SNIA

FMS

the Future of Memory and Storage

Computational Memory

An Evolution of Computational Storage

Jason Molgaard

What is Computational Memory?

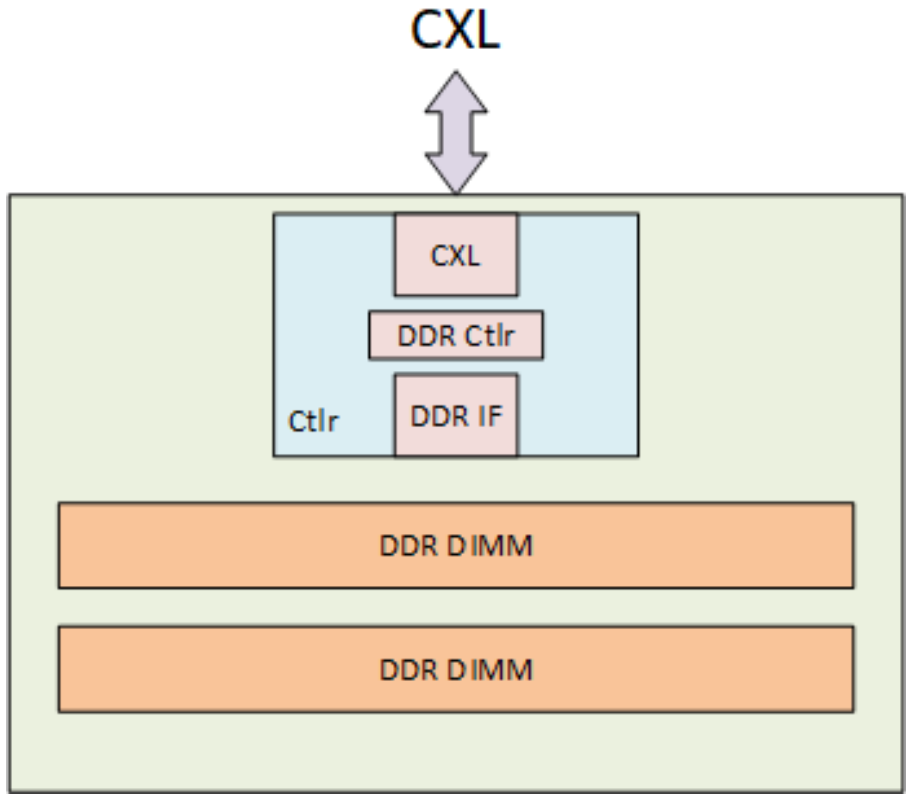
- Great Question!
- Let's start by considering where memory is heading
  - Memory and Storage are converging!

Drivers for Convergence of Memory & Storage

- CXL
  - New paradigms for memory subsystem architecture
  - Enables both memory and accelerators on the same device
  - Multiple use cases beyond simple memory
- Persistent Memory
  - Memory becomes storage
  - Specific storage IOPs no longer required
  - Still strong interest despite one technology ending

CXL Memory Device

- Key Features:
  - CXL Interface
  - Controller that processes CXL FLITs and converts to DDR protocol
  - DDR Controller
  - DDR DIMM(s) or Persistent Memory



