

# SNIA SDXI TWG and Its Forays Towards Standardizing Memory Data Movement

Presented by: Shyam Iyer  
SDXI TWG Chair  
Distinguished Engineer, Dell

# Agenda

- Abstract
- Current Data Movement Standard
- Application Patterns and benefits of Data Movement & Acceleration
- SNIA SDXI TWG, Goals and Tenets
- A brief introduction to internals of SDXI Specification
- SDXI Community
- SDXI Futures
- References, Links and Announcements

# Abstract

- Data in use performance needs are expanding. At the same time, given the diversity of accelerator and memory technologies, standardizing memory to memory data movement and acceleration is gaining ground.
- SNIA's SDXI (Smart Data Accelerator Interface) TWG is at the forefront of standardizing this and has been working towards a v1.0 and recently put out a v0.9-rev1 version of the specification for public review.
- Many TWG members have discussed their motivation towards contributing to this specification.
- A memory to memory data movement interface is increasingly being seen as an enabler to persistent memory technologies and computational storage use cases.
- This talk will summarize the various use cases, various features discussed in the SDXI v0.9-rev1 specification, and SNIA's efforts around standardizing this work.

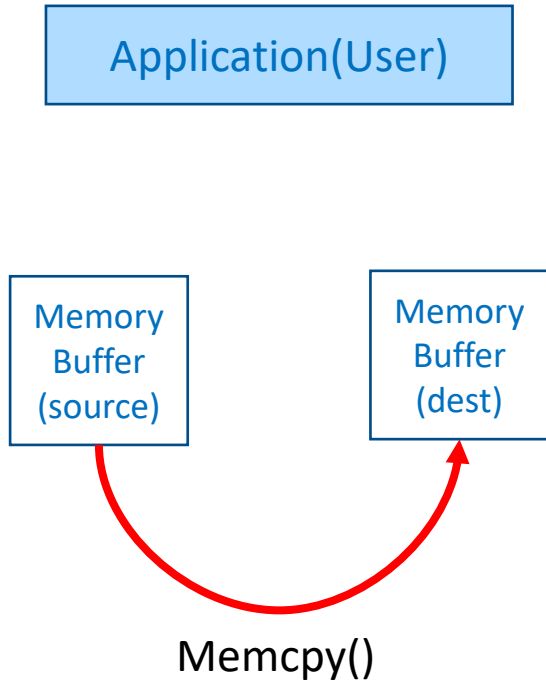
# Abstract

- Data in use performance needs are increasing
- Variety of accelerator and memory technologies, standardizing memory to memory data movement and acceleration is gaining ground
- SNIA's SDXI (Smart Data Accelerator Interface) TWG standardizing memory to memory data movement and acceleration
- v0.9-rev1 version of the specification out for public review
  - Various use cases, features enabled
- TWG members have discussed their motivation towards contributing to this specification
- Enables to persistent memory technologies and computational storage use cases

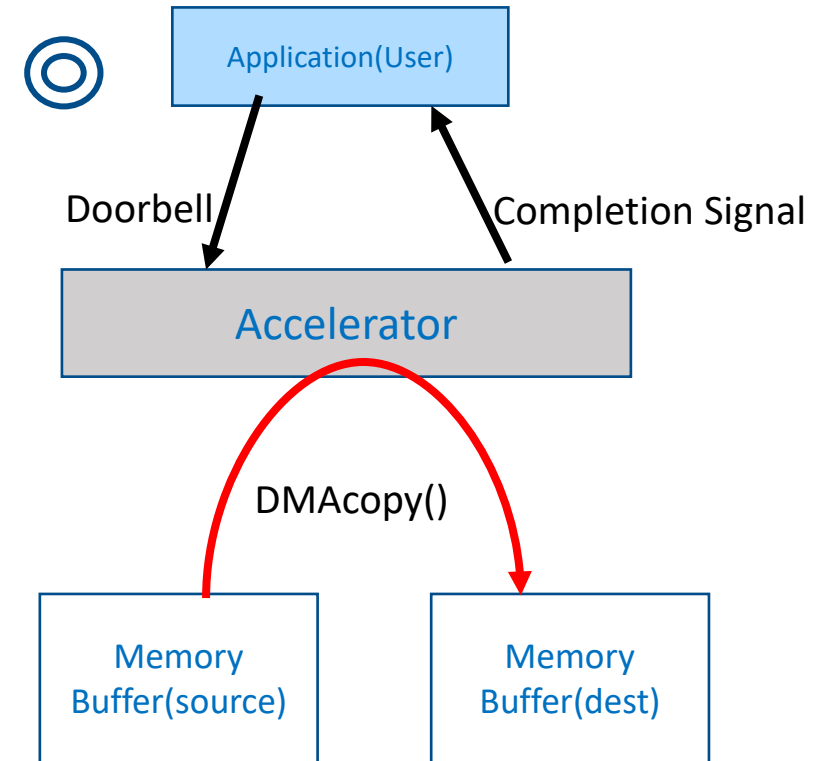
# Current Data Movement Standard

- Software memcpy is the current data movement standard
  - Takes away from application performance
  - Incurs software overhead to provide context isolation.
  - Offload DMA engines and their interfaces are vendor-specific
  - Not standardized for user-level software.

