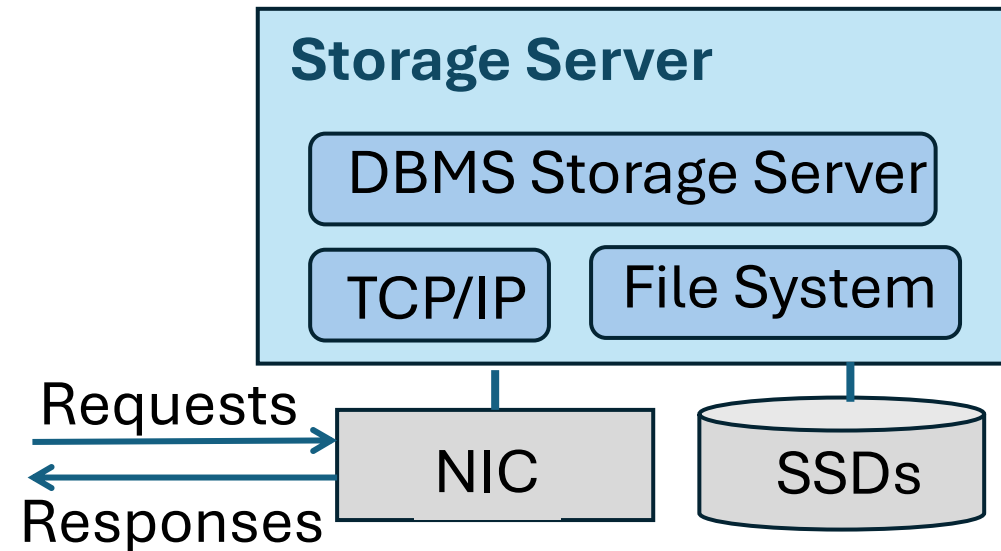
DPU Usage

- Network functions
- Hypervisor
- What about data management?

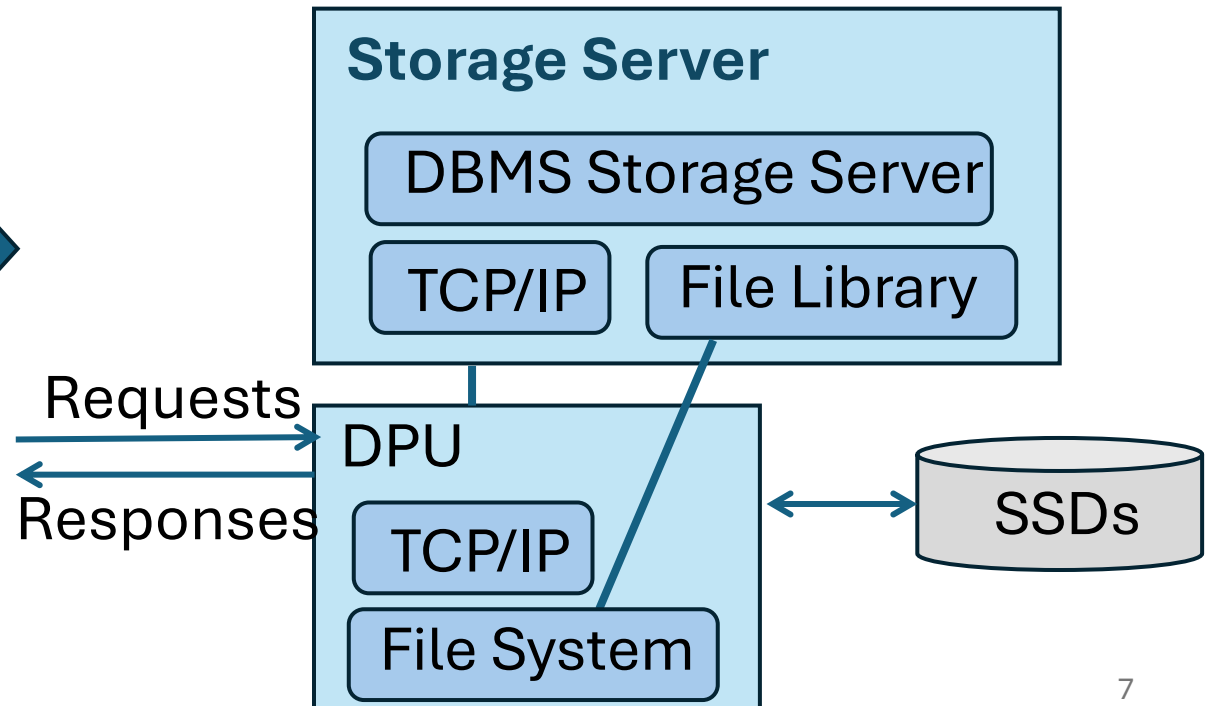
Usage Scenario 1: Page Server

- In a cloud-native DBMS, CPU can service GetPage requests
 - Saves host cores and reduces latency

