SNIA COMPUTE + MEMORY + STORAGE SUMMIT

Architectures, Solutions, and Community
VIRTUAL EVENT, APRIL 11-12, 2023

Standardizing Computational Storage

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
- Next Steps
- Conclusion

The Continued Growth of Experience

- CS TWG is continuing to see growth
  - 48 companies, 258 individual members
- Work within SNIA Efforts
  - CS SIG – Webinars, Blogs, Events
  - SDXI – Sub-Group Collaboration
  - Security TWG – Addressing Security
- Collaborating with External Groups
  - NVM Express – Computational Programs

48 Participating Companies - 258 Member Representatives

Current Progress of TWG Output

- Architectural Document v1.0 has been released
- v1.1 under development
  - Security enhancements for multiple tenants
  - Chaining of Commands
  - Expansion of use cases
- API v0.8 public review version also available
- API v1.0 under development
  - Abort/reset handling
  - Device Memory
  - NVMe Computational Programs support
  - Other Miscellaneous Updates


SNIA
Advancing storage & information technology

Computational Storage Architecture and Programming Model

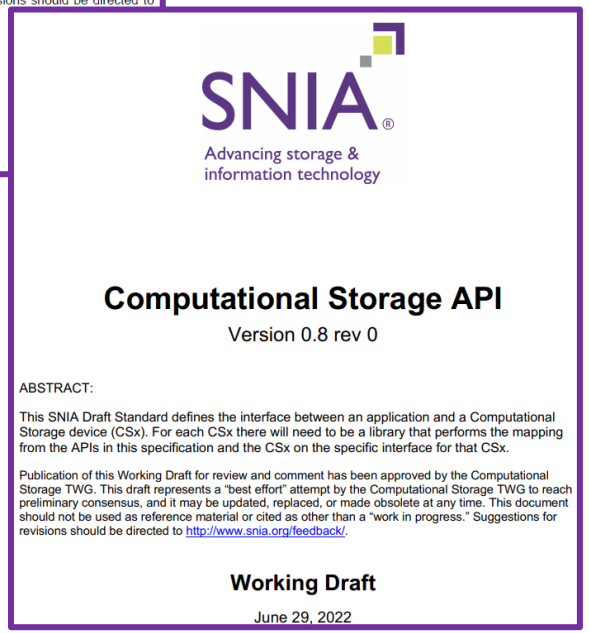
Version 1.0

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

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

SNIA Standard

August 30, 2022


SNIA
Advancing storage & information technology

Computational Storage API

Version 0.8 rev 0

ABSTRACT:

This SNIA Draft 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.

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/.

