Flash Memory Summit

Standardizing Computational Storage

Bill Martin, Samsung

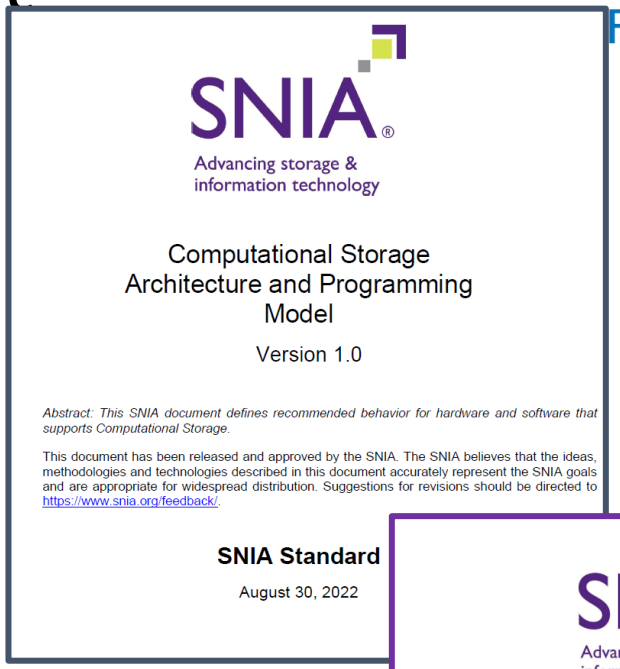
Jason Molgaard, Solidigm

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

- 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


SNIA
Advancing storage & information technology

Computational Storage Architecture and Programming Model

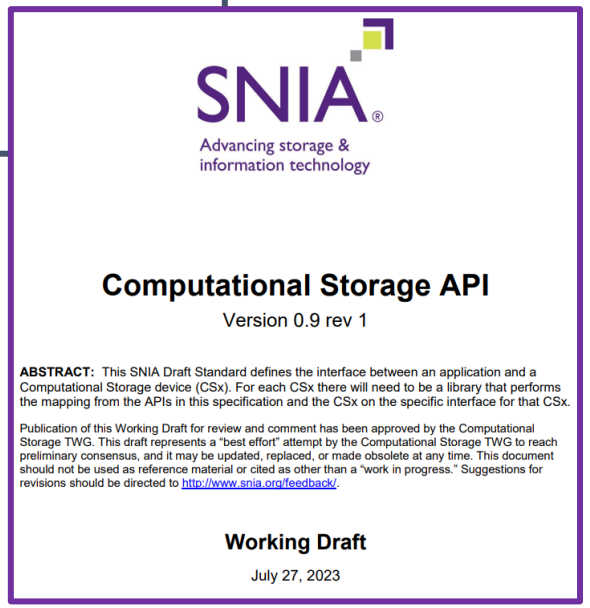
Version 1.0