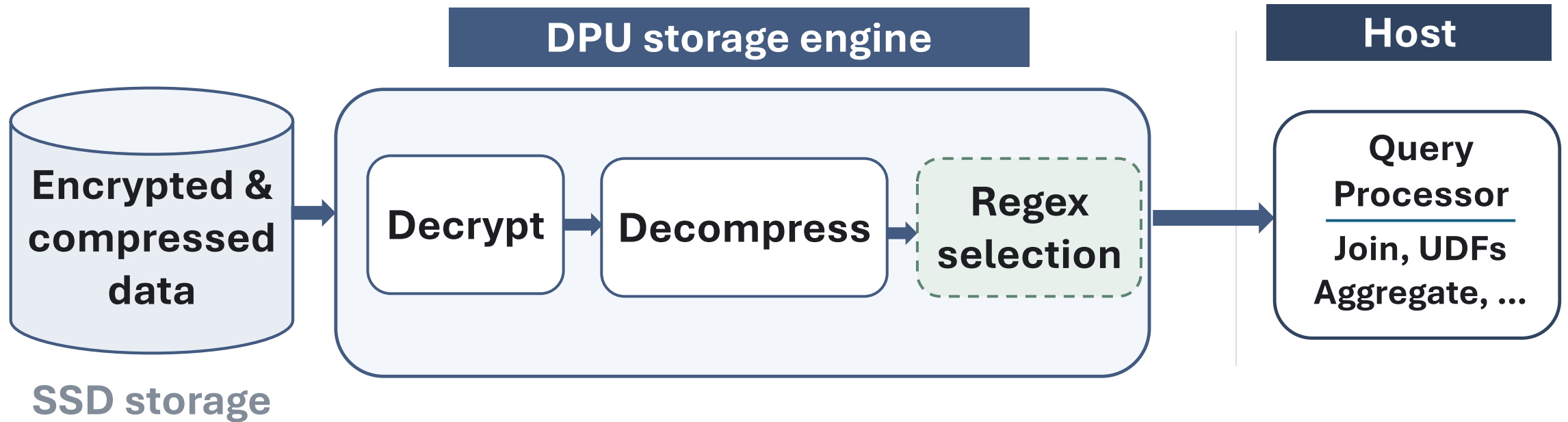
Today



Future

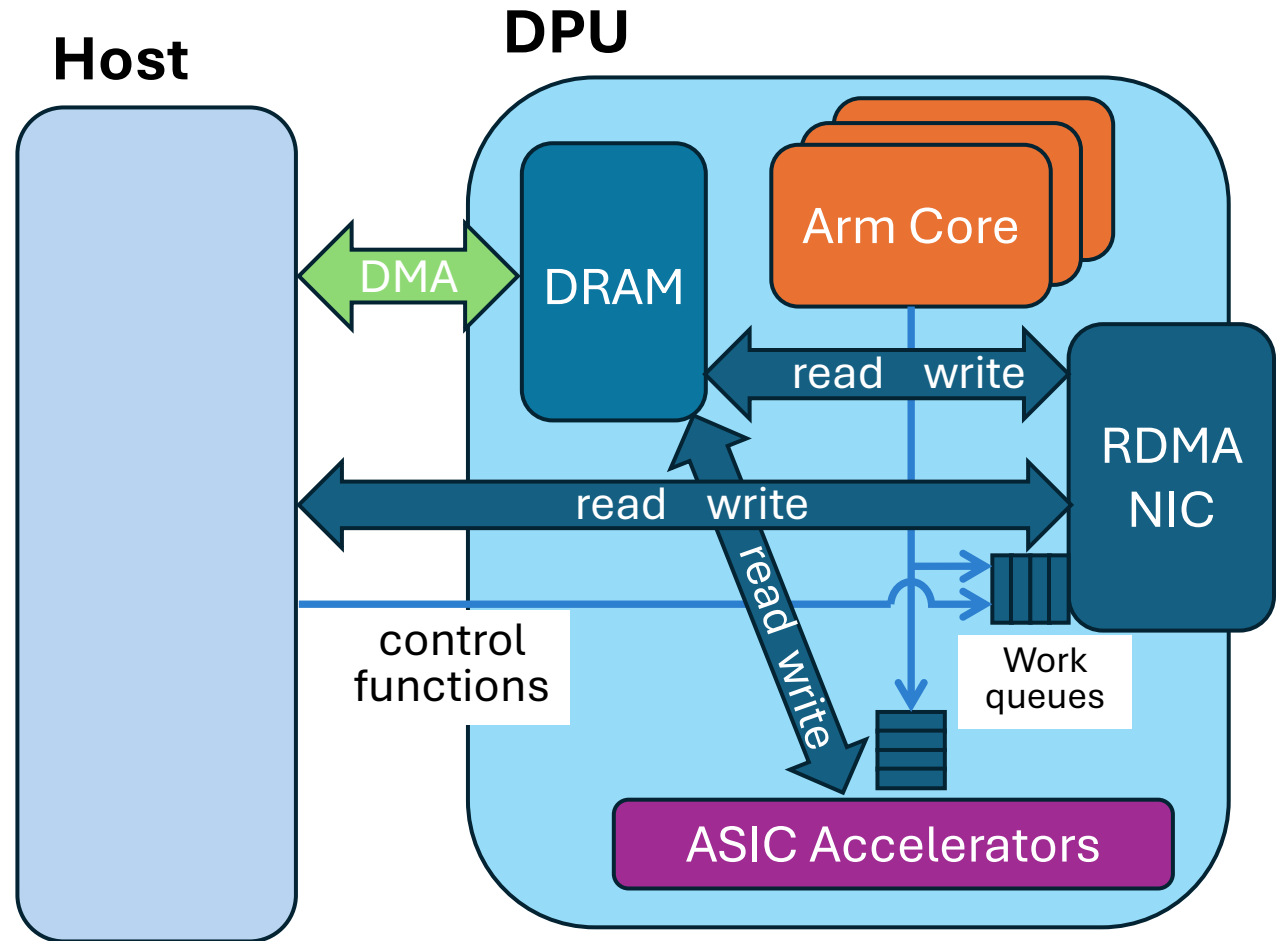


Usage Scenario 2: Cloud Analytics Service



Challenge 1 – How to Program a DPU