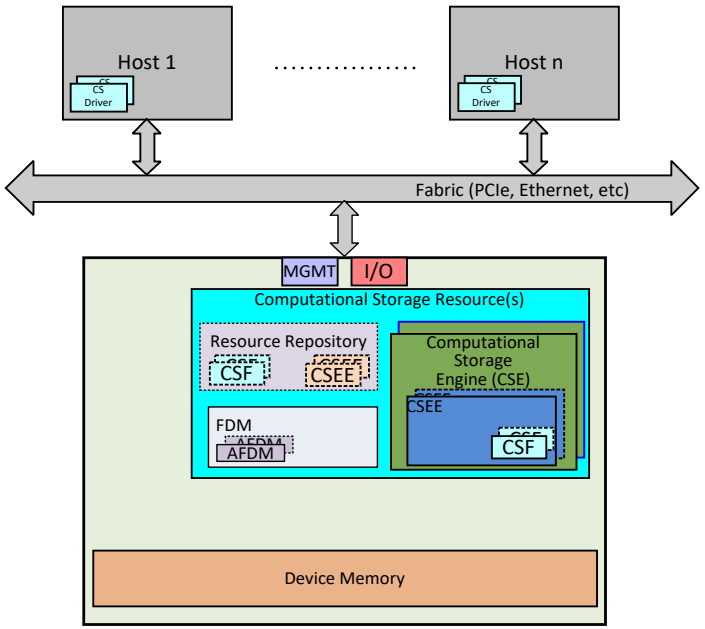
Working Draft

June 29, 2022

Architecture Overview

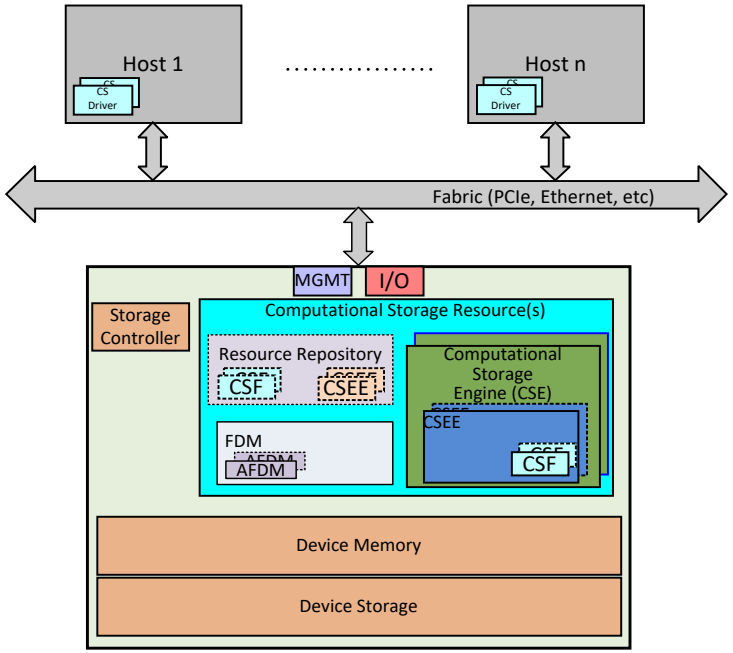
Computational Storage Architecture

Computational Storage Processor



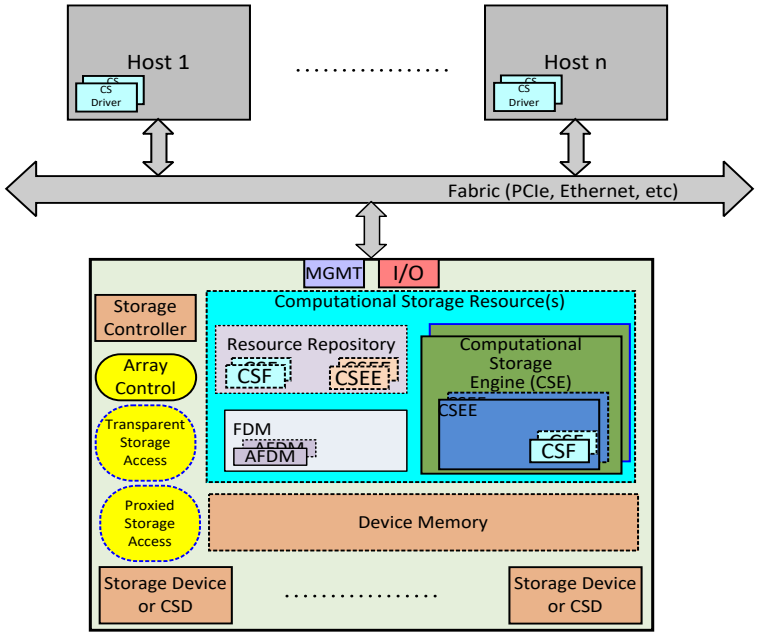
Computational Storage Processor (CSP)

Computational Storage Drive



Computational Storage Drive (CSD)