Computational Storage Refresher

SNIA

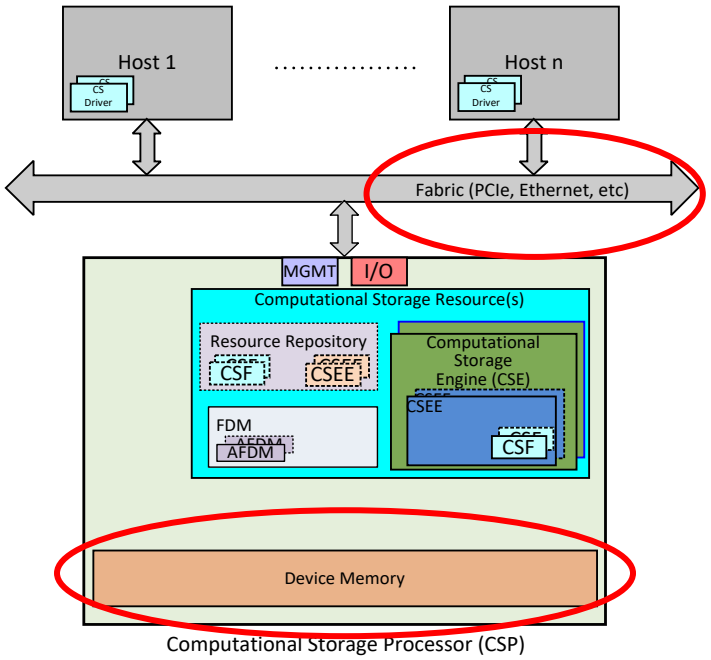
Computational Storage Refresher

Computation coupled to storage, offloading host processing or reducing data movement

Computational Storage Architecture - Extensible

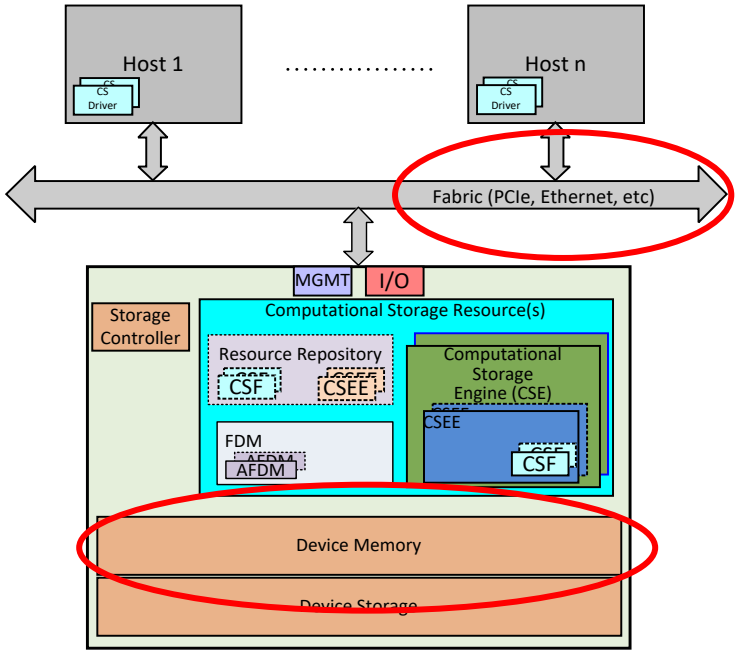
All 3 architectures are fabric agnostic and contain device memory

Computational Storage Processor



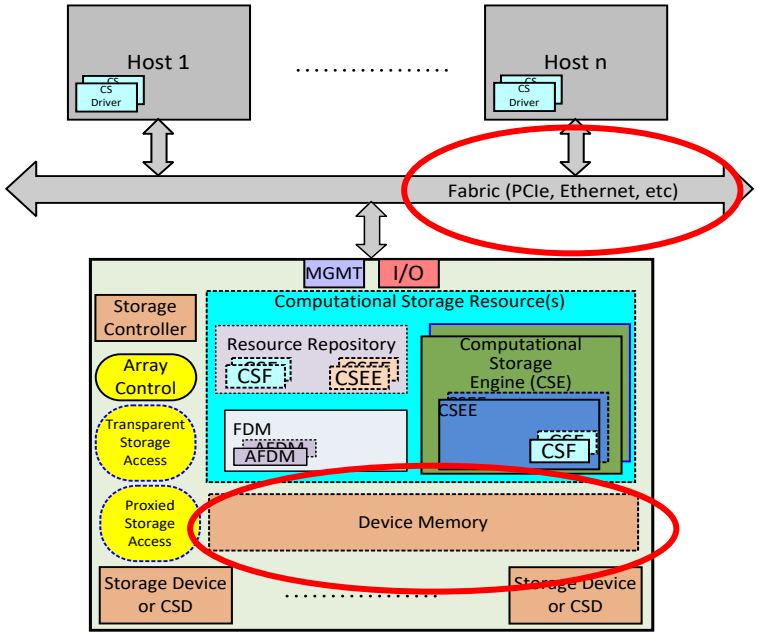
Computational Storage Processor (CSP)

