SNIA DEVELOPER CONFERENCE

SDC 24

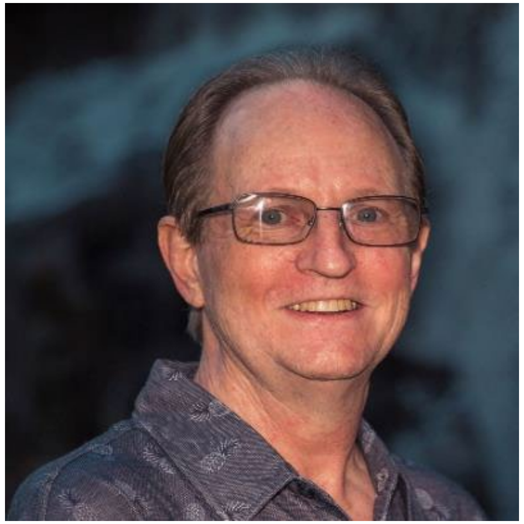
BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

SNIA Computational Storage Standards

Bill Martin and Jason Molgaard

Speakers



Bill Martin

Jason Molgaard

SAMSUNG

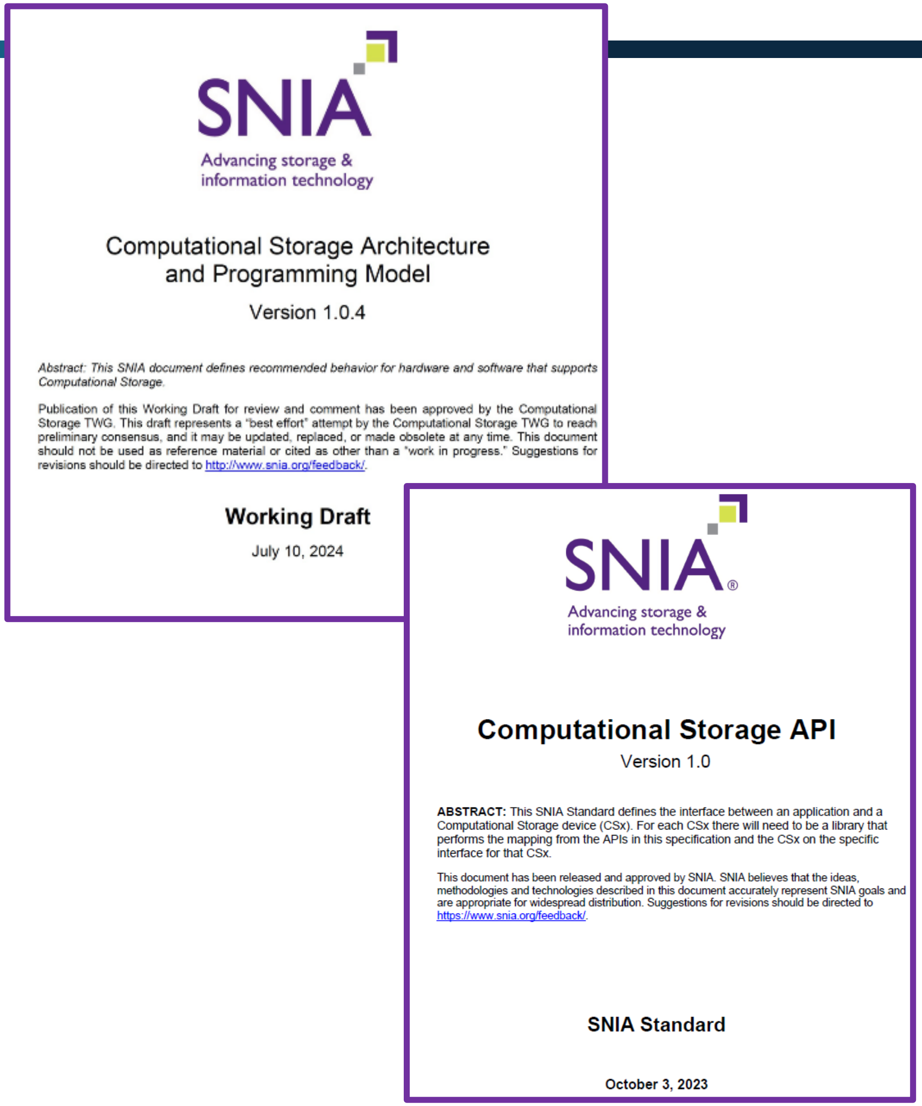
SOLIDIGM

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

- Architectual Document
  - v1.0 Released August 2022
    - Received the Most Innovative Memory Technology award at FMS 2022
  - v1.0.4 Public Review available
    - Sequencing of Commands
    - Security enhancements for multiple tenants
- API
  - v1.0 was released in October 2023
    - Received the Most Innovative Memory Technology award at FMS 2023
  - v1.1 under development



SNIA
Advancing storage & information technology

Computational Storage Architecture and Programming Model

Version 1.0.4

Abstract: This SNIA document defines recommended behavior for hardware and software that supports Computational Storage.