- Abstract the components into a *flexible* programming framework
- Vendor-specific SDKs are for packet processing
 - Too low-level

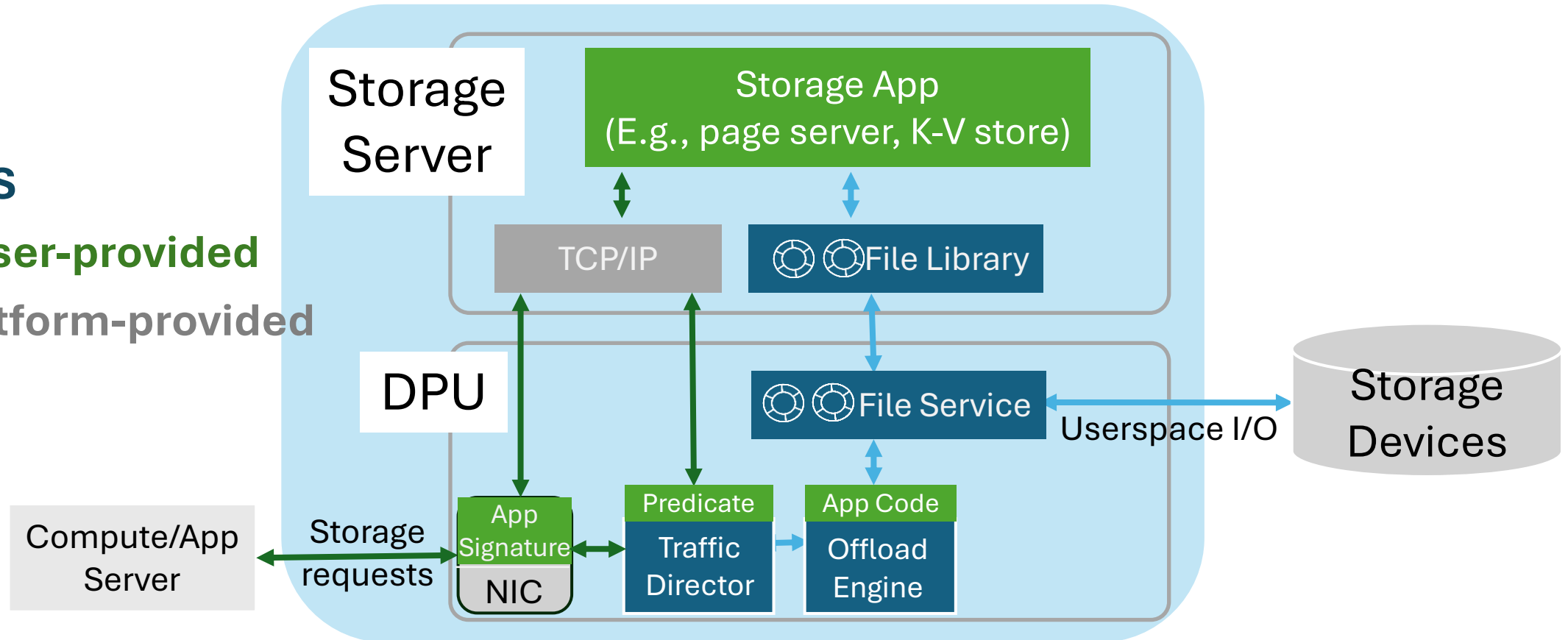


DPU-optimized Disaggregated Storage (DDS)