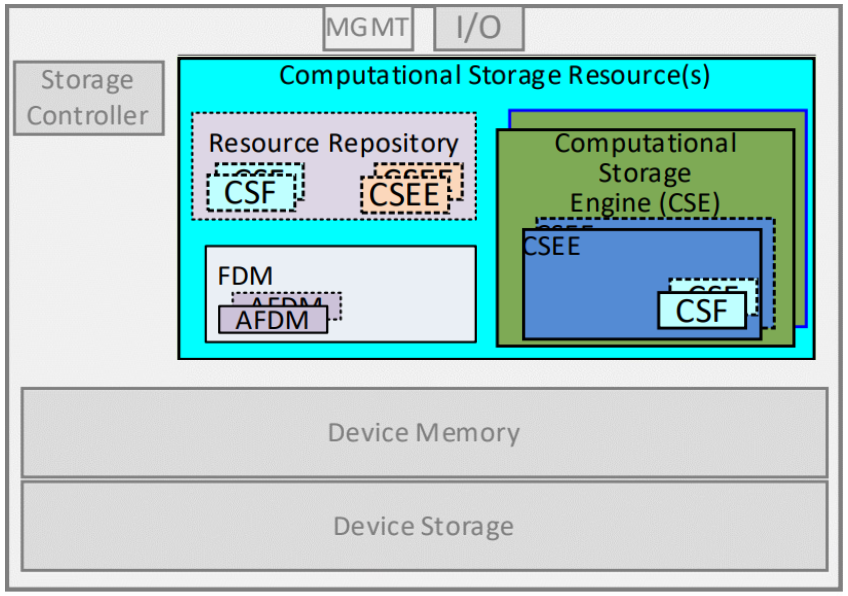
Computational Storage Array



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
- Sanitization
  - All instances of CSF operations should be terminated
  - All activated CSEEs and CSFs should be deactivated
  - All memory allocations associated with FDM should be removed and associated FDM cleared

Additional Security Considerations for v1.0

- Data at-rest Encryption
  - A CSx may need to encrypt or decrypt data to operate on it
- Key Management
  - A CSx may need to manage keys to support encryption or decryption of data
- Software Security
  - Software executing on a CSx may require:
    - Verification of the integrity of the downloaded program
    - Validation that the code originates from a particular source and has not been compromised