Publication of this Working Draft for review and comment has been approved by the Computational Storage TWG. This draft represents a "best effort" attempt by the Computational Storage TWG to reach preliminary consensus, and it may be updated, replaced, or made obsolete at any time. This document should not be used as reference material or cited as other than a "work in progress." Suggestions for revisions should be directed to http://www.snia.org/feedback/.

Working Draft

July 10, 2024

SNIA®
Advancing storage & information technology

Computational Storage API

Version 1.0

ABSTRACT: This SNIA Standard defines the interface between an application and a Computational Storage device (CSx). For each CSx there will need to be a library that performs the mapping from the APIs in this specification and the CSx on the specific interface for that CSx.

This document has been released and approved by SNIA. SNIA believes that the ideas, methodologies and technologies described in this document accurately represent SNIA goals and are appropriate for widespread distribution. Suggestions for revisions should be directed to https://www.snia.org/feedback/.

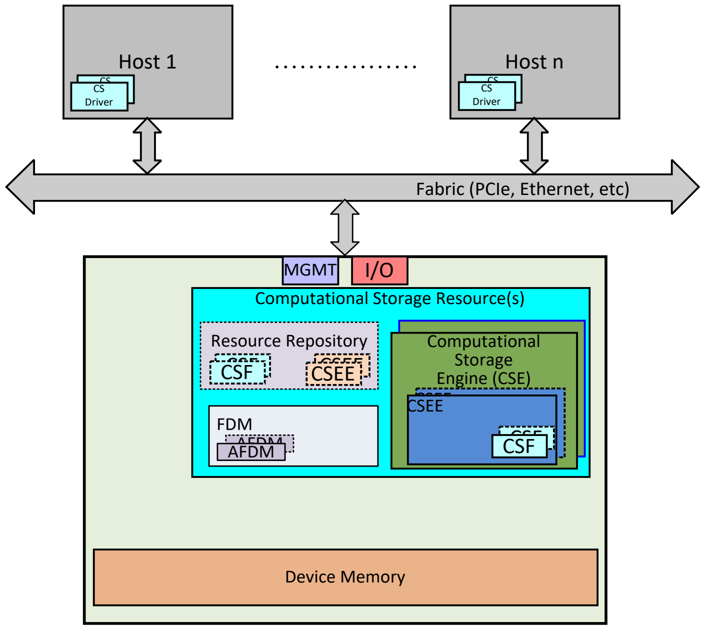
SNIA Standard

October 3, 2023

Architecture Overview

Computational Storage Architecture

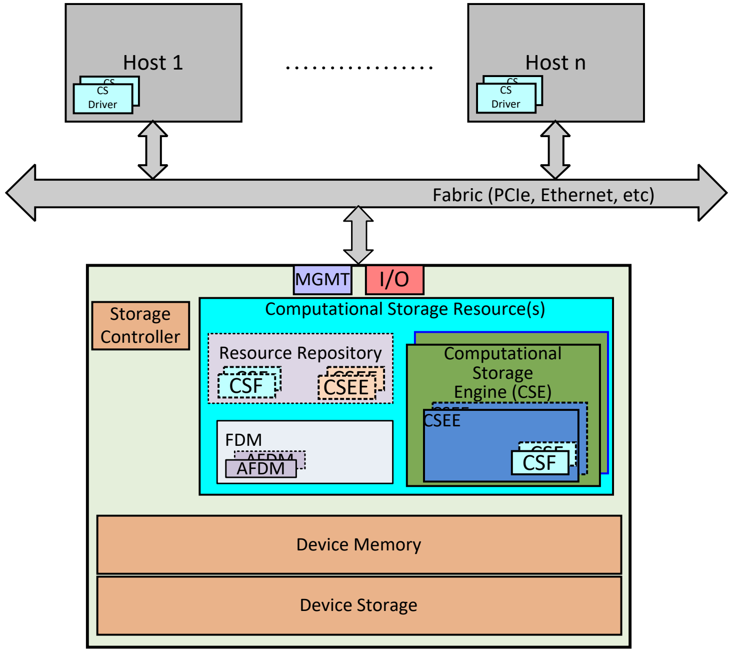
Computational Storage Processor



Host 1
CS Driver
Host n
CS Driver
Fabric (PCIe, Ethernet, etc)
MGMT
I/O
Computational Storage Resource(s)
Resource Repository
CSF
CSEE
Computational Storage Engine (CSE)
CSEE
CSF
FDM
AFDM
Device Memory