# Application Pattern 1

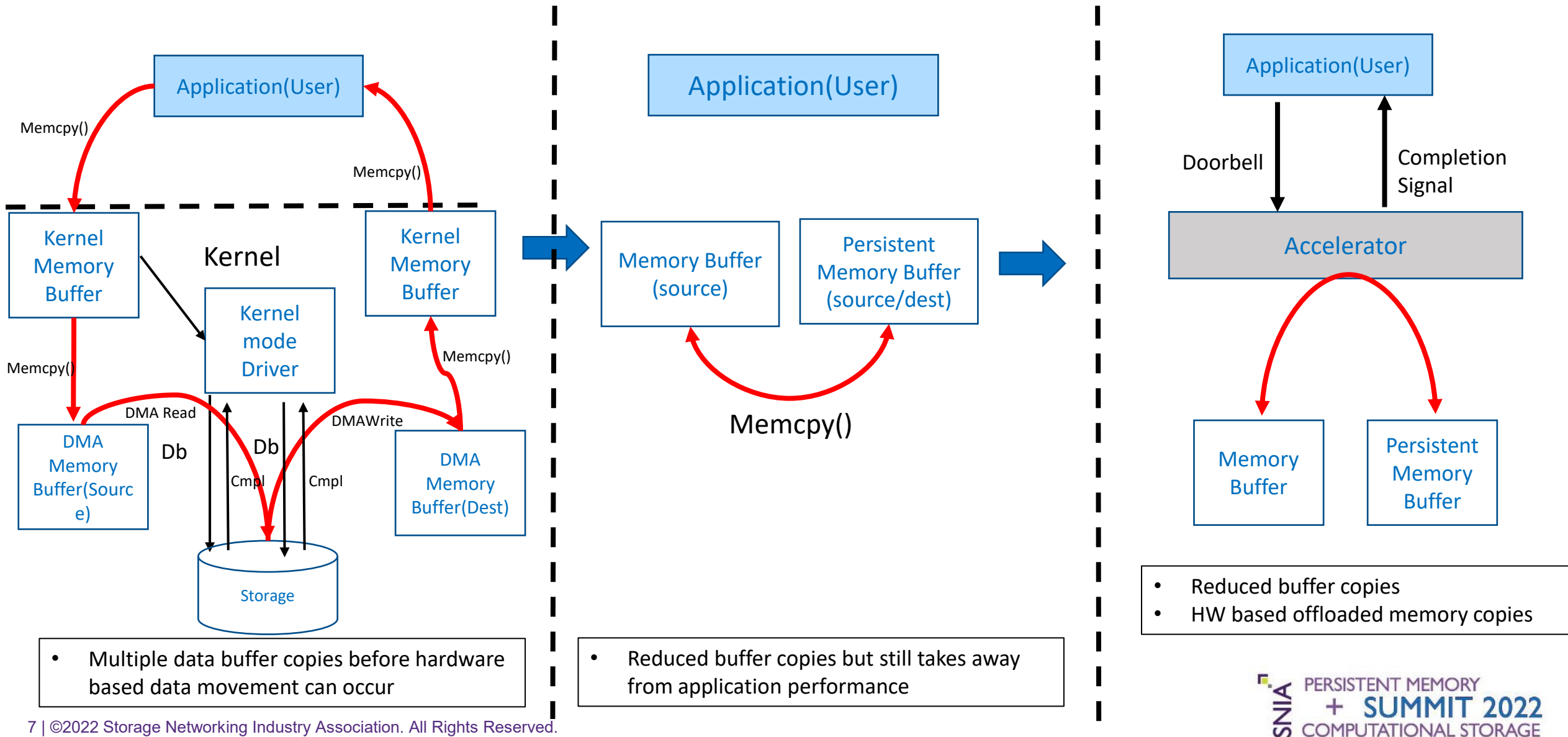


- Takes away from application performance

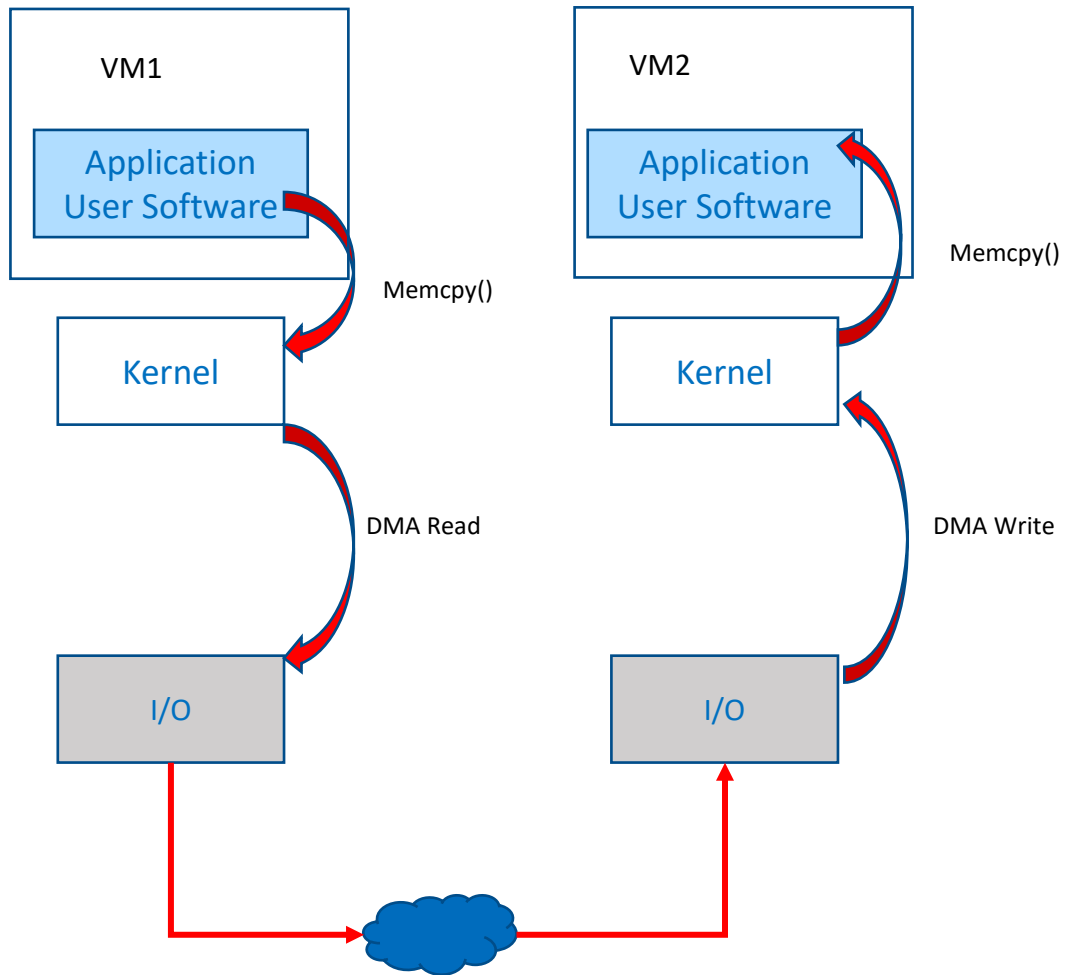


- HW based memory copies can be offloaded without affecting application performance