Blue: DDS

Green: User-provided

Gray: Platform-provided

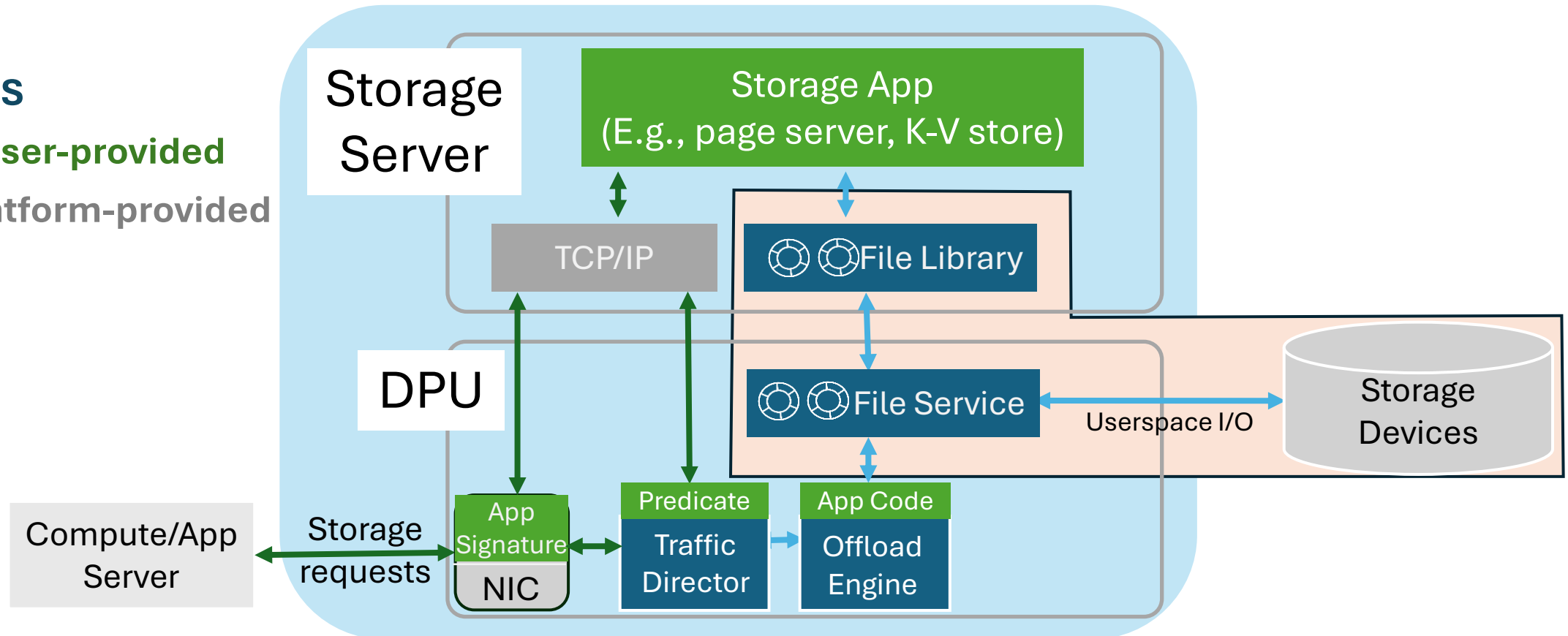


DDS Architecture - Storage

Blue: DDS

Green: User-provided

Gray: Platform-provided

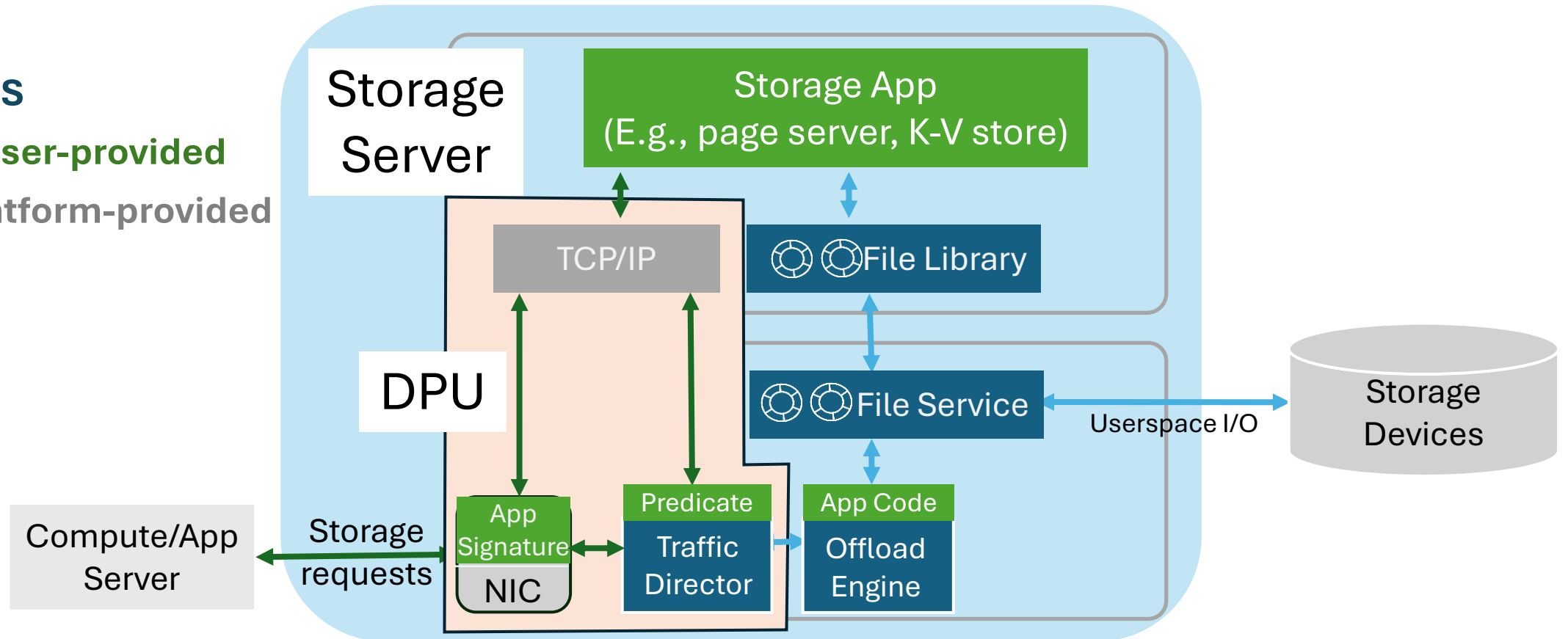


DDS Architecture - Network

Blue: DDS

Green: User-provided

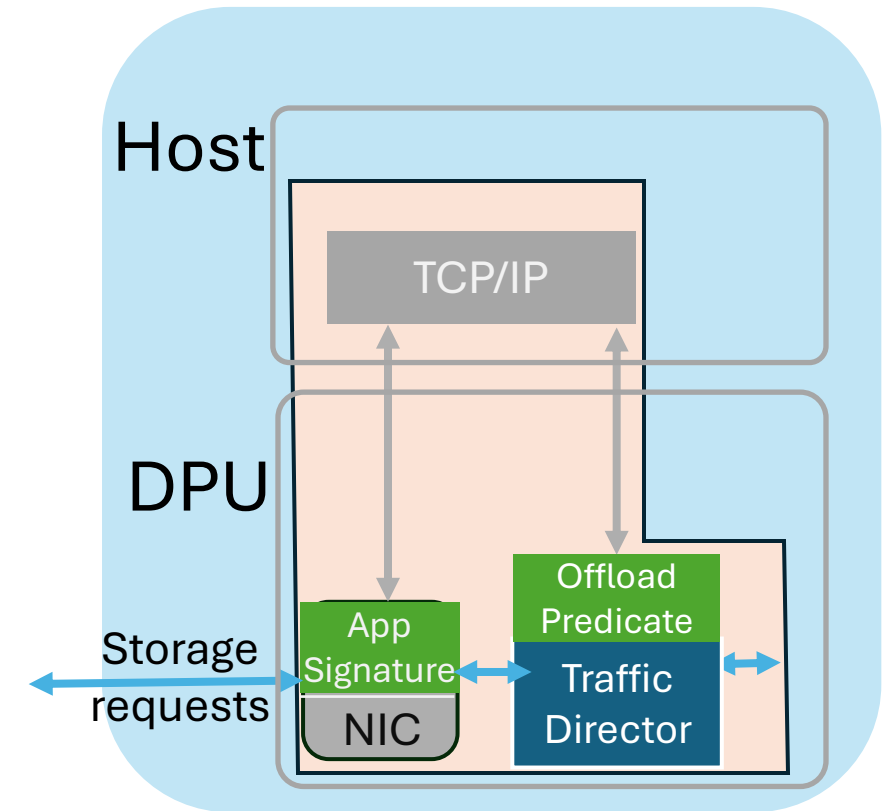
Gray: Platform-provided



DPU executes GetPage only if log replay isn't needed (i.e., SSD is fresh enough)

DDS Network – Request routing

- Application signature in packet header tells NIC if a packet belongs to database service
- Traffic Director's offload predicate tells if DPU can execute the request
- Must split TCP session into 2 sessions:
Client → DPU → Host

