STORAGE DEVELOPER CONFERENCE

SDC 23

BY Developers FOR Developers

SNIA Computational Storage Standards

Bill Martin

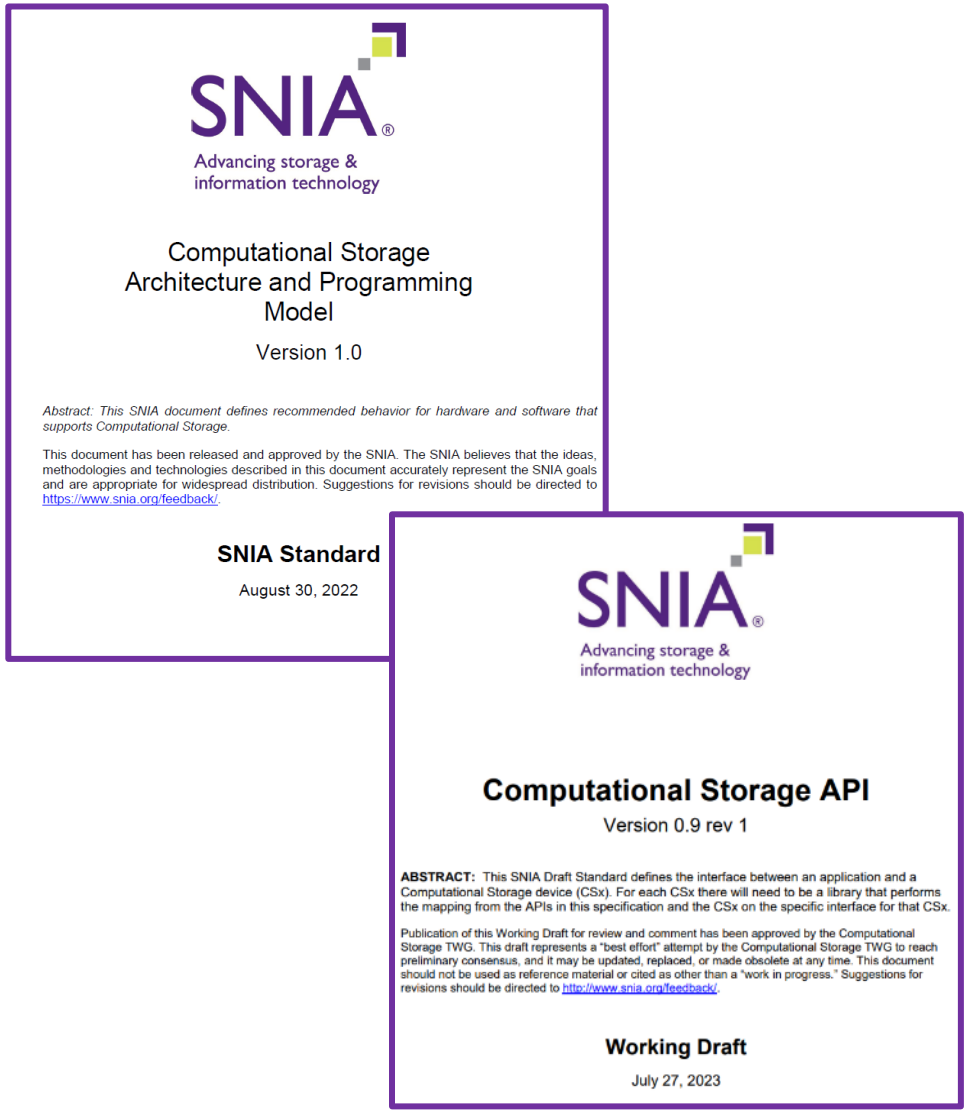
Jason Molgaard

Agenda

- Current status of SNIA Computational Storage Standardization
- Overview of SNIA CS Architecture
- Overview of SNIA CS API
- SNIA and NVMe™ Computational Storage
- CS and SDXI

Current Progress of TWG Output

- Architectural Document
  - v1.0 Released August 2022
    - Received the Most Innovative Memory Technology award at FMS 2022
  - v1.1 under development
    - Security enhancements for multiple tenants (complete)
    - Sequencing of Commands (in-progress)
- API
  - v0.8 public review version was available June 2022
  - v0.9 public review version available
    - In SNIA Membership vote towards v1.0