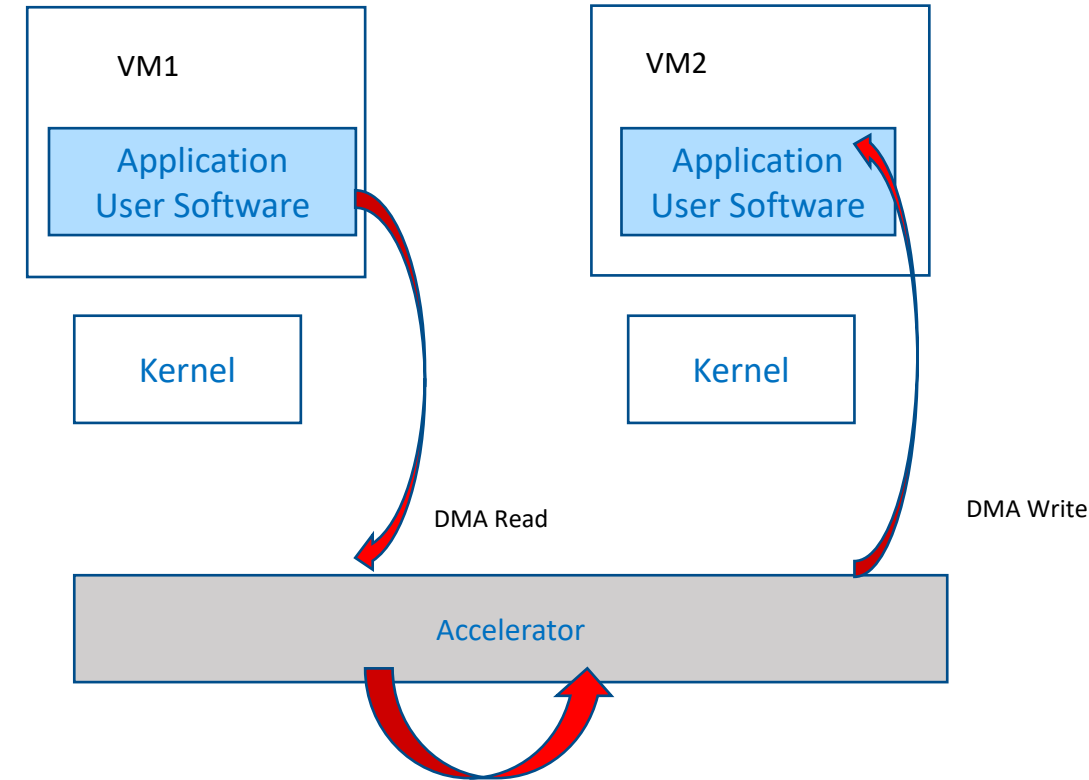
# Application Pattern 2



# Application Pattern 3



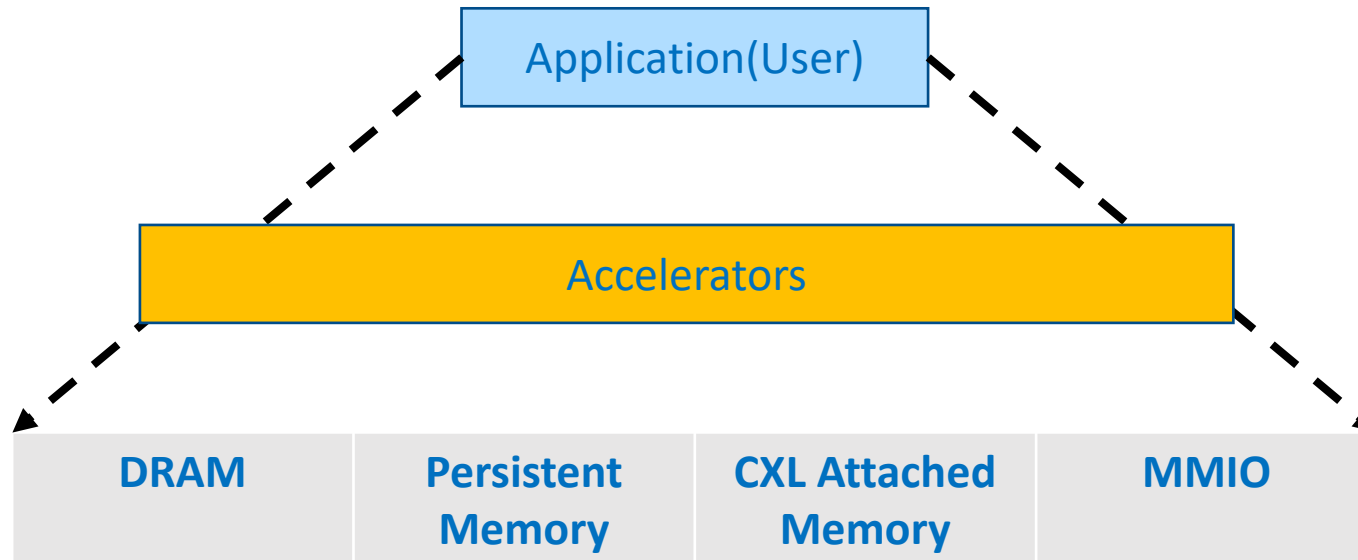
- Context isolation layers introduce multiple buffer copies



- Best of both: Context isolation layers and optimized HW based memory buffer copies