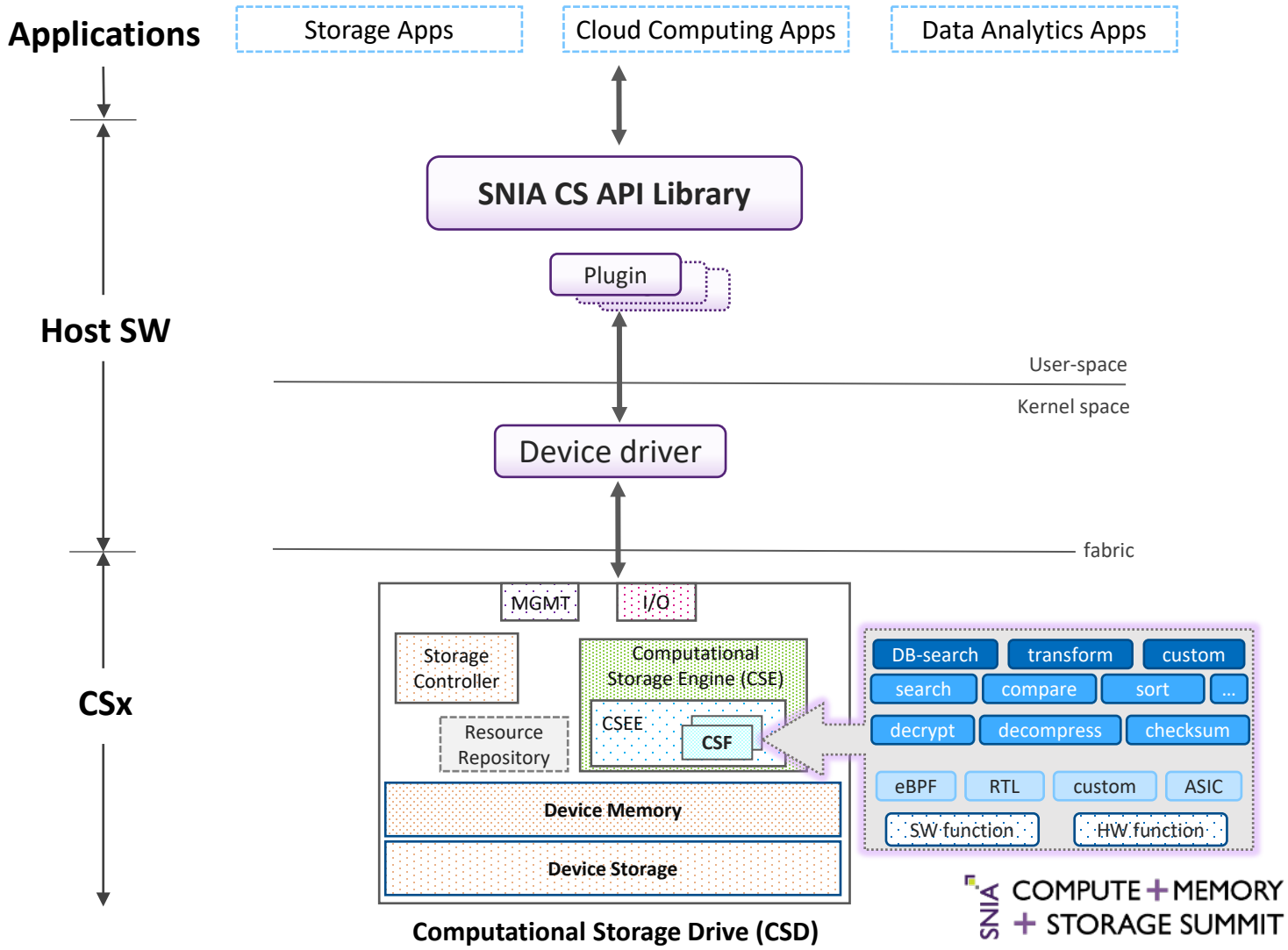
Multi-Tenant Security Considerations for version 1.1

- 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 is done with
  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 Authorization 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
  - CSP, CSD, CSA
- APIs hide device details
  - Hardware, Connectivity (local/remote)
- Abstracts device details
  - Discovery
  - Access
  - Device Memory (mapped/unmapped)
  - Near Storage Access
  - Copy Device Memory
  - Download CSFs
  - Execute CSFs
  - Device Management
- Extensible Interface
  - Plugins connect CSx to abstracted APIs
  - Hides vendor specific implementation details
- APIs are OS agnostic