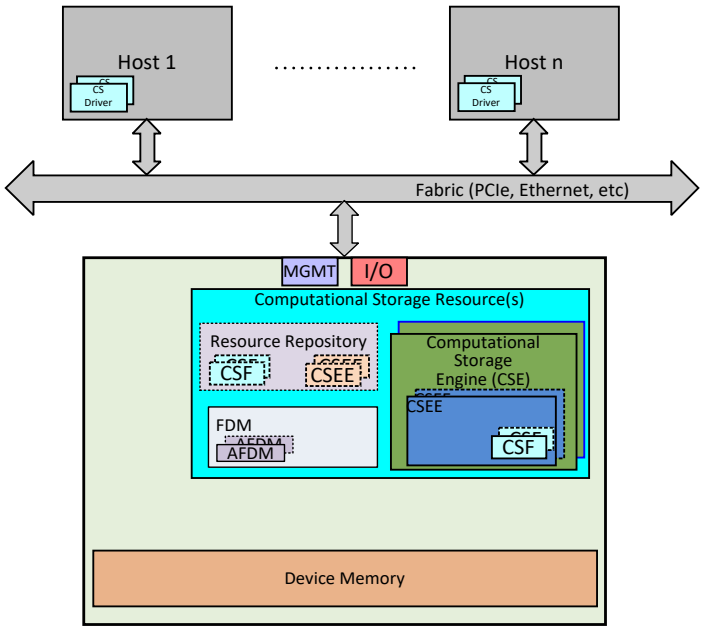


SDC 23

Architecture Overview

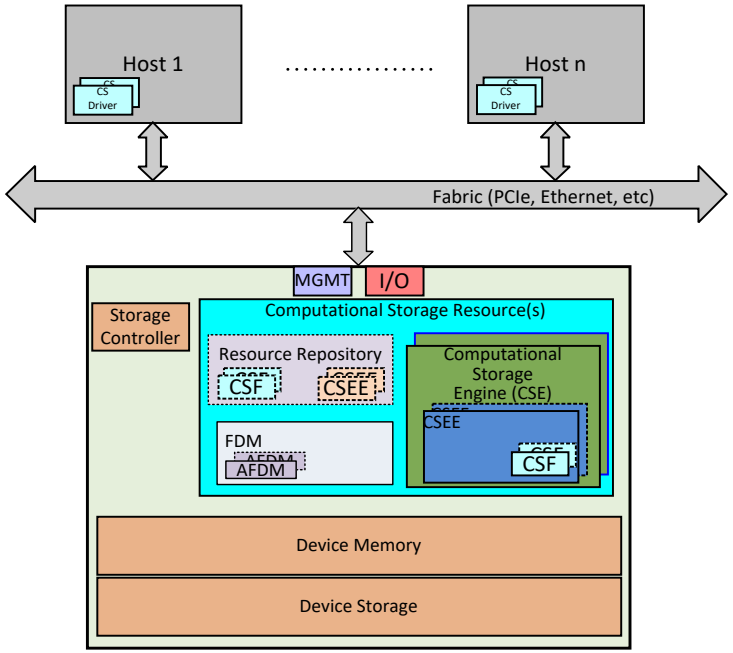
Computational Storage Architecture

Computational Storage Processor



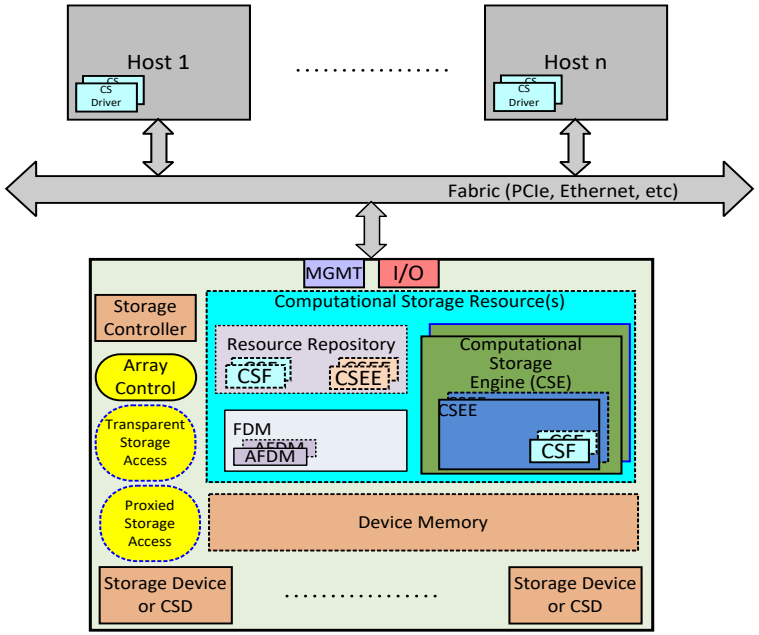
Computational Storage Processor (CSP)