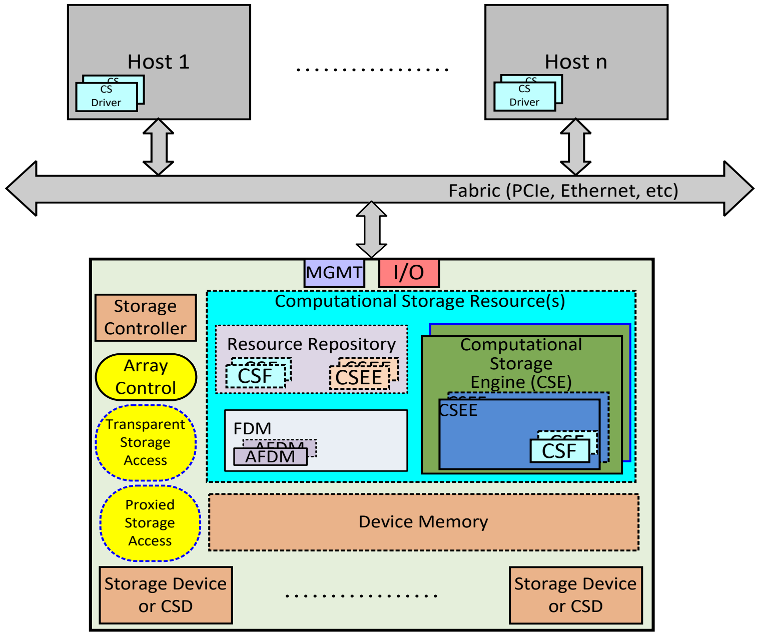
Computational Storage Processor (CSP)

Computational Storage Drive



Computational Storage Drive (CSD)

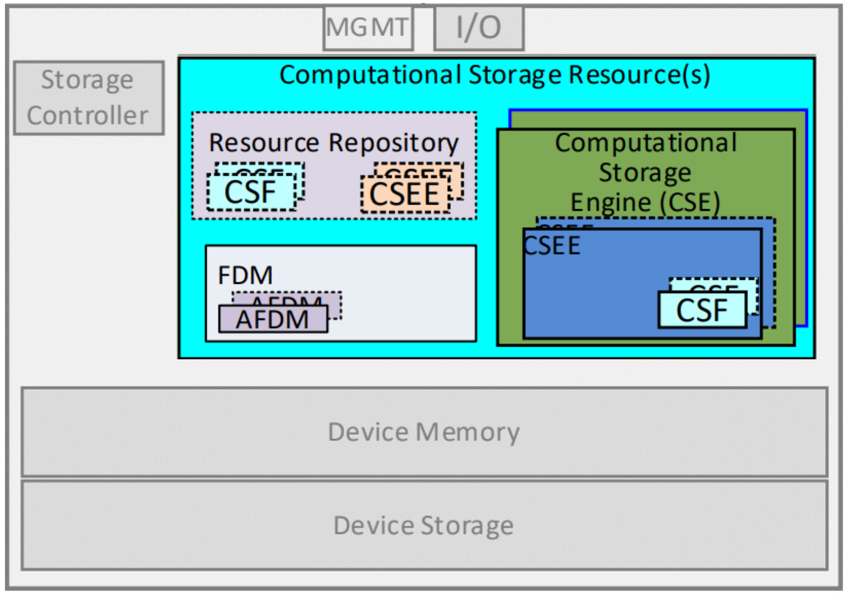
Computational Storage Array



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
- **Resource Repository –** Resources that are available but not activated

Sequencing of Commands

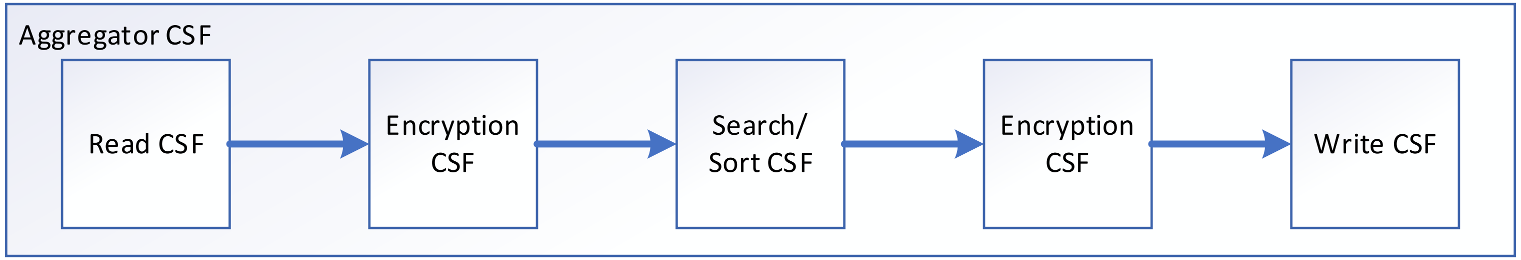
- Enables sequences of CSFs to execute in succession
    - Sequence executes in-order
    - Allows multiple CSFs to execute with minimal host involvement