# Data *in use* Memory Expansion

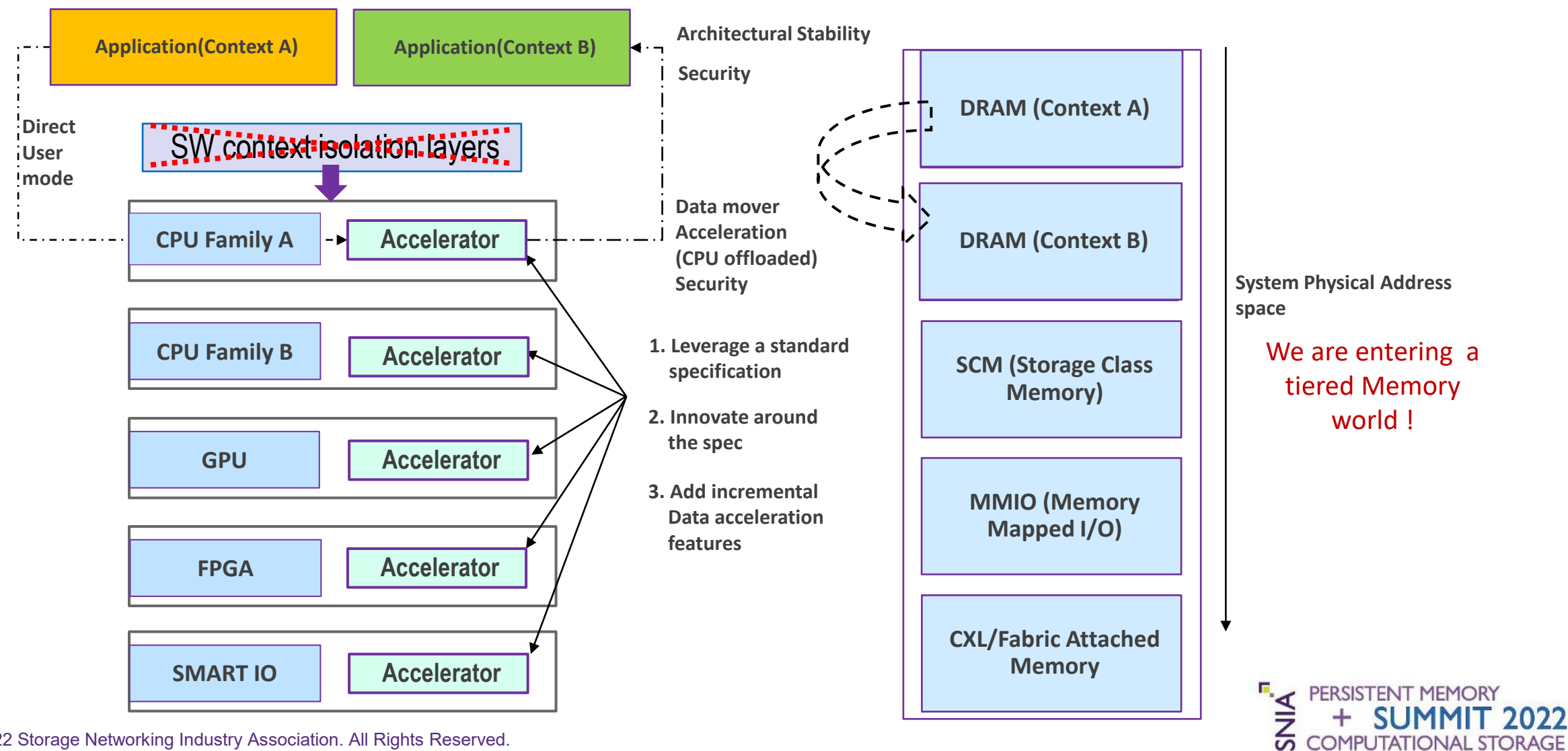


- Memory expansion expands the memory target surface area for accelerators
- Different tiers of memory
- Diversity in accelerator programming methods

# SDXI(Smart Data Accelerator Interface)

- Software memcpy is the current data movement standard
  - Takes away from application performance
  - Incurs software overhead to provide context isolation.
  - Offload DMA engines and their interfaces are vendor-specific
  - Not standardized for user-level software.
- Smart Data Accelerator Interface (SDXI) is a proposed standard for a memory to memory data movement and acceleration interface that is -
  - Extensible
  - Forward-compatible
  - Independent of I/O interconnect technology
- SNIA SDXI TWG formed in June 2020 tasked to work on this proposed standard
  - 28 member companies, 80+ individual members