Computational Storage Drive



Computational Storage Drive (CSD)

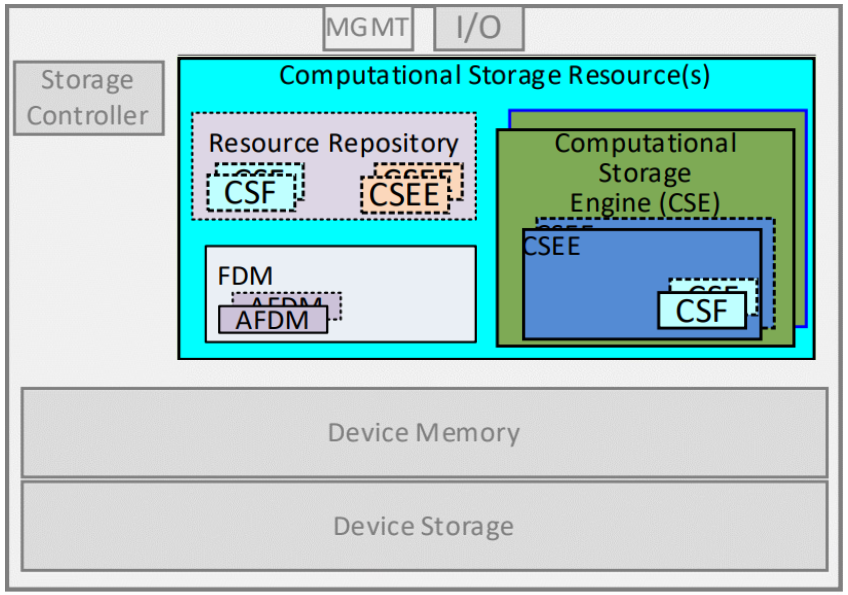
Computational Storage Array



Computational Storage Array (CSA)

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

A Deeper Dive of the 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.

Security Considerations for v1.0

- Assumptions
  - The environment consists of a single physical host or virtual host with one or more CSxes
  - The host is responsible for the security of the ecosystem that the CSxes operate within
  - CSx security requirements are comparable to the security requirements common to SSDs/HDDs
- Privileged Access
  - Elevated privileges necessary for operations

Security Considerations for v1.1

- Assumptions
  - The environment consists of multiple physical hosts or multiple virtual hosts with one or more CSxes
  - CSx security requirements are comparable to the security requirements common to SSDs/HDDs in multi-tenant environment
- Trust Relationships
- Elements required for a trust relationship are
  1. Identification
     - Exchanged between participating parties
  2. Authentication
     - Is done following identification
     - Exchange of authentication information is done with the same element as Identification
  3. Authorization
     - Is done following authentication
     - Authorizes specific actions on specific resources
     - May be done at a lower-level element than the element that was authenticated
  4. Access Control
     - Controls access to elements of the CSx that are within the scope of the authorization
     - May be access to a CSE, a CSEE, or a CSF
- Different elements of the trust relationship may be at different levels
  - Identification and Authentication may be at the CSx
  - Authorization may be at the CSEE within the CSx
  - Access Control may be at the CSF activated in the CSEE