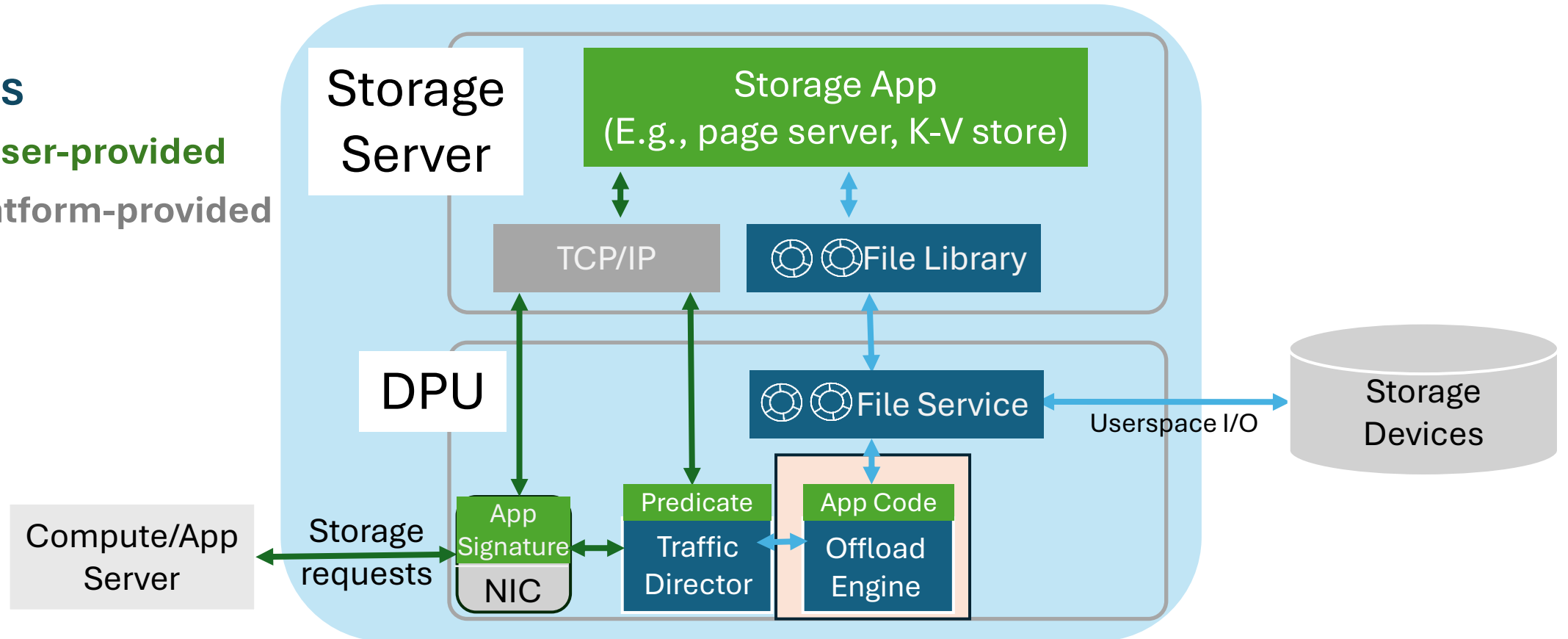


DDS Architecture - Application

Blue: DDS

Green: User-provided

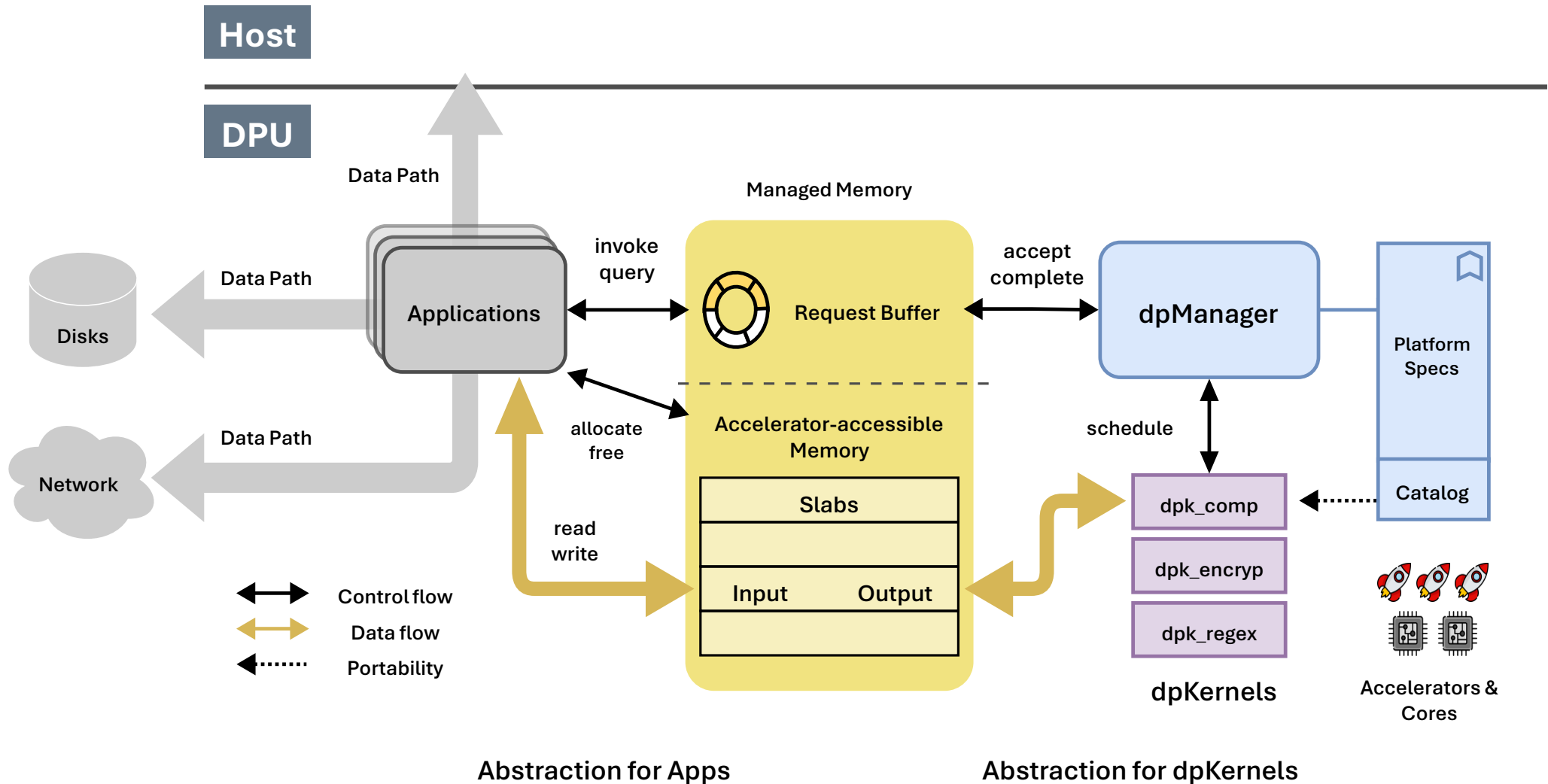
Gray: Platform-provided



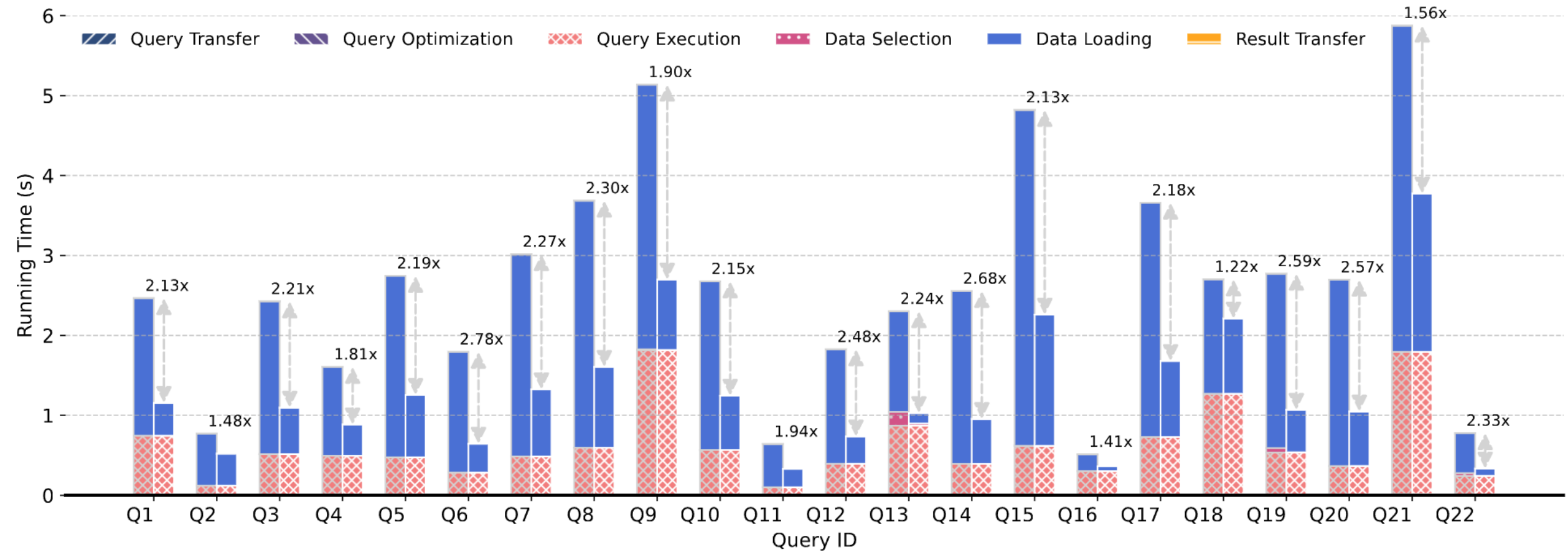
Challenge 2: Portability across DPU's

- How to avoid re-writing software for each type of DPU?
- A two-level computing framework
- dpKernels - higher level is a declarative interface for DPU functions
 - Vendor-agnostic async API: **ctx** $\leftarrow$ **invoke**(**dpk_name**, **in**, **out**, **args**)
 - Fixed semantics, parameterized by data schema
- dpManager - An implementation per device type
 - Platform spec – maps each dpKernel to DPU-specific SDK
 - Catalog - stores profiles of accelerators and execution strategies
 - A backup software implementation runs when there's no accelerator support