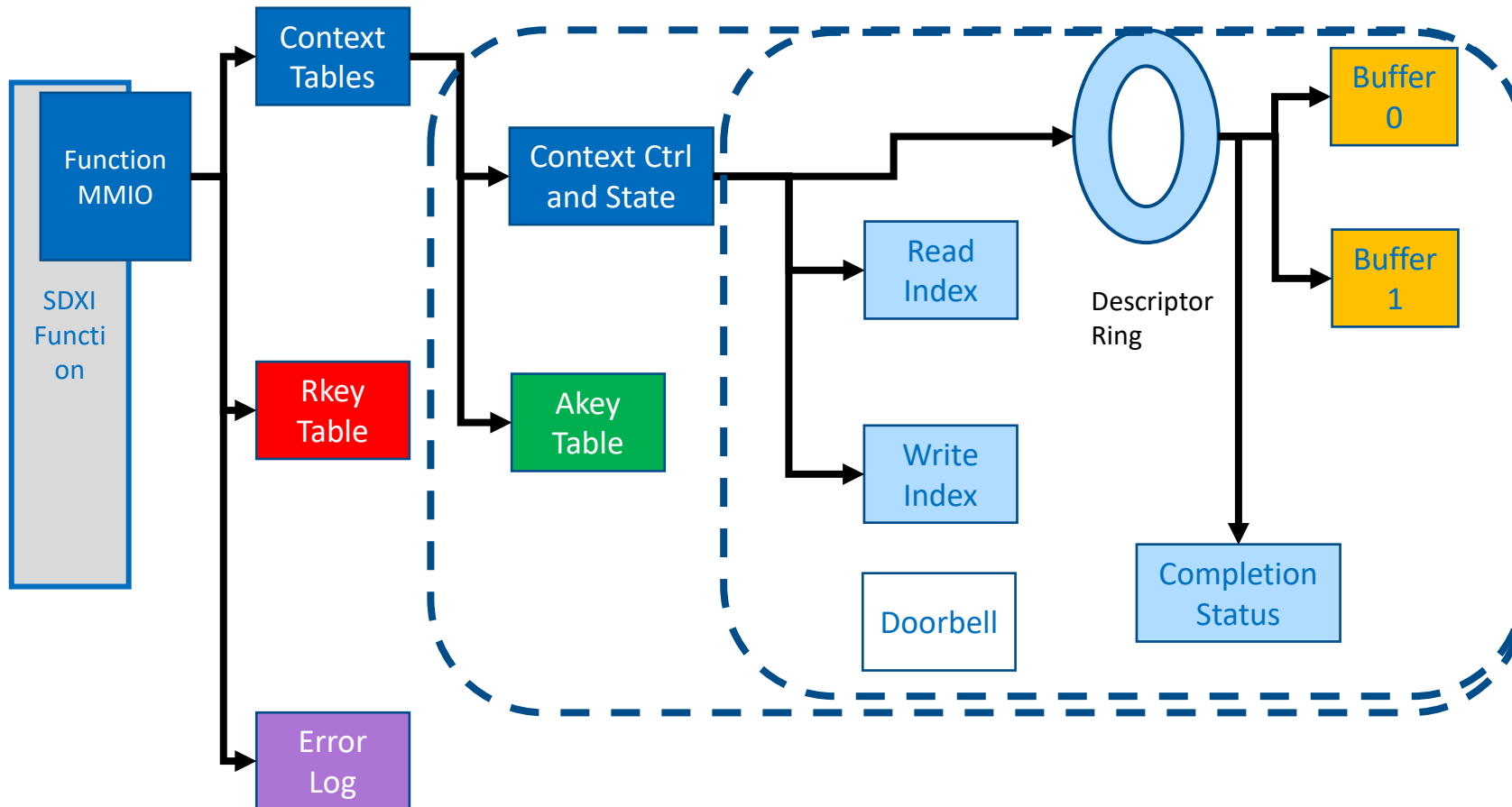
# SDXI Memory-to-Memory Data Movement



# SDXI Design Tenets

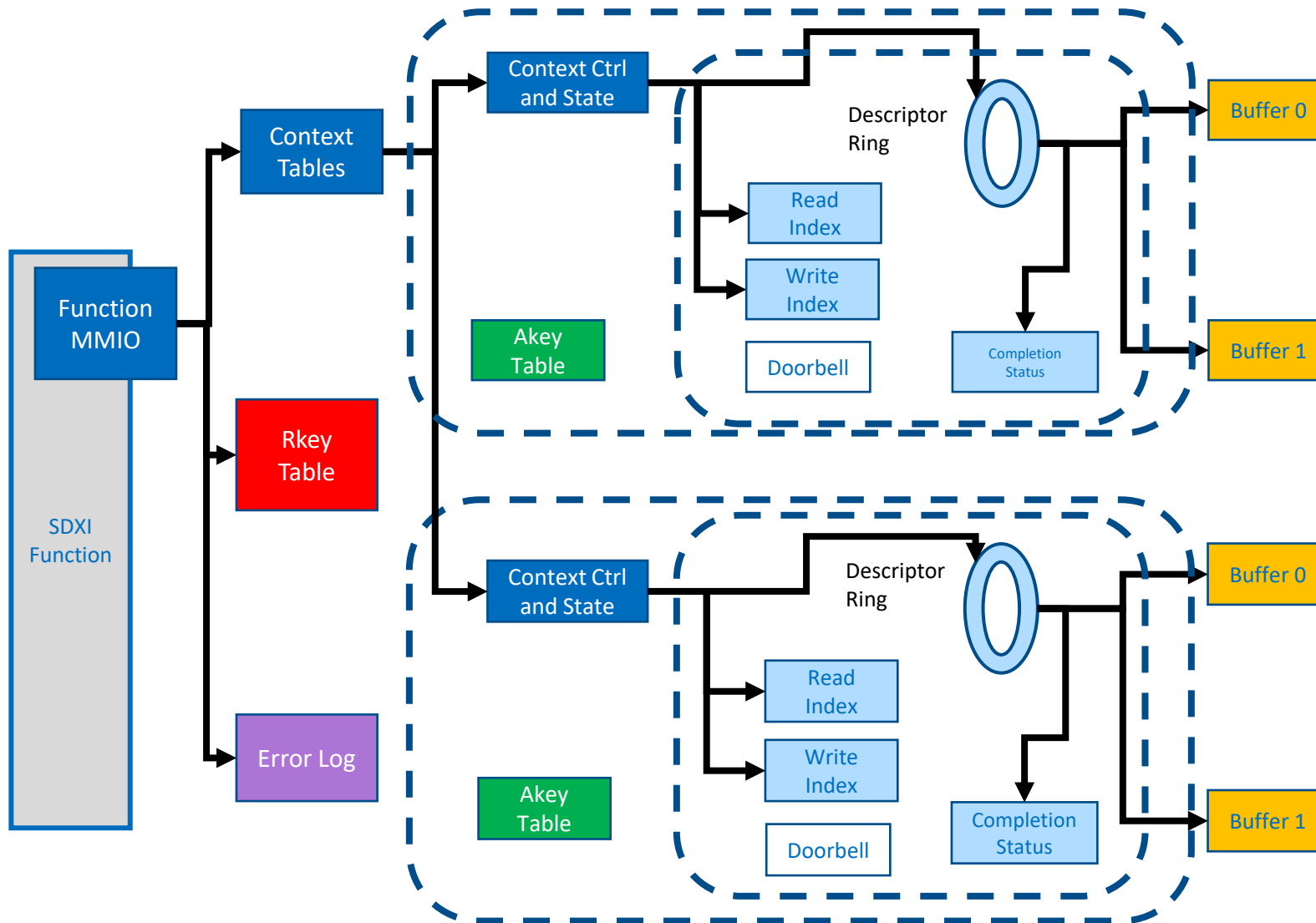
- Data movement between different address spaces.
  - Includes user address spaces, different virtual machines
- Data movement without mediation by privileged software.
  - Once a connection has been established.
- Allows abstraction or virtualization by privileged software.
- Capability to quiesce, suspend, and resume the architectural state of a per-address-space data mover.
  - Enable “live” workload or virtual machine migration between servers.
- Enables forwards and backwards compatibility across future specification revisions.
  - Interoperability between software and hardware
- Incorporate additional offloads in the future leveraging the architectural interface.
- Concurrent DMA model.

# Memory Structures(1) – Simplified view



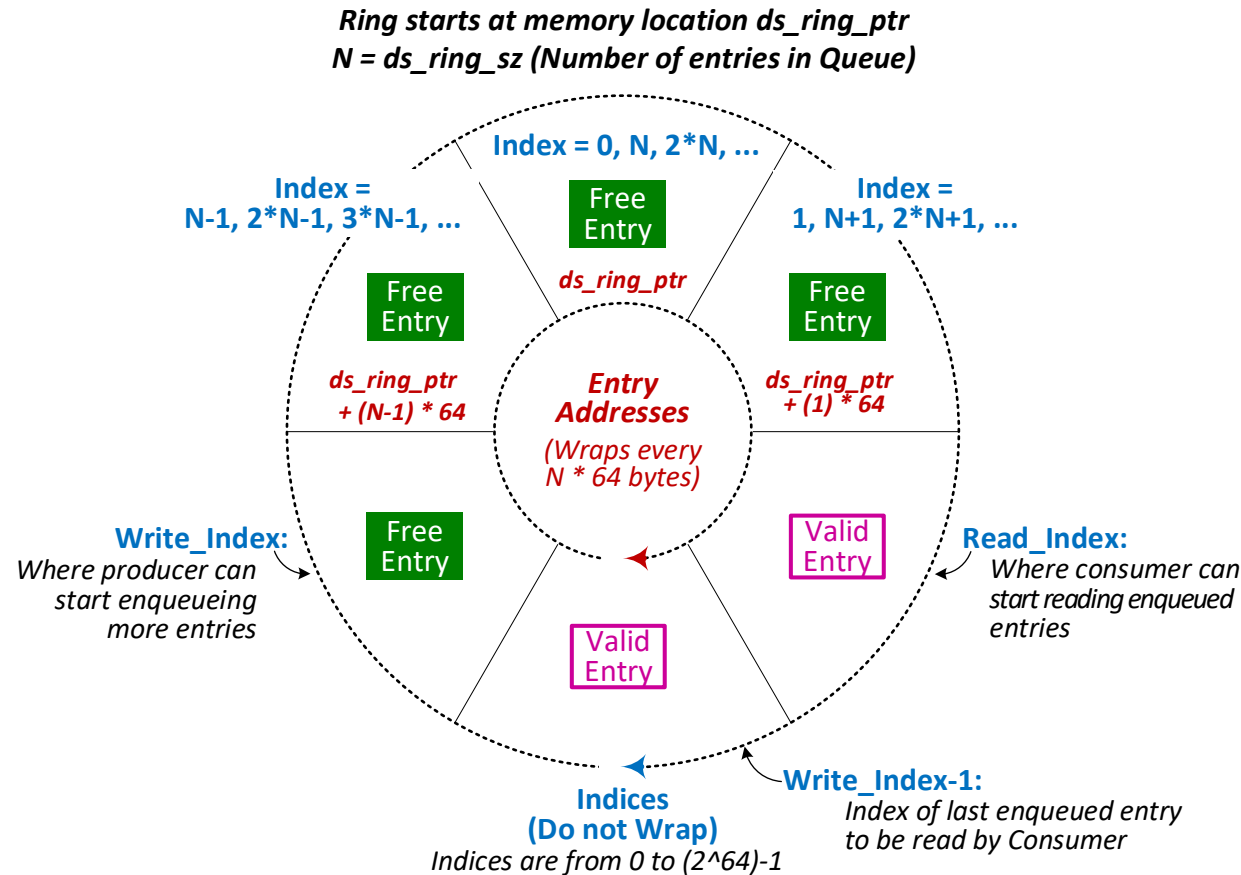
- Accelerator-independent standard way of function setup and control
- SDXI class code registered for PCIe implementations
- All states in memory
- One standard descriptor format
- Easy to virtualize