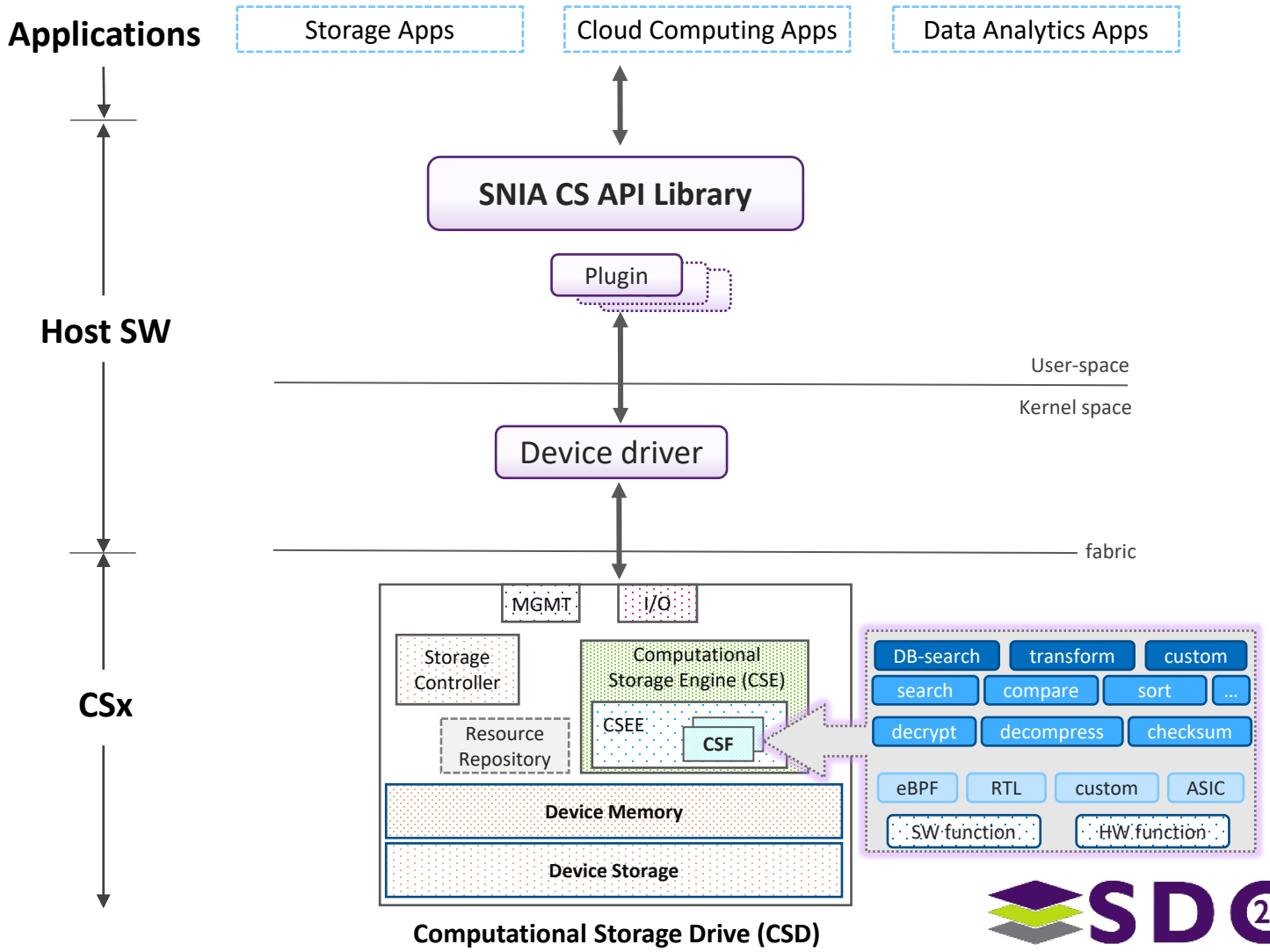
Sequencing of Commands

- Enables sequences of CSFs to execute in succession
  - Sequence executes in-order
  - Allows multiple CSFs to execute with minimal host involvement
- Aggregator CSF
  - Manages execution of the sequence
  - Tracks completion status of each CSF
  - May be downloaded or Pre-installed
  - Fixed Sequence or Variable Sequence defined by parameters passed by the host
- Error Handling
  - May be handled by the host or the aggregator CSF

API Overview

SNIA Computational Storage APIs

- One set of APIs for all CSx types
- APIs hide device details
  - Hardware, Connectivity
- Abstracts device details
  - Discovery
  - Access
  - Device Management
  - Memory Management
    - alloc/free/init
  - Storage/Memory Access
  - Download
  - Execute CSFs
- APIs are OS agnostic

