SNIA DEVELOPER CONFERENCE

SDC 24

BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

The NVM Express® Ratified details of Live Migration

Presented by Mike Allison (Samsung)

What is NVMe™ Live Migration?

- The capability that enables a host to migrate NVMe controllers (i.e., controllers) and the attached namespaces to those controllers as a Virtual Machine (VM) is migrated
- Wikipedia defines the Live Migration of VMs as moving a running VM between different physical machines without disconnecting storage
  - This new NVMe technology:
    - Standardizes how not to disconnect the attached storage
    - Enables moving the attached storage between different NVMe devices
      - Between the same type of product
      - Between two different types of product from the same vendor
      - Between two different vendors

What is NVMe™ Live Migration?

- Host Managed Live Migration
  - TP4159 PCIe Infrastructure for Live Migration
    - Ratified on 8/5/2024
    - Included in the:
      - NVM Express® Base Specification Revision 2.1
      - NVM Express® NVM Command Set Specification Revision 1.2
  - I was the main author of the technical proposal but final documented protocol was due to major inputs from:
    - Host companies
    - Controller ASIC vendors
    - SSD vendors

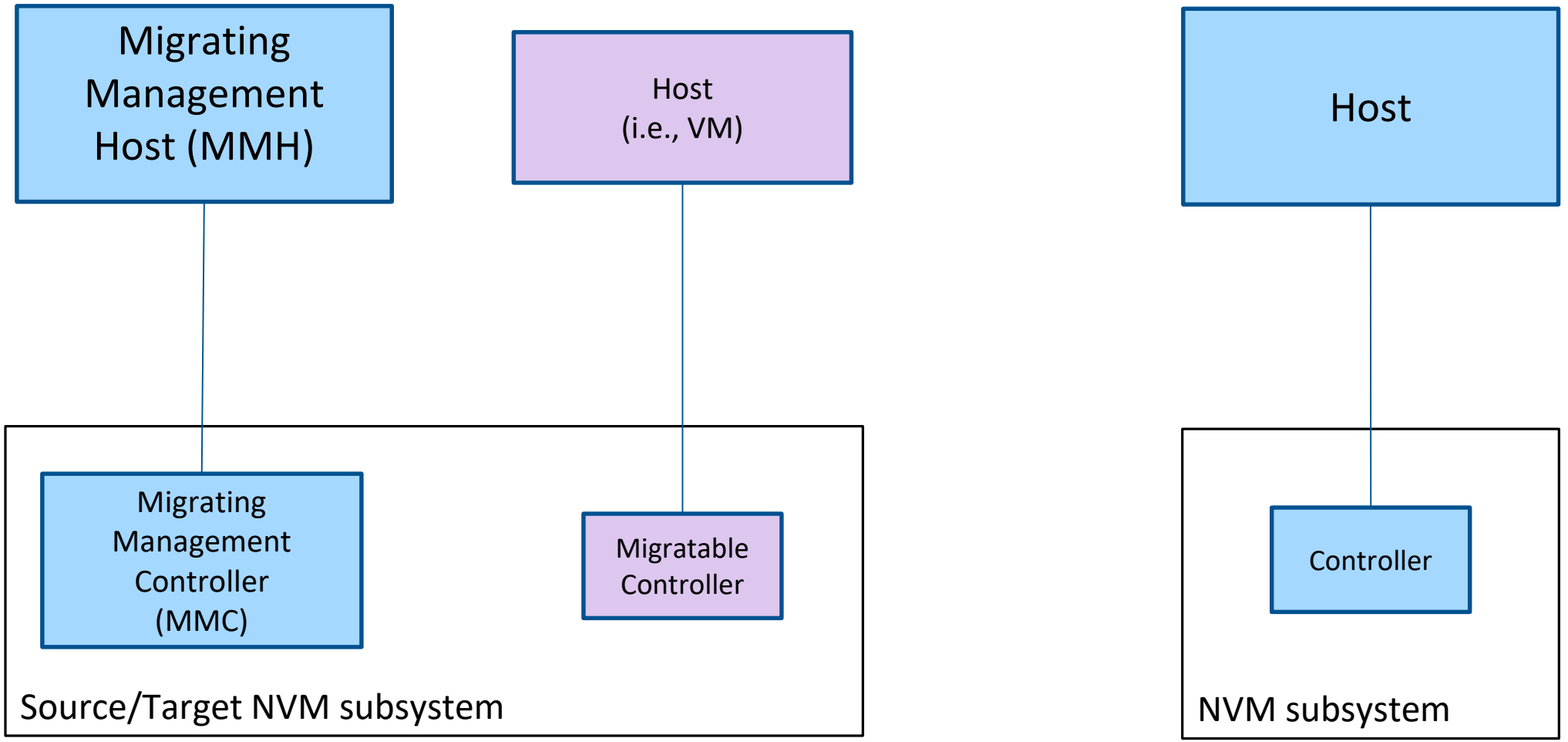
NVMe™ Host Managed Live Migration

- The new NVMe protocol allows the HOST to manage the storage migration similar to the way Wikipedia explains the technique for migrating a Virtual Machine using the Pre-Copy Memory Migration technique
- Pre-copy Storage Migration
  - Pre-Copy Phase
    - HOST copies attached namespaces from source to destination while the controller is still processing commands
    - If a namespace is modified during the Pre-Copy Phase, the HOST is notified what changed so the HOST can copy what changed
    - If the controller modifies host memory due to processing commands, the HOST is notified what changed so the host can copy that VM memory again
  - Stop-and-copy Phase
    - HOST suspends the source controller
    - HOST copies over any remaining changed namespaces or change VM memory
    - HOST copies the NVMe controller state from source to destination
    - HOST resumes the destination NVMe controller
- The Pre-copy Storage Migration is allows the HOST to migrate the VM and the storage in parallel

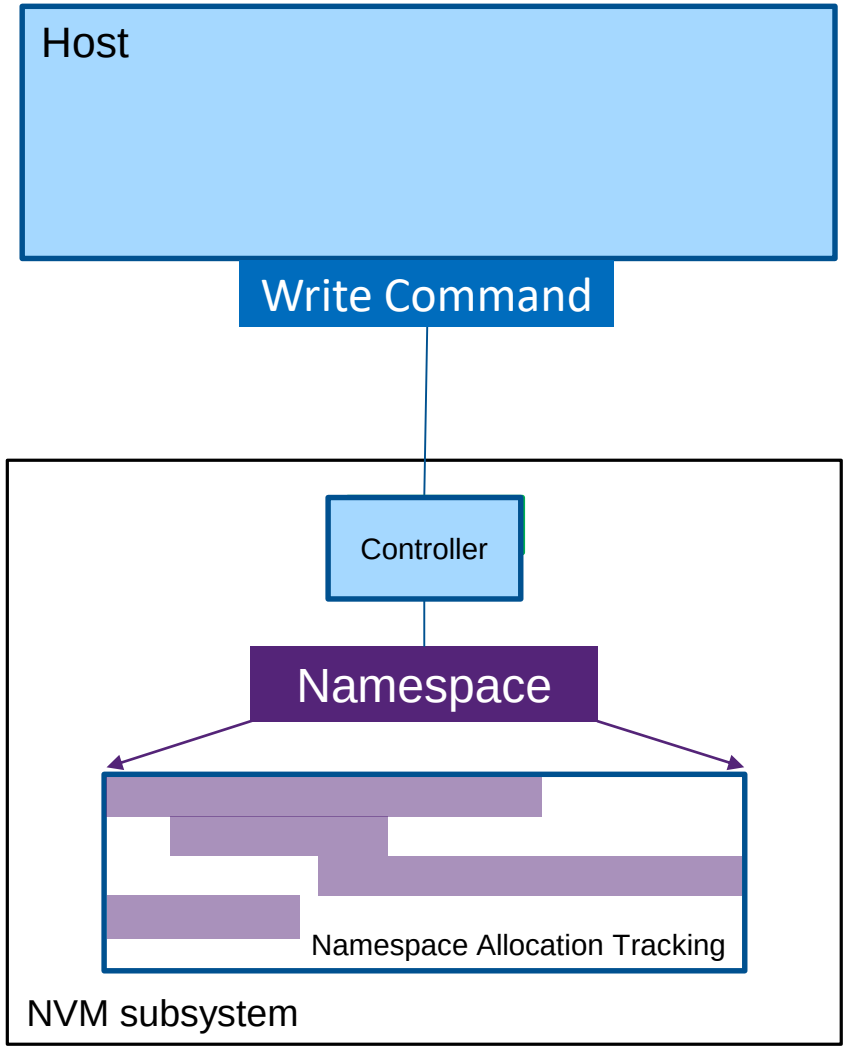
Copying Attached Namespaces from Source to Destination