# Memory Structures(2) – Multiple Contexts



- Ring State directly managed by user space
- Multiple Contexts per function
- One way to log errors
- One way to control access to local memory buffers from remote functions(Rkey)
- Per context access to target address spaces(Akey)
- One way to start, stop and administer contexts

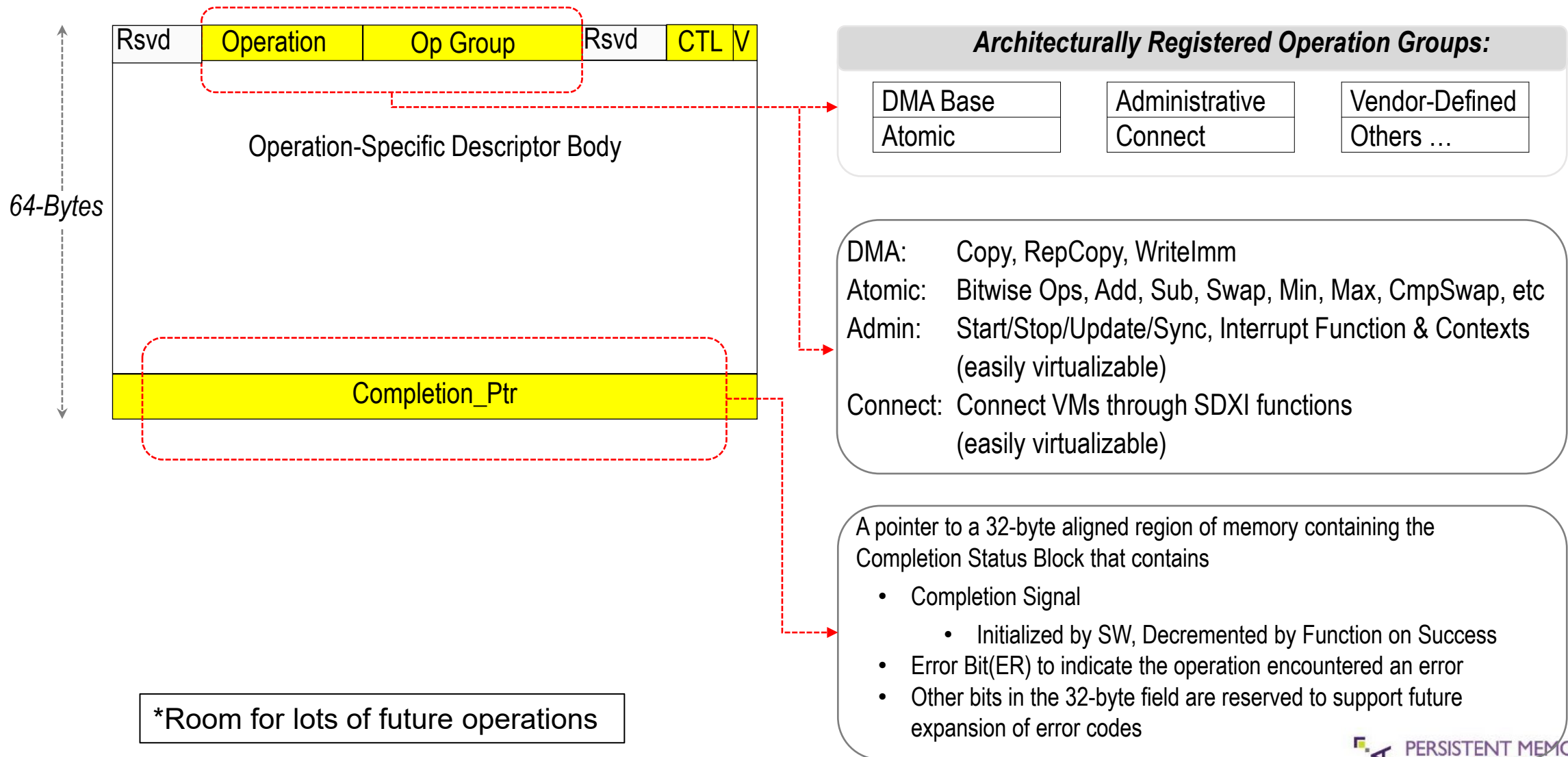
# Descriptor Ring



$$\text{EntryAddress} = ds\_ring\_ptr + ((\text{Index} \% ds\_ring\_sz) \ll 6)$$
$$\text{Write\_Index} - \text{Read\_Index} \leq ds\_ring\_sz$$