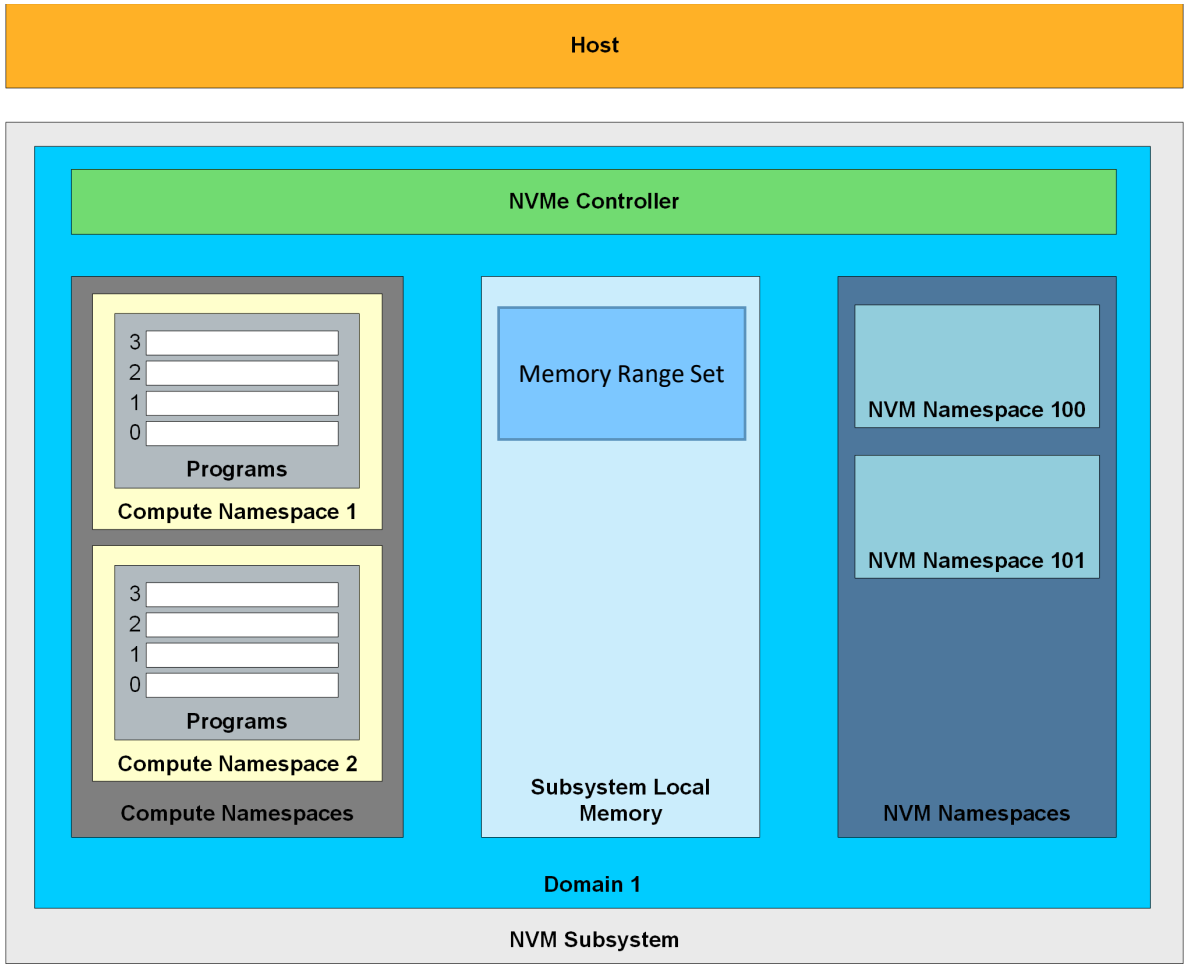


Computational Storage API

- For more information about the SNIA CS API, please attend:
  - “Programming Computational Storage with the SNIA API” by Oscar Pinto

SNIA and NVMe Computational Storage

NVMe Computational Storage Architectural Components



- Compute Namespaces
  - Compute Engines
  - Programs
- Programs operate on data in Subsystem Local Memory
  - Allocated as Memory Range Set
  - Includes program input, output
- NVM Namespaces
  - Persistent storage of data
  - NVM
  - ZNS
  - KV
- Data is transferred between NVM Namespaces and SLM using the Memory Copy command

This presentation discusses NVMe work in progress, which is subject to change without notice.