- 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

Sequencing Examples

Fixed Sequence

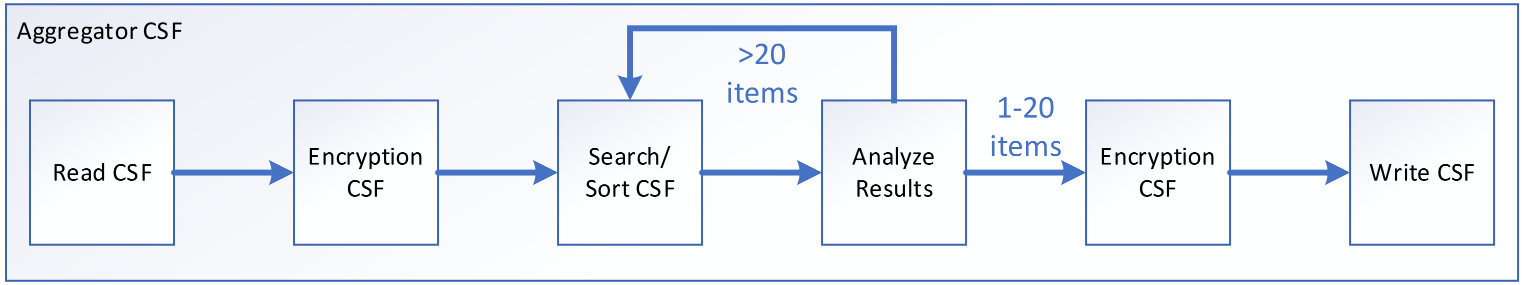


Read Data → Decrypt Data → Search & Sort Create Output List → Encrypt Output List → Store List

Considerations:

- What if the data read is already decrypted (no decryption required?)
- What if the decryption fails?

Variable Sequence



Read Data → Decrypt Data → Search & Sort Create Output List → Analyze results → Encrypt Output List → Store List

Variable Sequence Aggregators support recursive calls to itself.