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 0.9 rev 1

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

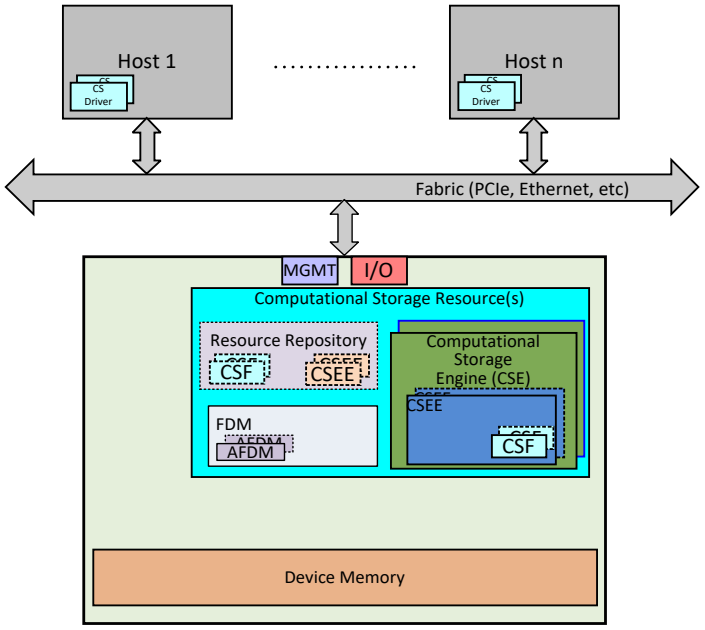
July 27, 2023

Architecture Overview

Computational Storage Architecture

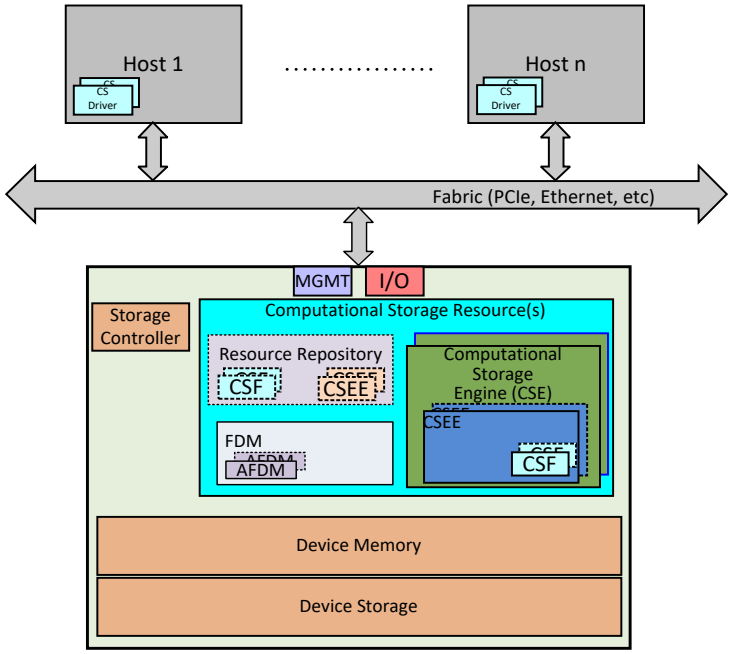
Flash Memory Summit

Computational Storage Processor



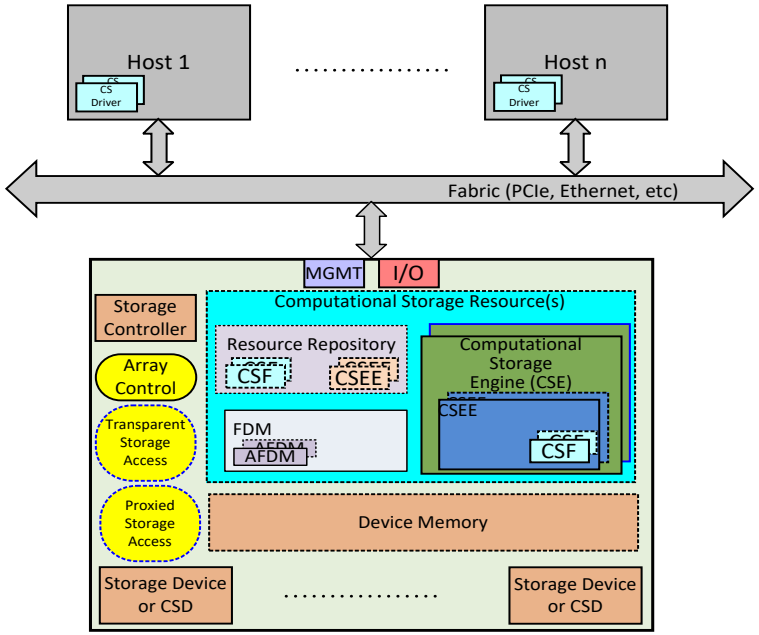
Computational Storage Processor (CSP)

Computational Storage Drive



Computational Storage Drive (CSD)