Correlation of SNIA/NVMe terms

SNIA Terms

- Computational Storage Engine
- Computational Storage Engine Environment
- Resource Repository
  - Downloaded CSF and CSEE
  - Pre-loaded CSF and CSEE
- Activation
- Function Data Memory (FDM)
- Allocated FDM (AFDM)
- Device Storage

NVMe Terms

- Compute Engine/Compute Namespace
- Virtual (Not currently defined)
- Programs
  - Downloaded programs
  - Device-defined programs
- Activation
- Subsystem Local Memory (SLM)
- Memory Range Set
- NVM Namespaces

NVMe Computational Storage

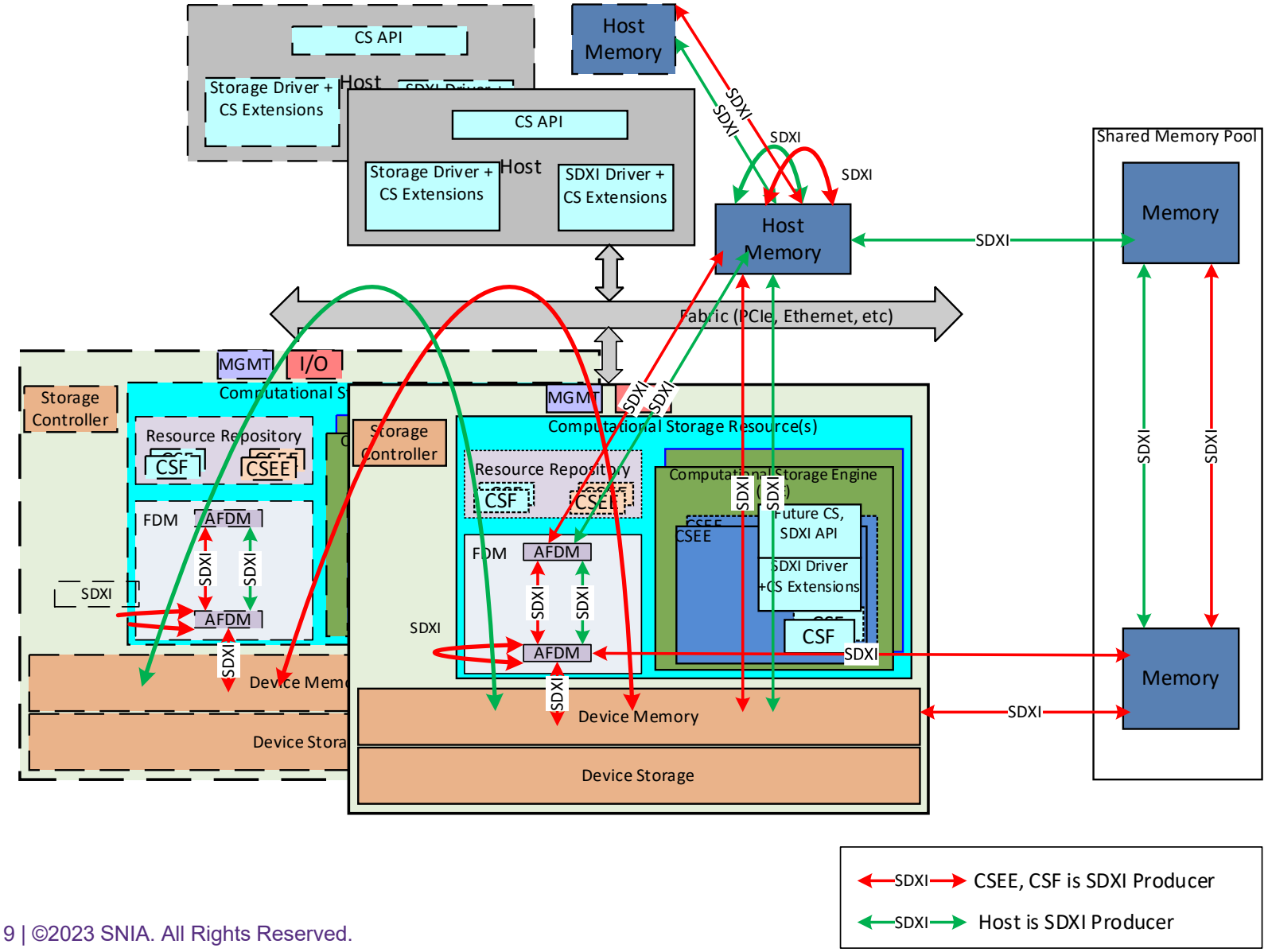
- For more information about NVMe Computational Storage, attend:
  - “NVMe Computational Storage Standards” by Kim Malone and Bill Martin