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

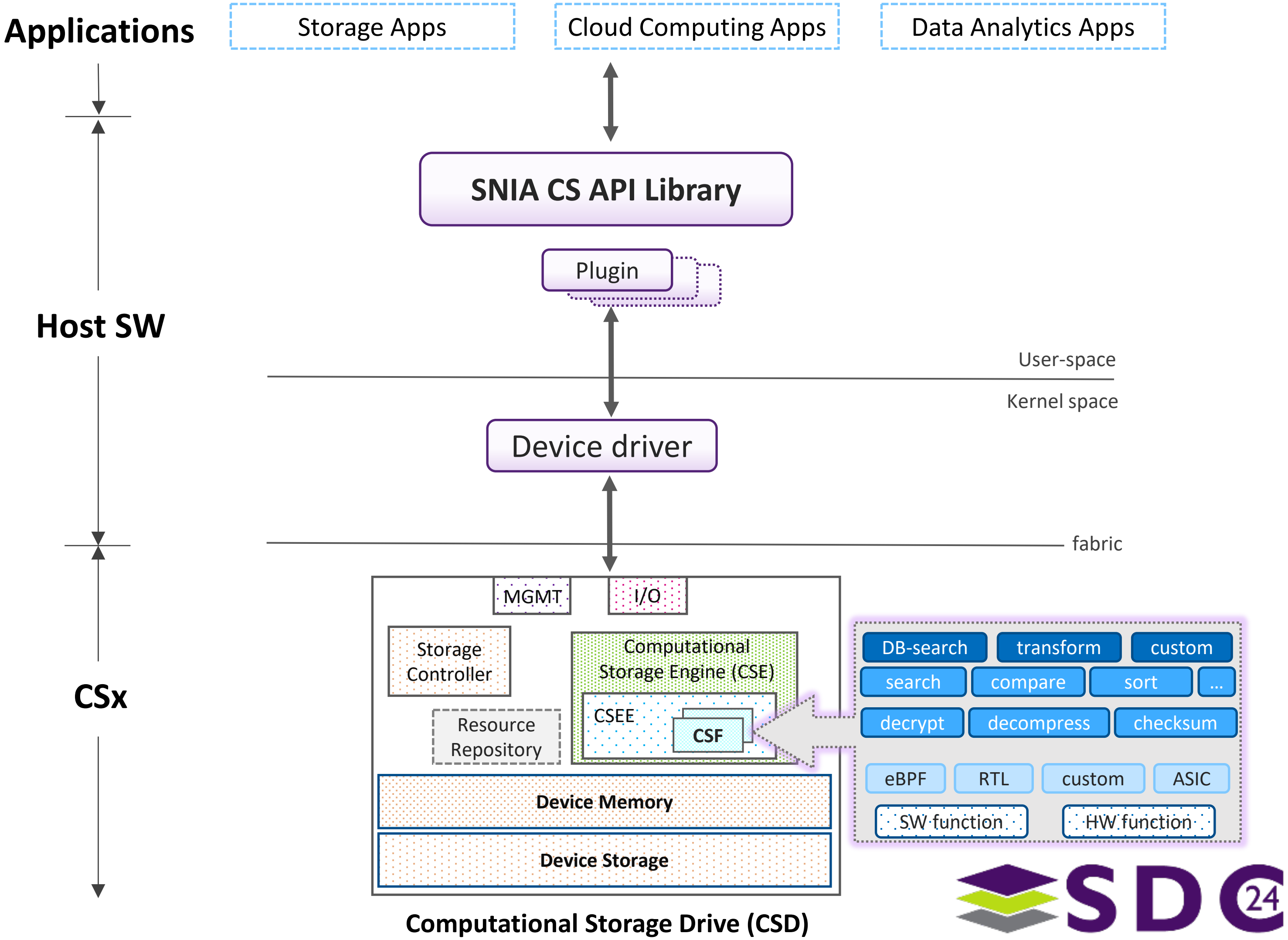
Security Considerations for v1.1

- 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 a 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

API Overview

SNIA Computational Storage APIs

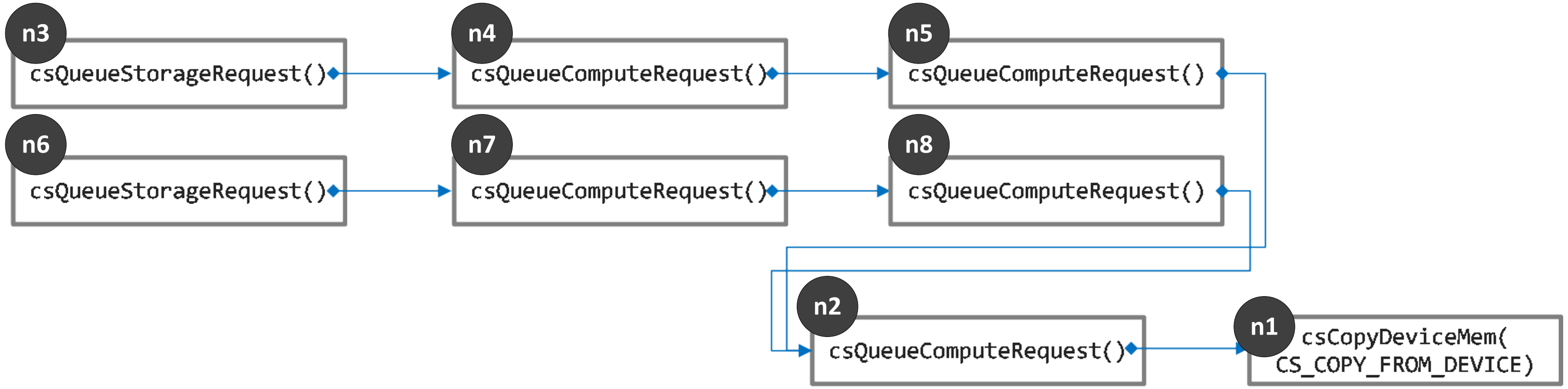
- One set of APIs for all CSx types
- APIs hide device details
  - Hardware, Connectivity
- Abstracts device details
  - Discovery
  - Access
  - Device Management
  - Memory Management
    - alloc/free/init
  - Storage/Memory Access
  - Download
  - Execute CSFs
- APIs are OS agnostic



Updates to CS API v1.1

- Clarified batch process
  - Changed the functions that create and modify batch requests
    - CS_STATUS csConfigureBatchEntry(CS_BATCH_HANDLE BatchHandle, CS_BATCH_CONFIG_TYPE Action, const CsBatchRequest *Req, CS_BATCH_INDEX Before, CS_BATCH_INDEX After, CS_BATCH_INDEX *Curr)
      - Actions: ADD, DELETE, RECONFIG, JOIN, SPLIT
  - Updated flow diagrams for creating and updating batch requests
- Clarifying error return values
- Additional editorial clean-up from v1.0