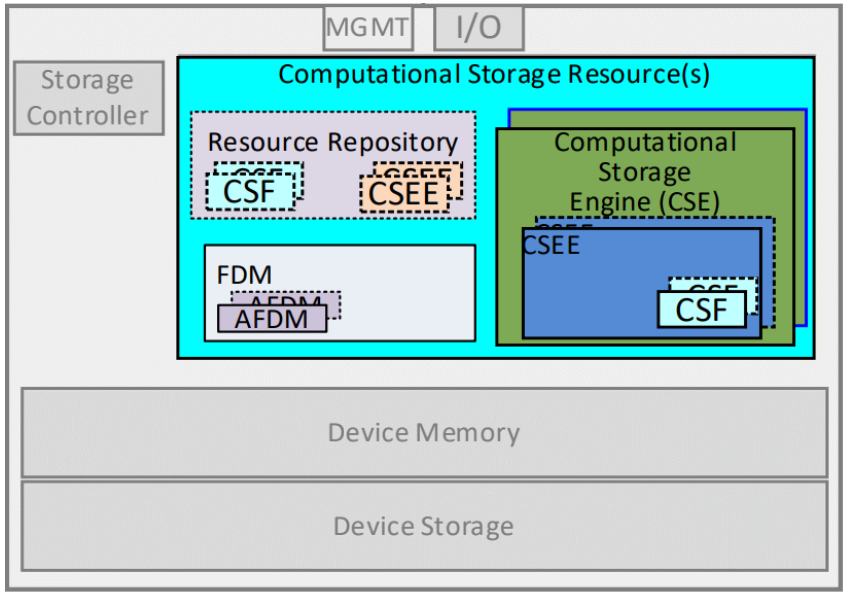
Computational Storage Array



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 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

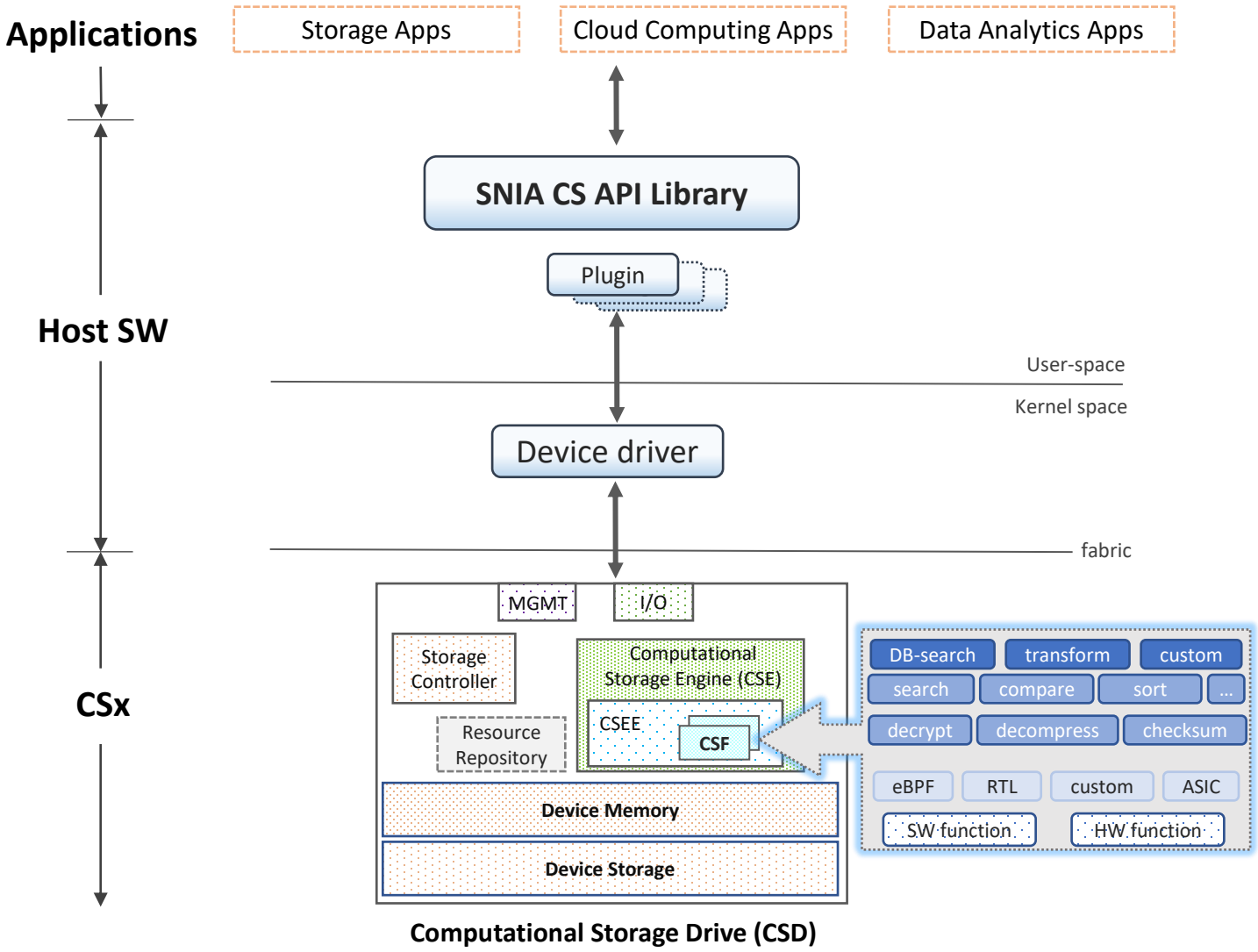
Sequencing of Commands

- 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
- 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