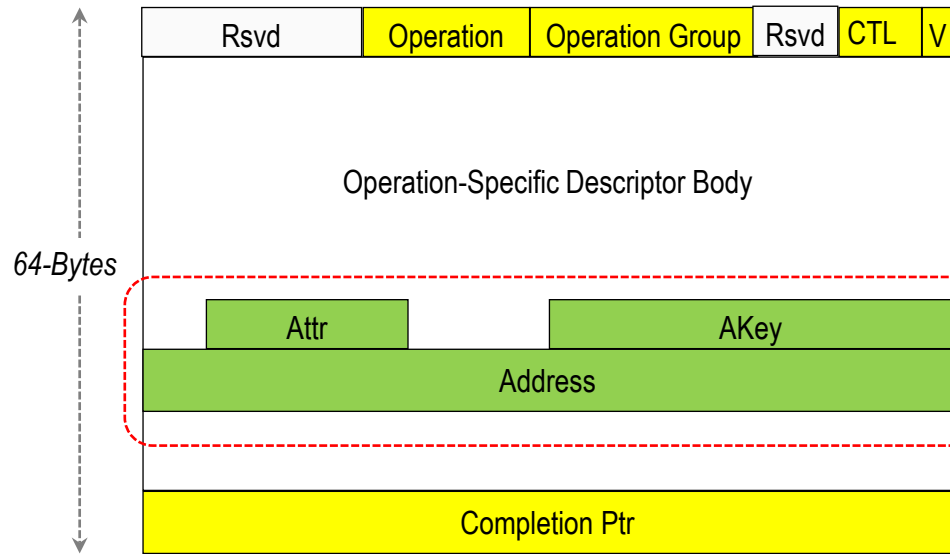
- Descriptors are processed (issued) in-order by function.
  - Executed out-of-order.
  - Completed out-of-order.
  - Read\_Index is incremented after each issue step.
- Function may aggressively read valid descriptors...
  - Between Read & Write indices w/o waiting on Doorbells from producers.
  - Doorbell ensures new descriptors are recognized.
- Maximum parallelism of operations. Quiescing & Serializing state at well-defined boundaries.

# A Standard Descriptor Format (1)





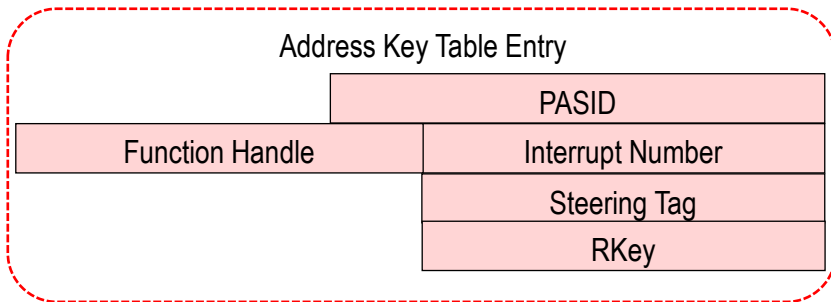
# A Standard Descriptor Format (2)



A memory location is always specified as a triple:

- Address Space ID: Index to Context Address Key Table Entry
- 64-bit Address
- Cacheability Attributes

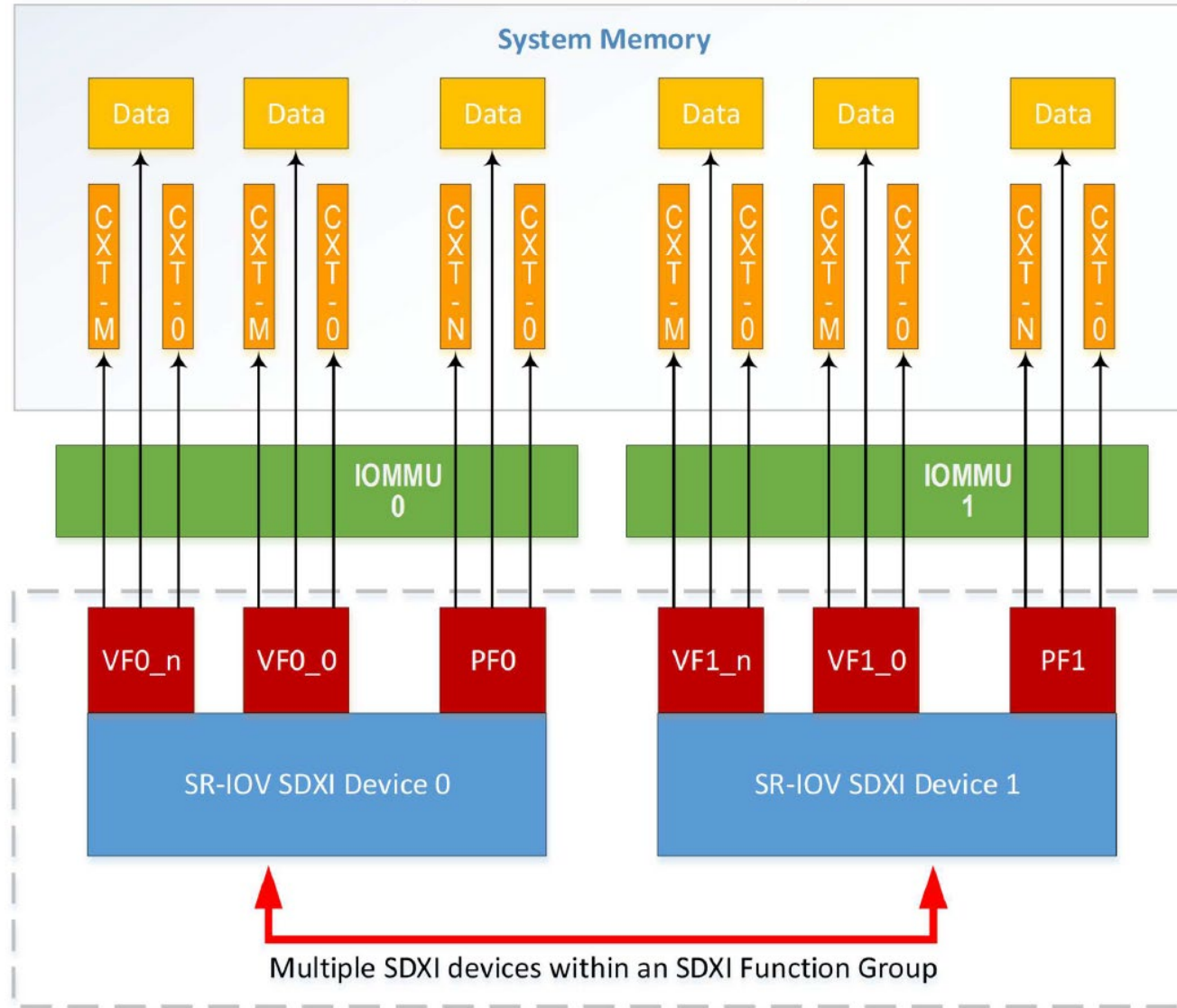
Generated Address can be HPA, HVA, GPA, GVA and always translated through IOMMU.