Creating Batch Request



csConfigureBatchEntry() Function Parameters

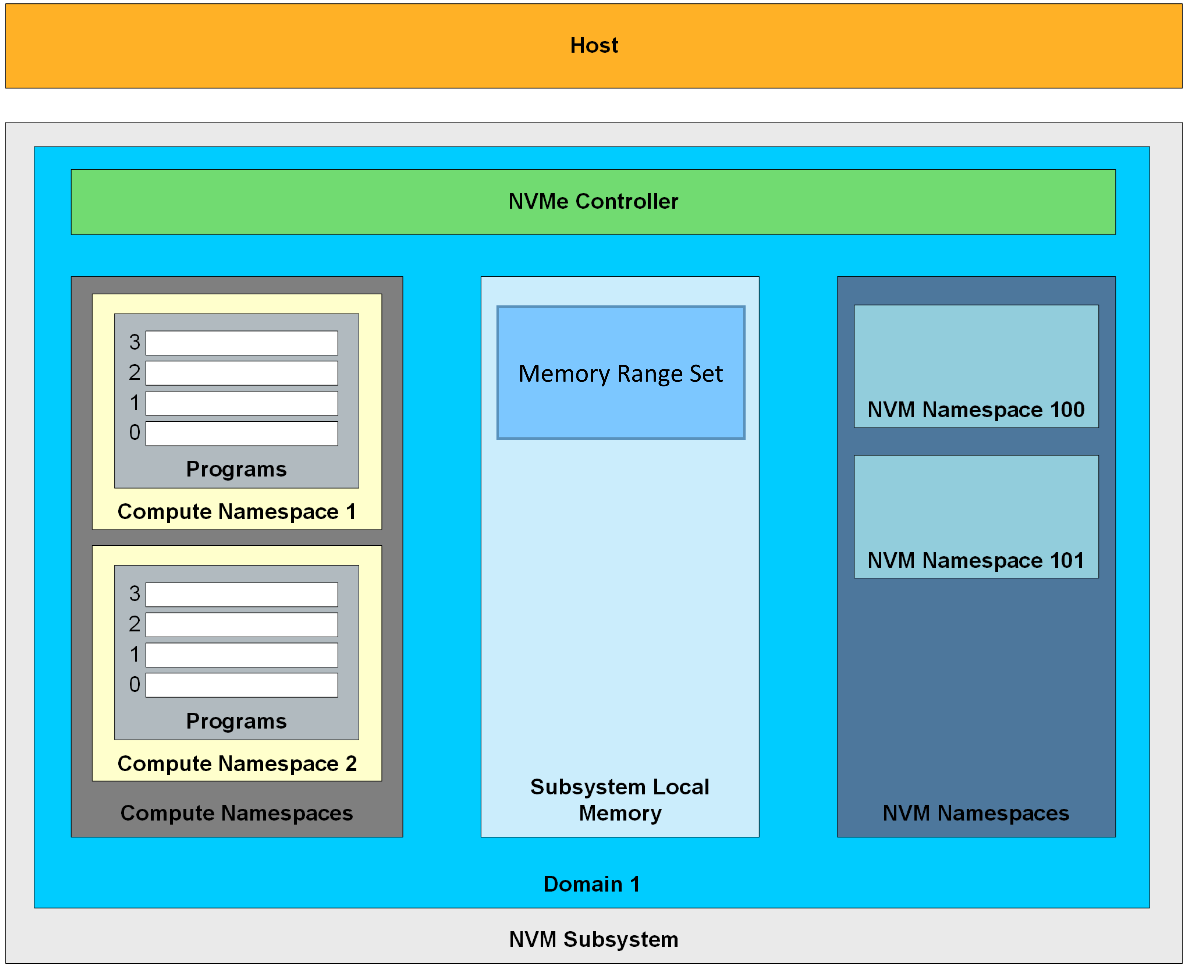
Before	After	Current	Comments
0	0	n1	New entry not associated with any others.
n1	0	n2	New entry before n1.
0	0	n3	New entry not associated with any others.
0	n3	n4	New entry after n3.
n2	n4	n5	New entry after n4 and before n2. This entry links the two sequences.
0	0	n6	New entry not associated with any others.
0	n6	n7	New entry after n6.
n2	n7	n8	New entry after n7 and before n2. This entry links the two sequences and creates the requirement that n5 and n8 complete prior to execution of n2.

SNIA and NVMe Computational Storage

NVMe Computational Storage and SNIA Architecture

- NVMe Computational Storage ratified January 2024
- NVMe Computational Storage implements the SNIA Computational Storage Model
- SNIA API supports 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