Computational Storage API

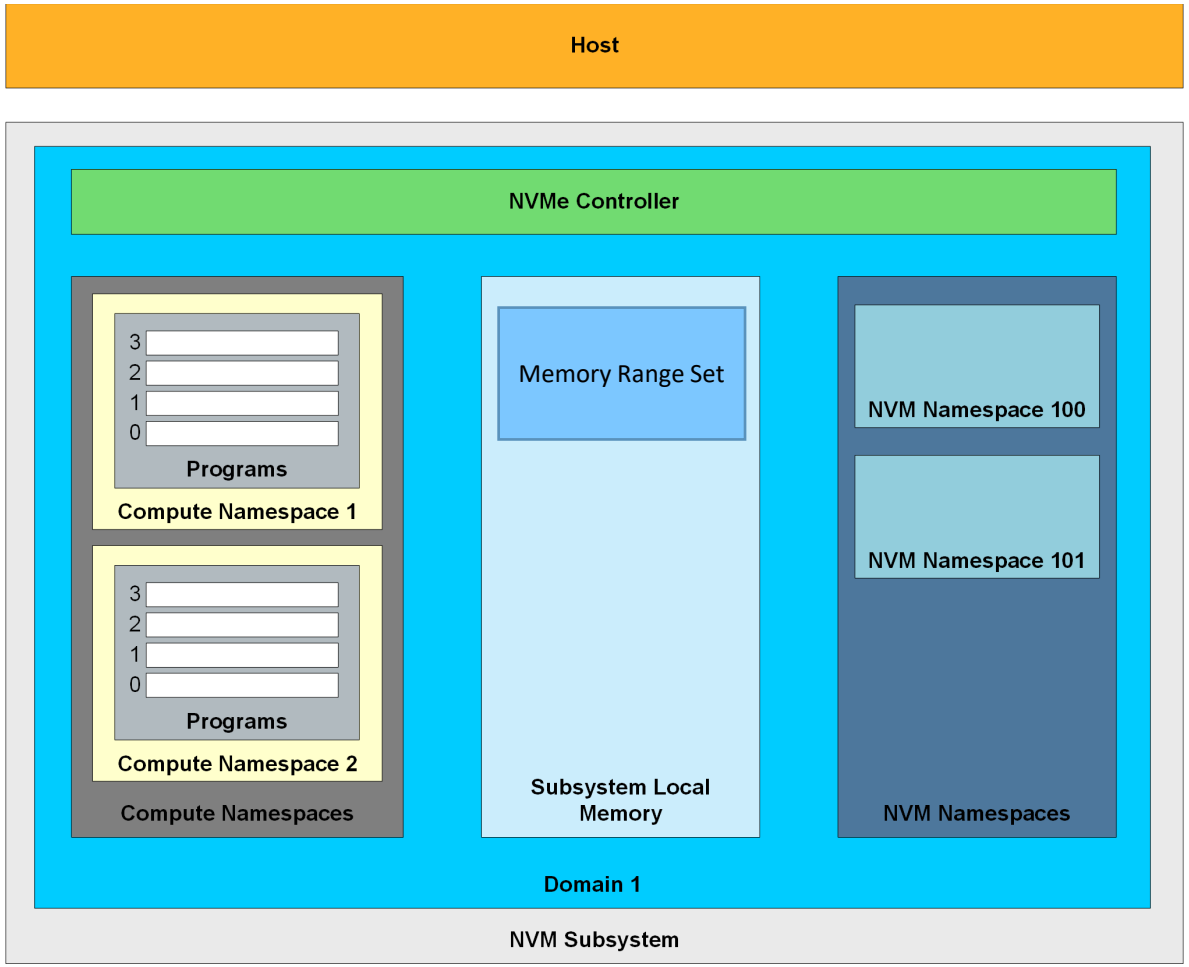
- For more information about the SNIA CS API, please attend:
  - SNIA Computational Storage API by Oscar Pinto

SNIA COMPUTE + MEMORY + STORAGE SUMMIT

Architectures, Solutions, and Community
VIRTUAL EVENT, APRIL 11-12, 2023

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 a 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 CSF
- 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

Differences between SNIA and NVMe

SNIA

- Defines CSEE
- CSF can directly access AFDM or Storage
- Supports an indirect model