CS and SDXI Collaboration

SDXI (Smart Data Accelerator Interface)

- Smart Data Accelerator Interface (SDXI) is:
  - A SNIA standard for a memory to memory data movement and acceleration interface
  - Extensible
  - Forward-compatible
  - Independent of I/O interconnect technology
  - Provides data transformation features
- v1.0 was published November 2022
  - https://www.snia.org/sdxi

Combined SDXI+CS Architecture



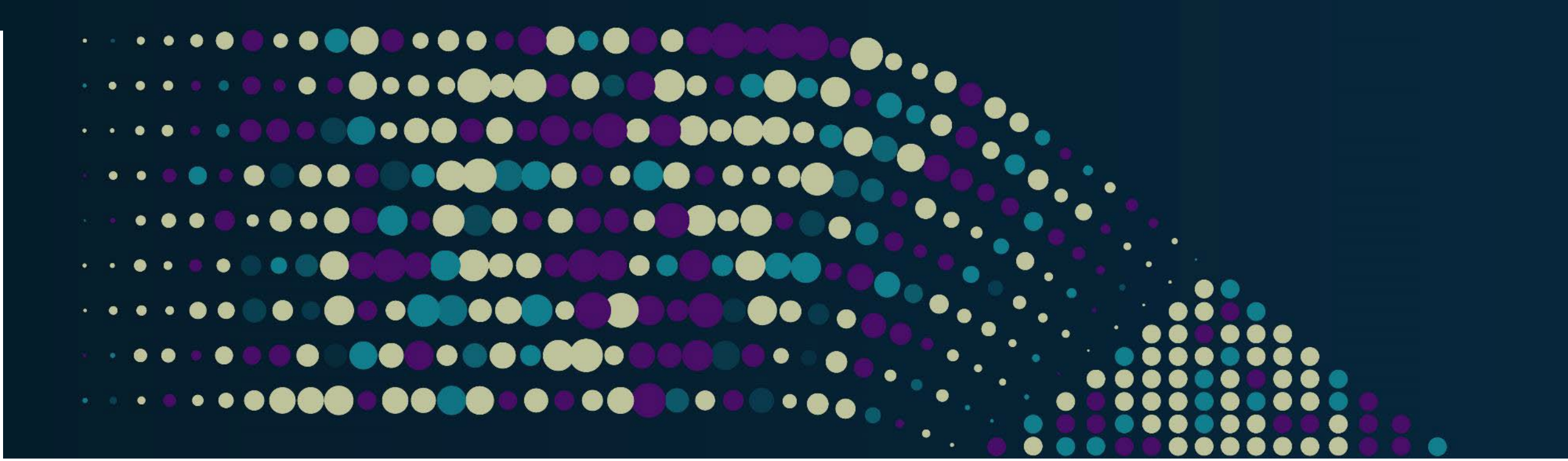
- SDXI used for data movement with Computational Storage used for compute
- Multiple SDXI producers in a CS Architecture
- SDXI enables data movement across multiple AFDM regions

CS + SDXI

- For more information about CS + SDXI, please attend:
  - “Envisioning a Computational Storage Architecture with an SDXI Data Mover: Early Efforts” by Shyam Iyer and Jason Molgaard
- For more information about SDXI, please attend:
  - “SNIA SDXI Specification v1.0 and Beyond” by Shyam Iyer

Interested? Join Us!

- Join SNIA: https://www.snia.org/member_com/join-SNIA
- Join the Computational Storage TWG: https://members.snia.org/workgroup/index



Please take a moment to rate this session.

Your feedback is important to us.