Computational Storage Drive



Computational Storage Drive (CSD)

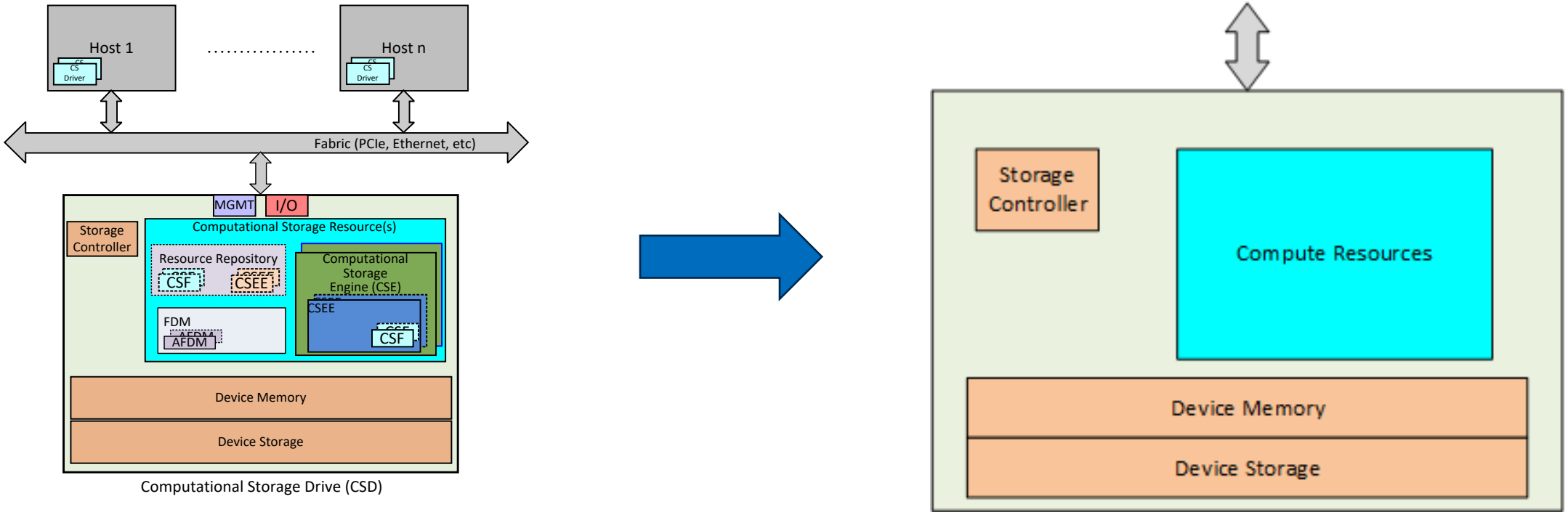
Computational Storage Array



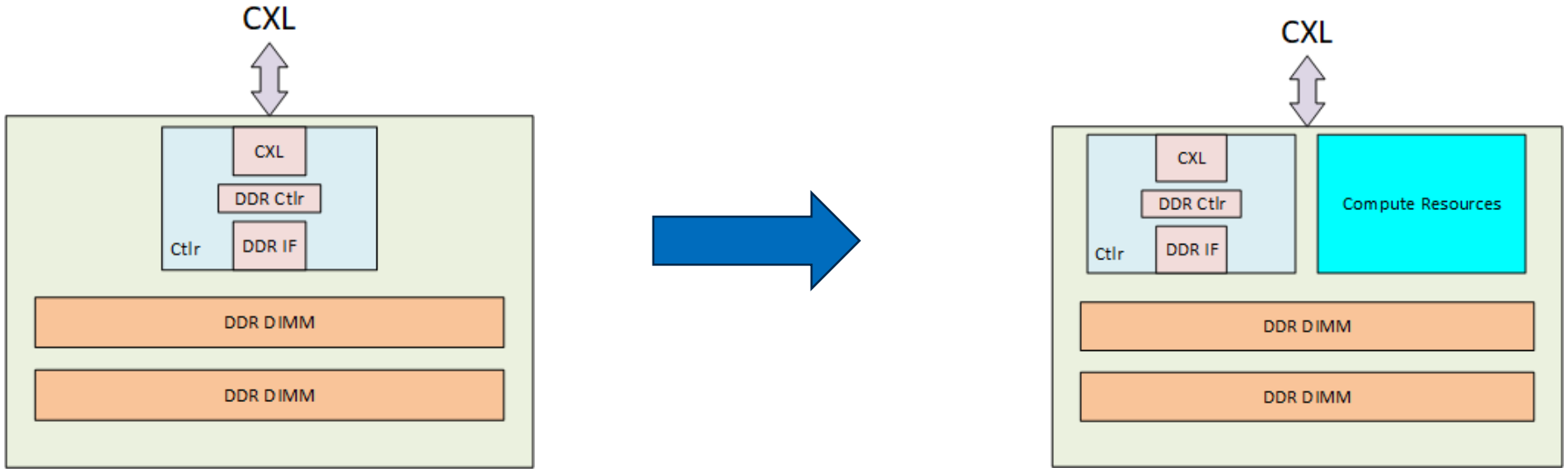
Computational Storage Array (CSA)

CSx = Computational Storage **Device** – CSP or CSD or CSA

High-Level CSD



Computational Memory

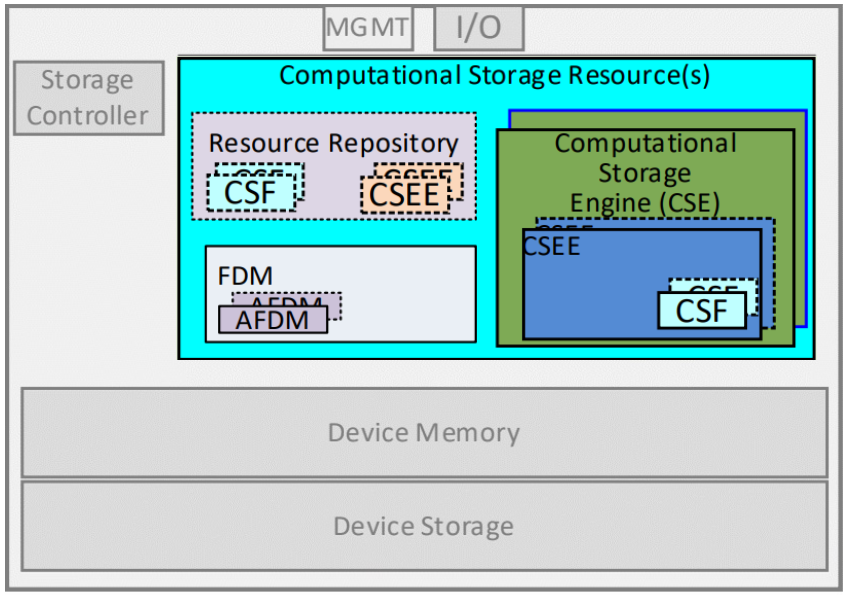


Computation coupled to memory, offloading host processing or reducing data movement

What Needs to Change for Computational Memory

- SNIA Computational Storage Architecture and Programming Model
  - Primarily Extensions of the existing architecture
    - Define Computational Memory related Terminology
    - Add Computational Memory diagrams
    - Extend examples for Computational Memory
- SNIA Computational Storage API
  - Not Much needs to change
    - Architecture should be largely unchanged by adding Computational Memory
    - Terminology updates to match the Architecture and Programming Model
- SNIA Persistent Memory Hackathon
  - All efforts apply to Computational Memory
  - Future Hackathons can test the compute resources and execute functions in the device

CSx Resources



Computational Storage Drive (CSD)

CSR - Computational Storage Resources are the resources available in a CSx necessary for that CSx to store and execute a CSF.