NVMe

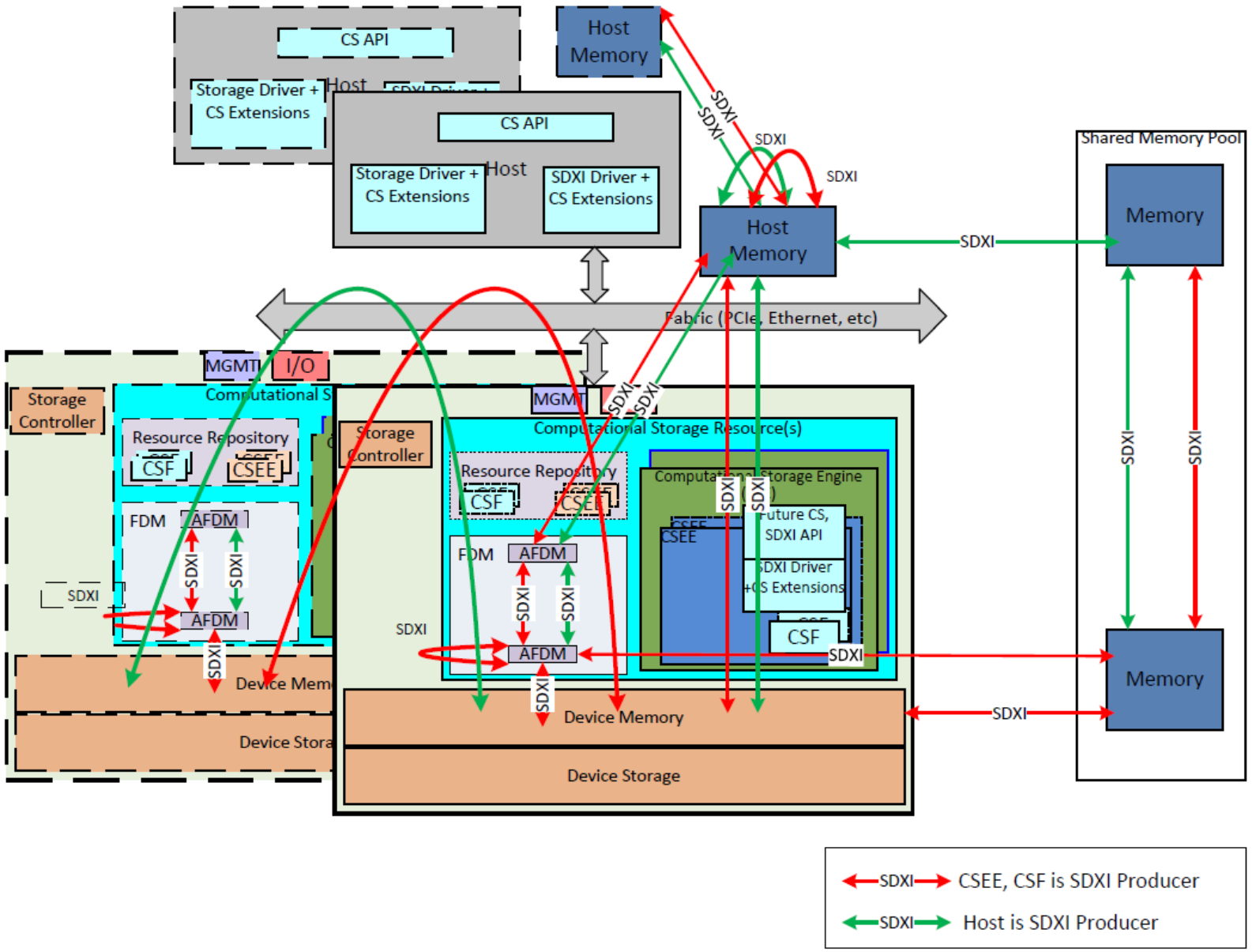
- CSEE is logical – no specific definition
- Program only accesses Memory Range Set
- Specific Execute command only

Next Steps

Progressing Computational Storage

- Complete API v1.0 Specification
  - Publicly release the API
- Complete Architecture and Programming Model v1.1
  - Multi-tenant Security Considerations
  - Chaining of Commands
  - Illustrative Examples growth
- CS and SDXI Collaboration
  - Combining Technologies

CS + SDXI



SNIA COMPUTE + MEMORY + STORAGE SUMMIT

Architectures, Solutions, and Community
VIRTUAL EVENT, APRIL 11-12, 2023

Conclusion

Interested? Join Us!

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

Additional Presentations on Computational Storage Standardization

- See the following Compute+Memory+Storage Summit presentations on Computational storage
  - SNIA Computational Storage API by Oscar Pinto
  - NVMe® Computational Storage by Kim Malone and Bill Martin

SNIA COMPUTE + MEMORY + STORAGE SUMMIT

Architectures, Solutions, and Community
VIRTUAL EVENT, APRIL 11-12, 2023

Please take a moment to rate this session.

Your feedback is important to us.

Post-Summit, visit www.snia.org/cms-summit for additional content.