Correlation of SNIA/NVMe terms

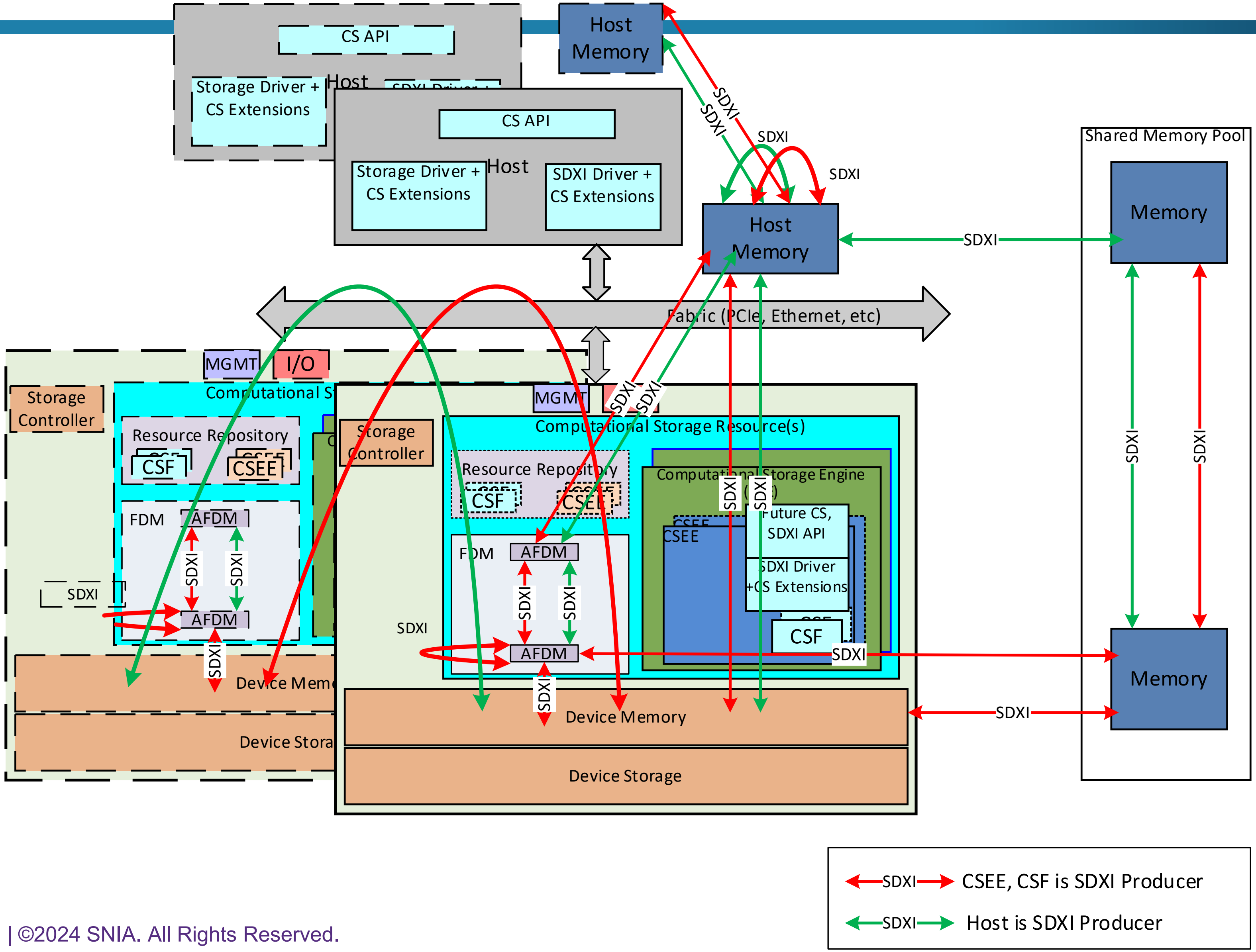
SNIA Terms	NVMe Terms
Computational Storage Engine	Compute Engine/Compute Namespace
Computational Storage Engine Environment	Virtual (Not currently defined)
Resource Repository	Programs
Downloaded CSF and CSEE	Downloaded programs
Pre-loaded CSF and CSEE	Device-defined programs
Activation	Activation
Function Data Memory (FDM)	Subsystem Local Memory (SLM)
Allocated FDM (AFDM)	Memory Range Set
Device Storage	NVM Namespaces