CSF - A Computational Storage Function is a set of specific operations that may be configured and executed by a CSE in a CSEE.

CSE - Computational Storage Engine is a CSR that is able to be programmed to provide one or more specific operation(s).

CSEE - A Computational Storage Engine Environment is an operating environment space for the CSE.

FDM - Function Data Memory is device memory that is available for CSFs to use for data that is used or generated as part of the operation of the CSF.

AFDM - Allocated Function Data Memory is a portion of FDM that is allocated for one or more specific instances of a CSF operation.

Resource Repository – Resources that are available but not activated

Possible Computational Memory Terminology

Rename CS Terms

- Computational Storage Resources → Compute Resources
- Computational Storage Function → Compute Function
- Computational Storage Engine → Compute Engine
- Computational Storage Engine Environment → Compute Engine Environment

Define Similar Terms

- Computational Memory Resources
- Computational Memory Function
- Computational Memory Engine
- Computational Memory Engine Environment

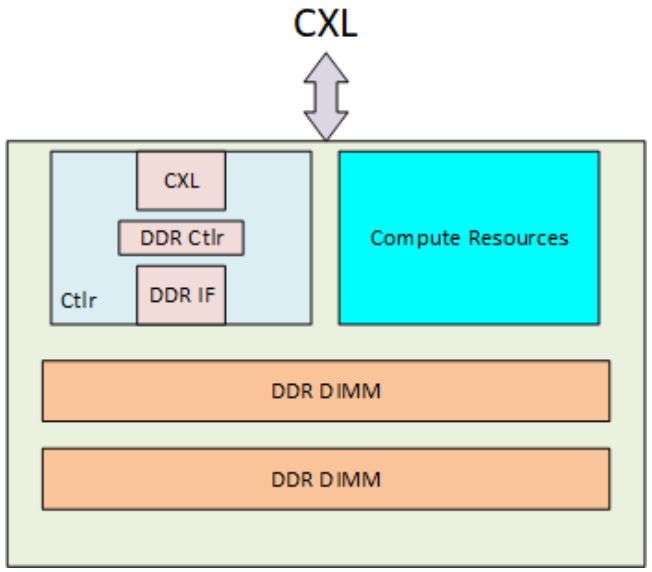
SNIA

I’m Interested – How do I get involved?

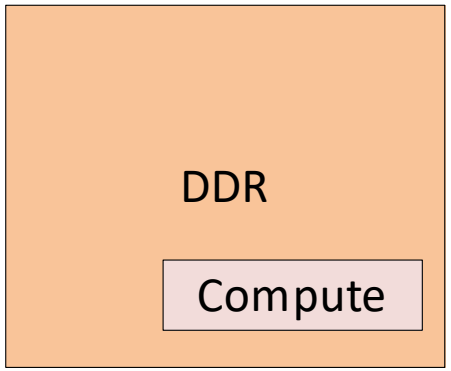
- Great Question!
  - We would be happy to have folks join us to define Computational Memory
  - 1) If you aren’t already a SNIA member, join SNIA
  - 2) If you aren’t already a member of the Computational Storage TWG, join the Computational Storage TWG
  - 3) Get Involved! – Help determine a new name for the TWG

Computational Memory vs In-Memory Compute

Computational Memory vs In-Memory Compute



Computational Memory



In-Memory Compute

Summary

- Computational Memory architecture is very similar to Computational Storage architecture
- SNIA CS Architecture and API can be readily updated to support Computational Memory
- Your participation will help standardize Computational Memory!

Thank You

SNIA