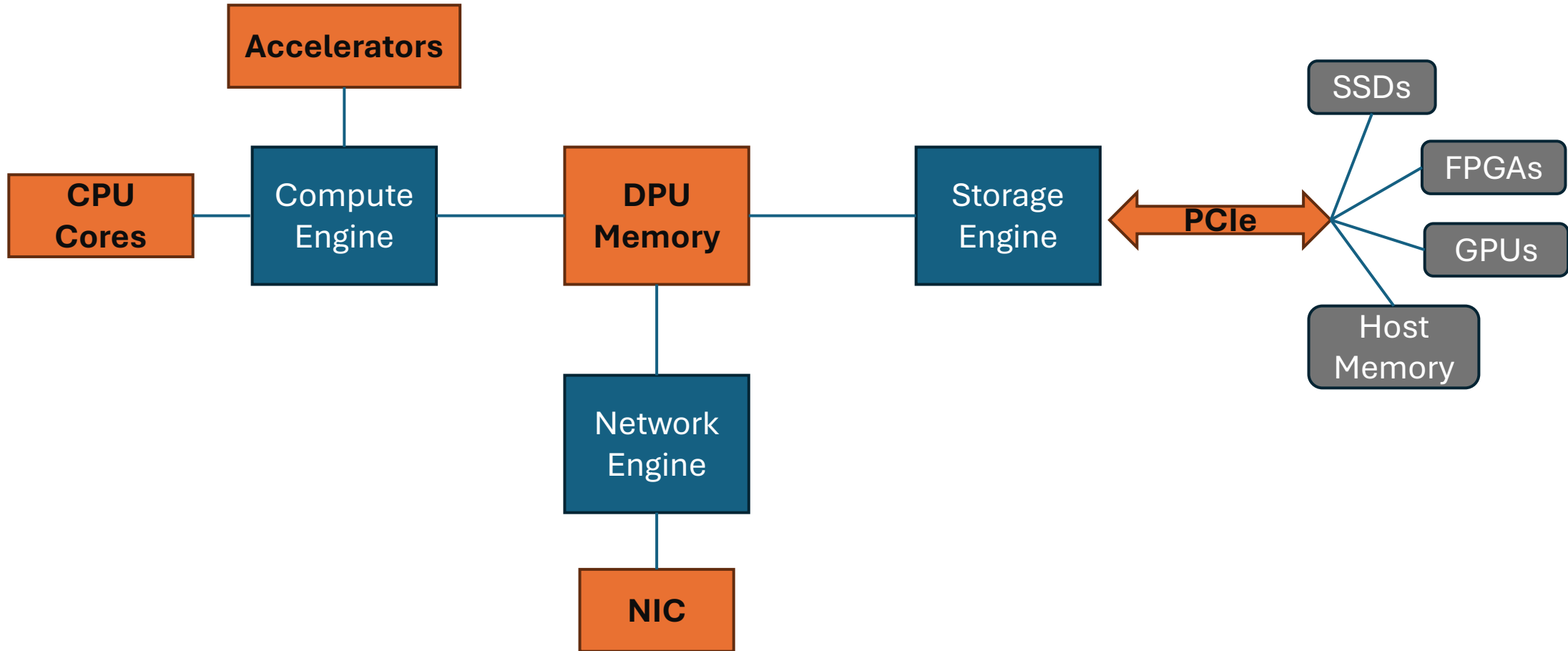
dpKernel Architecture



Data Loading Accelerated



The DPU Software Platform



Conclusion

- References
 - Qizhen Zhang, Philip A. Bernstein, Badrish Chandramouli, Jason Hu, Yiming Zheng: **DDS: DPU-optimized Disaggregated Storage**. *Proc. VLDB Endow.* 17(11): 3304-3317 (2024)
 - Jiasheng Hu, Kaiwen Zheng, Anna Li, Sidharth Sankhe, Philip A. Bernstein, Qizhen Zhang: **dpKernels: Harvesting DPU Compute Resources for Data-path Efficiency in Cloud Data Processing**, *Proc. VLDB Endow.* 19(10): 2563-2576 (2026)
- Coming Soon
 - **Making Sense of DPUPerformance for Cloud Data Processing** [SoCC 2026]
 - Integrating dpKernels and DDS