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

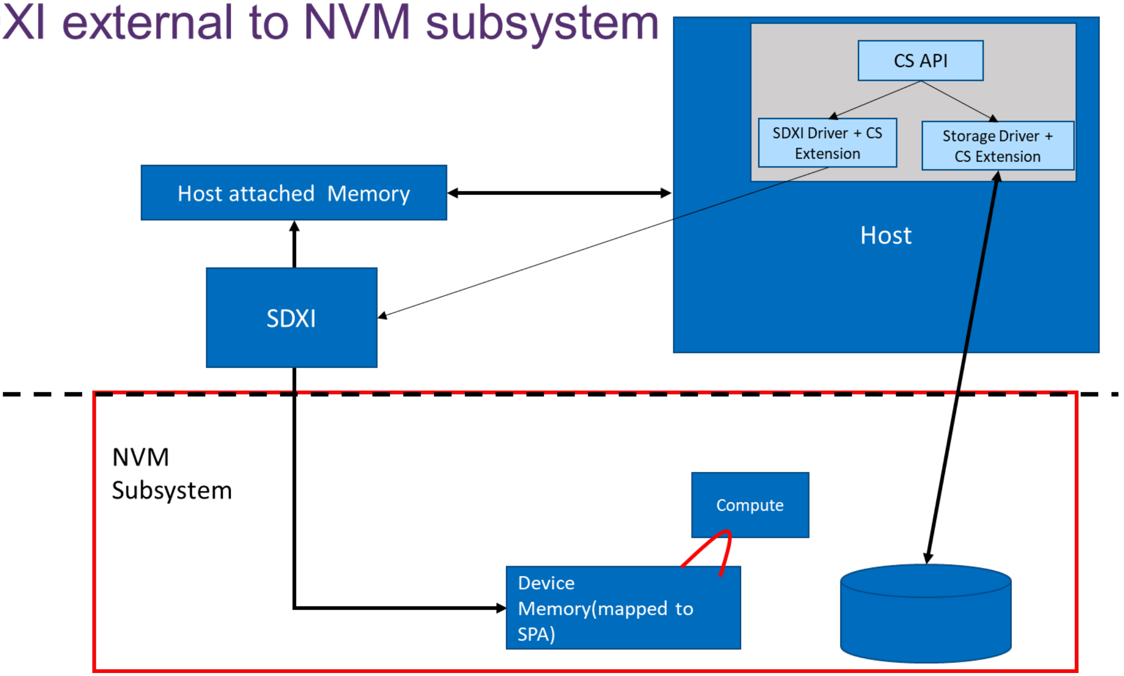
Combined SDXI+CS Architecture



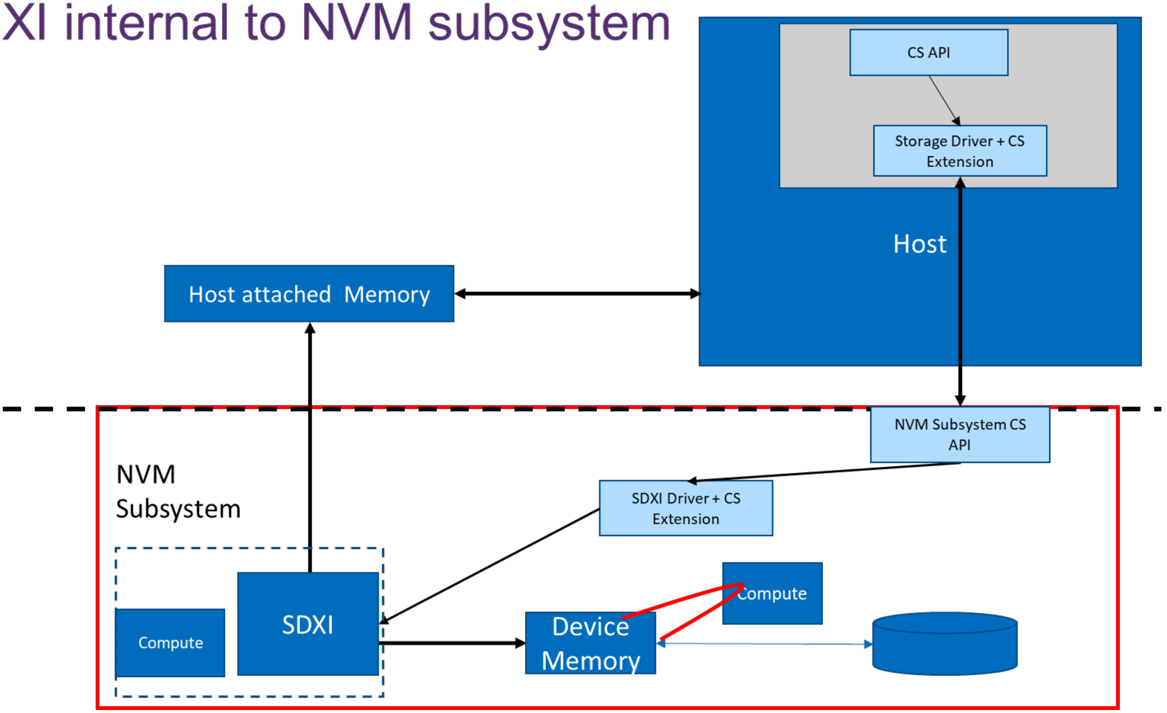
- SDXI used for data movement with Computational Storage used for compute
- Multiple SDXI producers in a CS Architecture
- SDXI enables data movement across multiple AFDM regions

SDXI and NVM Subsystem

SDXI external to NVM subsystem


CS API
SDXI Driver + CS Extension
Storage Driver + CS Extension
Host attached Memory
Host
SDXI
NVM Subsystem
Compute
Device Memory(mapped to SPA)

SDXI internal to NVM subsystem



CS + SDXI

- For more information about CS + SDXI, please attend:
  - “SDXI + Computational Storage Overview and Panel Discussion” by Jason Molgaard, Shyam Iyer, Fred Knight, Mats Oberg, Bill Lynn
- For more information about SDXI, please attend:
  - “Smart Data Accelerator Interface: Use Cases, Proof Points, v1.1 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

What do you think Computational Storage is

- Please complete a survey on your view of Computational Storage
  1) What is Computational Storage? (Multiple Choice)
  2) How would you use computational storage? (Fill in the blank)
  3) What is the future and evolution of computational storage? (Multiple Choice)
  4) Any other thoughts/ideas on computational storage?

Please take a moment to rate this session.

Your feedback is important to us.