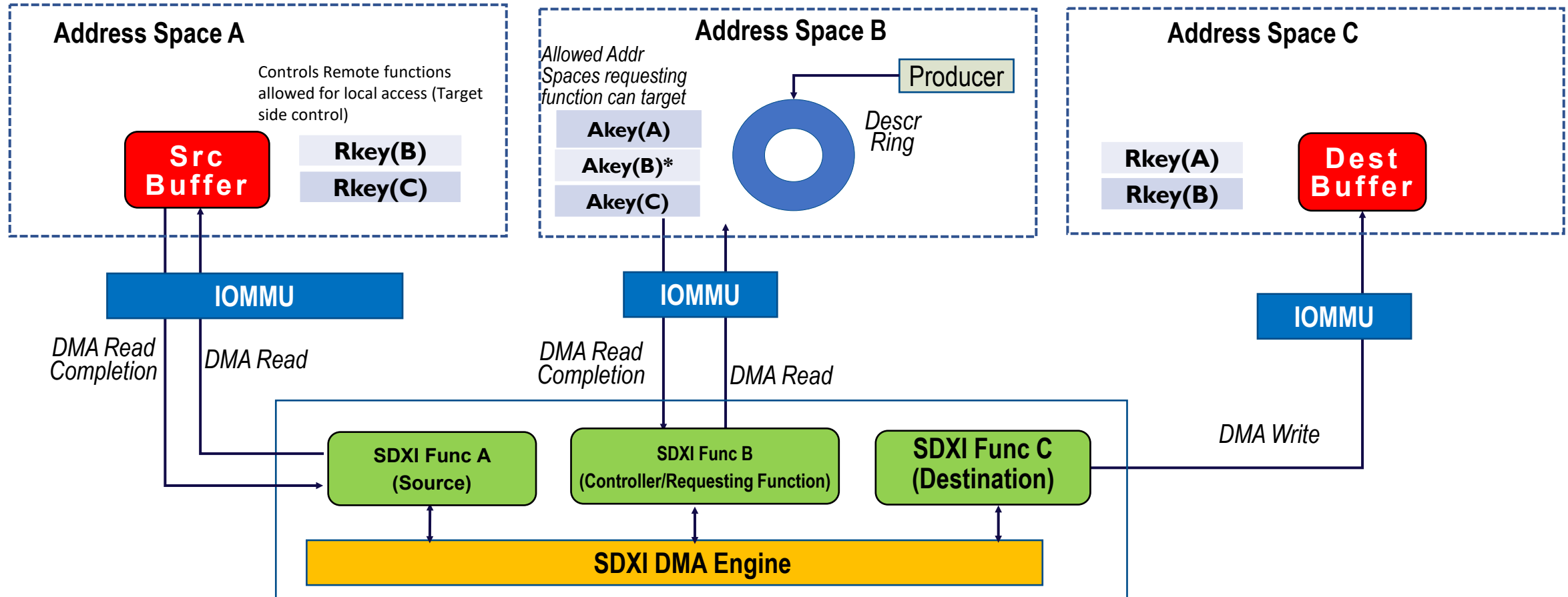
- An AKey table entry encodes all valid address spaces, PASIDs and interrupts available to the function context.
- Any descriptor within a context can reference an AKey table entry.
- An AKey is a requester side control
- The Akey also encodes the Rkey to be used by the Target address space. The Target Function uses the supplied RKey value to index into its RKey Table and obtain an RKey Table Entry. RKey is target side control.

\*Room for lots of future operations

# Contexts and SDXI Function Groups



# Multi-Address Space Data Movement within an SDXI function group (2)



# Active Community of SDXI TWG members

- TWG members have contributed in various ways towards improving the specification since initial contribution by founding members
- The contributions have resulted in improvements in many areas like -
  - Architectural behavior of Administrative Start/Stop operations
  - Architectural Context States and Function States
  - Cache Injection mechanisms
  - Deterministic discovery of SDXI Function Groups
  - Rich Completion Status and Error Reporting
  - Improved mechanisms to control local memory access from remote functions
  - Usage models
  - Interconnect agnostic specification language

# Active Contributors

- Curtis Ballard, HPE
- Beau Beachamp, MemVerge
- Richard Brunner, VMware
- Xiangping Chen, Dell
- Don Dutile, IBM
- Paul Hartke, AMD
- Shyam Iyer, Dell Inc
- Travis Hamilton, Arm
- Brian Hirano, Micron
- Frederick Knight, NetApp
- Santosh Kumar, SK Hynix
- James Leighton, Western Digital
- Bill Martin, Samsung
- John Maroney, Micron
- J Metz, AMD
- William Moyes, AMD
- Philip Ng, AMD
- Murali Ravirala, Microsoft
- Dwight Riley, HPE
- Alexandre Romana, Arm
- Glen Sescila, Dell Inc
- Paul Von Stamwitz, Fujitsu
- Jason Wohlgemuth, Microsoft

# What to expect: SDXI Futures

- Release v1.0
- Plan post v1.0 activities. Current charter includes:
  - New data mover operations for smart acceleration
  - Data mover operations involving persistent memory targets
  - Cache coherency models for data movers
  - Security Features involving data movers
  - Management architecture for data movers(includes connection manager)
- Some additional discussion topics being considered post v1.0
  - RAS improvements
  - QoS improvements
  - Latency improvements
  - CXL related discussions
  - Heterogenous environments

# Links

- SDXI Specification v0.9-rev1 available for Public Review
  - [https://www.snia.org/tech\\_activities/publicreview](https://www.snia.org/tech_activities/publicreview)
- SNIA SDXI Page
  - <https://www.snia.org/sdxi>
- PM + CS Summit 2021
  - <https://www.snia.org/educational-library/new-path-better-data-movement-within-system-memory-computational-memory-sdxi>
- SDC 2020 presentation
  - <https://www.youtube.com/watch?app=desktop&v=iv2GUfnxG-A>
- In memory compute summit
  - <https://www.youtube.com/watch?v=iv2GUfnxG-A>
- SDC 2021 panel session
  - <https://www.youtube.com/watch?v=PrIQZF2a4YI>
- New Subgroup! SDXI + CS Subgroup
  - Membership Criteria:
    - Should be a member of SNIA's SDXI TWG & CS TWG

# Please take a moment to rate this session.

- Your feedback is important to us.