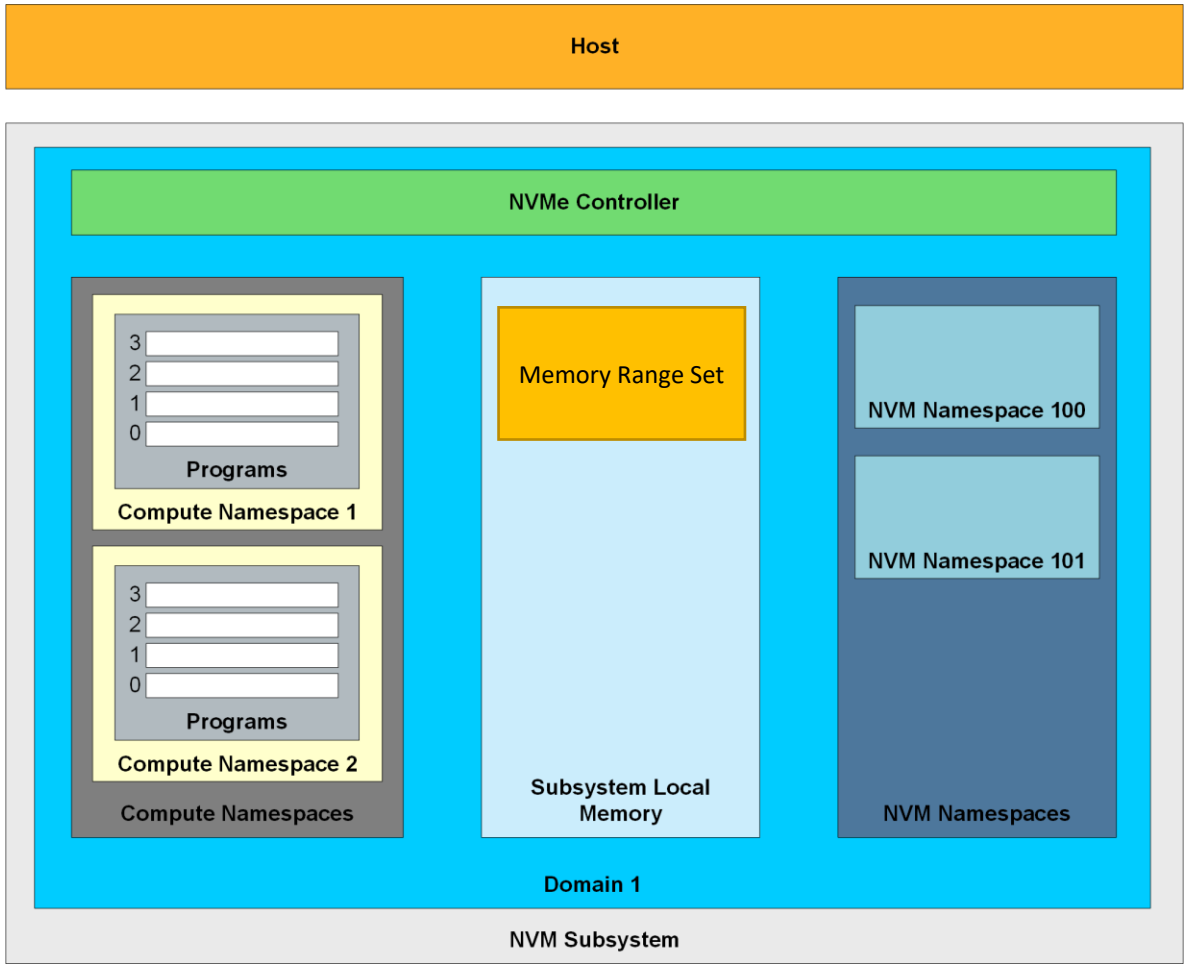
Computational Storage API

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

Flash Memory Summit

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 CSEE
- Activation
- Function Data Memory (FDM)
- Allocated FDM (AFDM)
- Device Storage

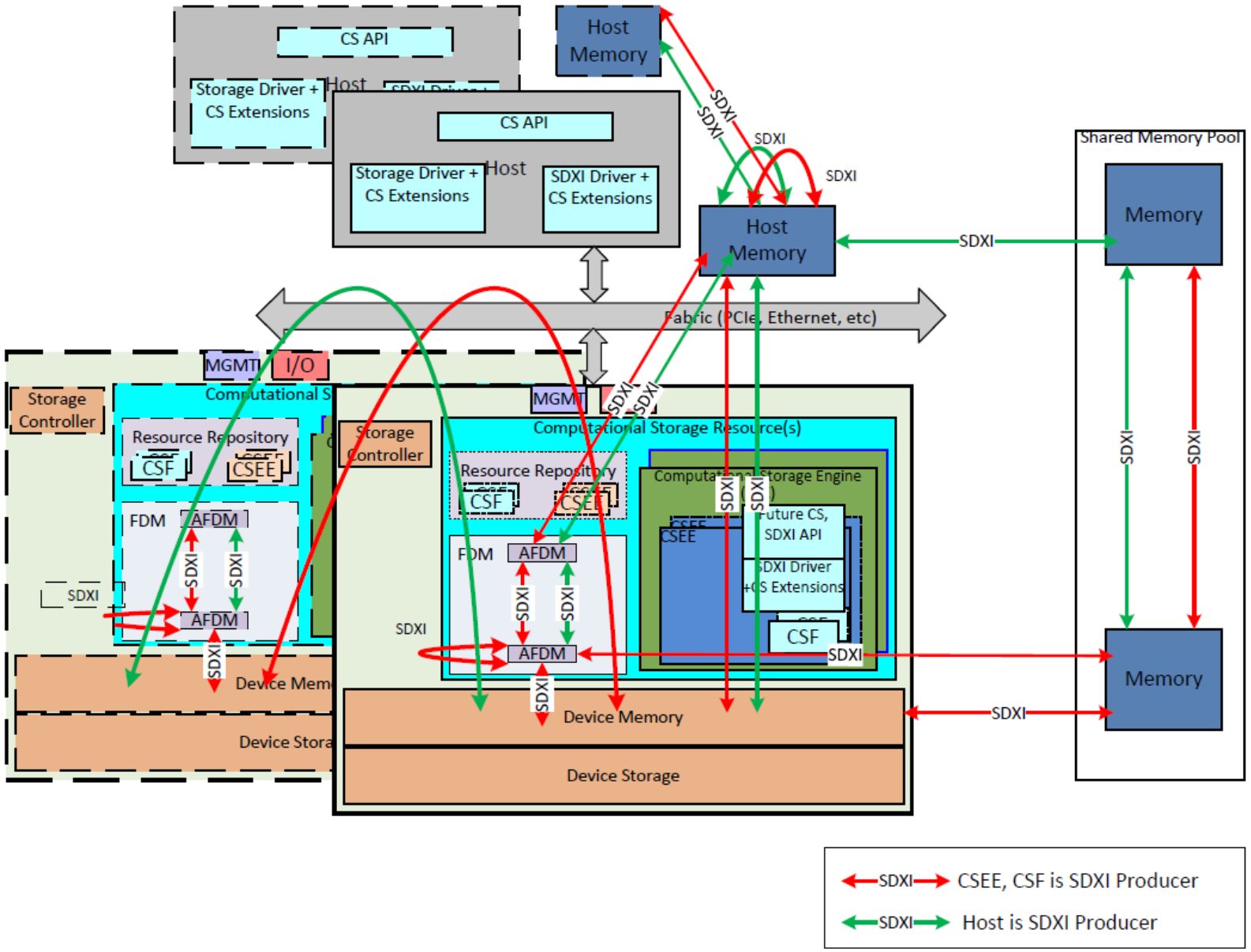
NVMe Terms

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

Flash Memory Summit

CS and SDXI Collaboration

CS + SDXI


CS API
Storage Driver + CS Extensions
Host
Host Memory
SDXI
CS API
Storage Driver + CS Extensions
Host
SDXI Driver + CS Extensions
Host Memory
Shared Memory Pool
Memory
Fabric (PCIe, Ethernet, etc)
MGMT
I/O
Storage Controller
Resource Repository
CSF
CSEE
FDM
AFDM
Computational Storage Resource(s)
Computational Storage Engine
Future CS, SDXI API
SDXI Driver + CS Extensions
Device Memory
Device Storage
SDXI CSEE, CSF is SDXI Producer
SDXI Host is SDXI Producer

Interested? Join Us!

Flash Memory Summit

- Join SNIA: https://www.snia.org/member_com/join-SNIA

- Join the Computational Storage TWG: https://members.snia.org/workgroup/index