- A HOST chooses to :
  - Read each LBA of a namespace and copy them to the destination
  - Read the allocated LBAs of a namespace and copy them to the destination
    - TP4165a Tracking LBA Allocation with Granularity
    - Ratified 9/11/2023
    - Included in NVM Express® NVM Command Set Specification Release 1.2

Live Migration consists of multiple Entities

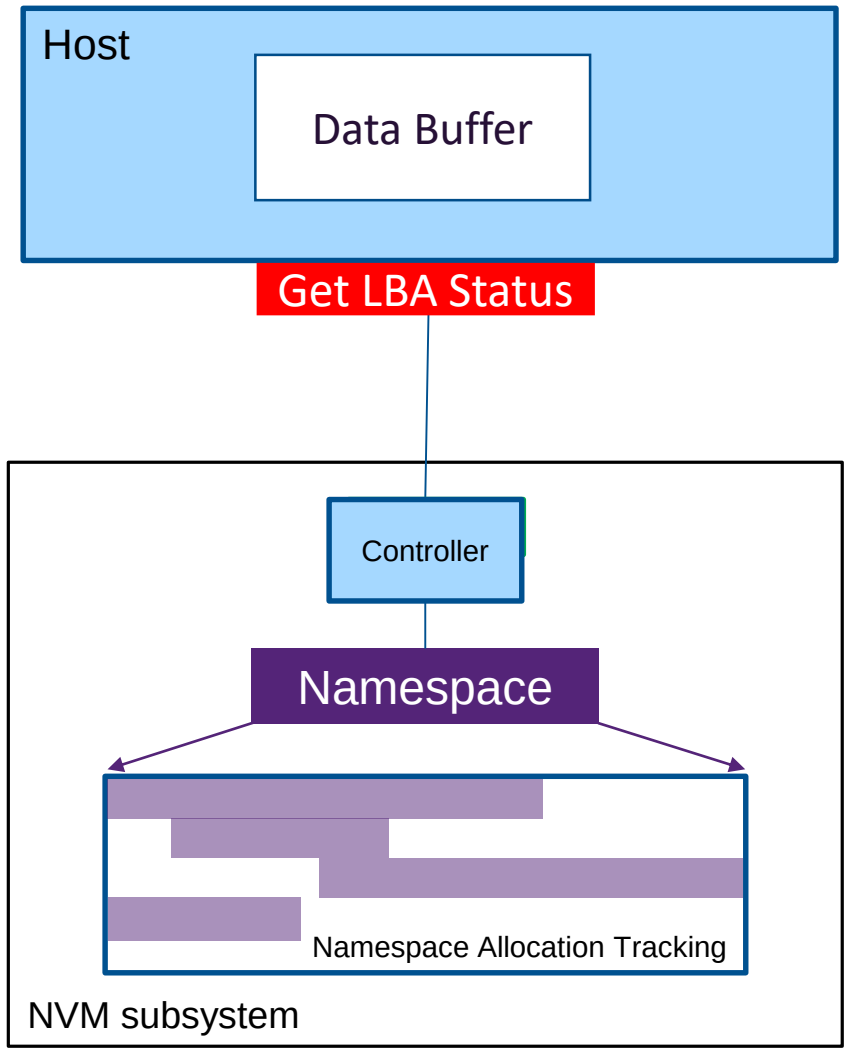


NVMe™ TP4165 Tracking LBA Allocation with Granularity



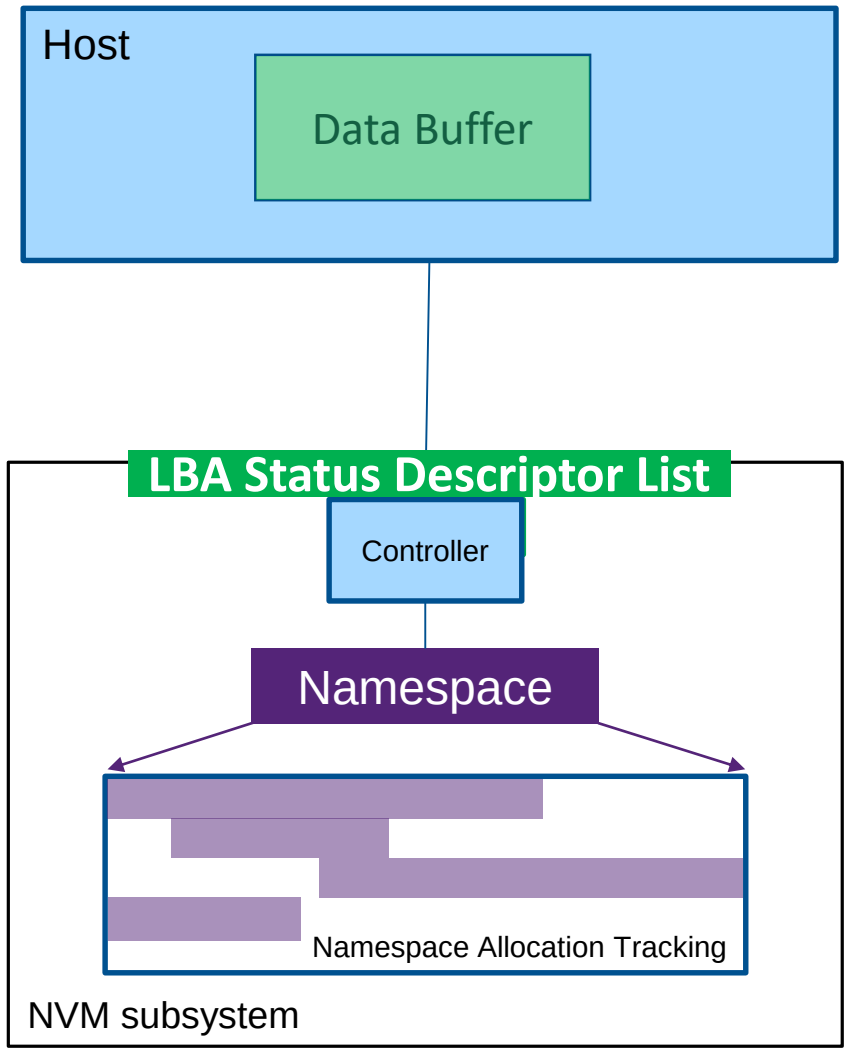
- For each namespace the NVMe controller tracks allocated LBAs at the granularity specified by the **Tracked LBA Allocation Granularity** (TLBAAG) field in the I/O Command Set Specific Identify Namespace data structure for the NVM Command Set
  - All LBAs within the reported granularity are tracked as being allocated if 1 LBA within the granularity is allocated
  - All LBAs within the reported granularity are tracked as being deallocated if all LBAs within the granularity are deallocated

NVMe™ TP4165 Tracking LBA Allocation with Granularity



- For each namespace the controller tracks allocated LBAs at the granularity specified by the **Tracked LBA Allocation Granularity** (TLBAAG) field in the I/O Command Set Specific Identify Namespace data structure for the NVM Command Set
  - All LBAs within the reported granularity are tracked as being allocated if 1 LBA within the granularity is allocated
  - All LBAs within the reported granularity are tracked as being deallocated if all LBAs within the granularity are deallocated
- The **Action Type** field in the Get LBA Status command was modified to allow a HOST to request the allocated LBAs of a namespace
  - Within a specified range of LBAs
  - Across the entire namespace

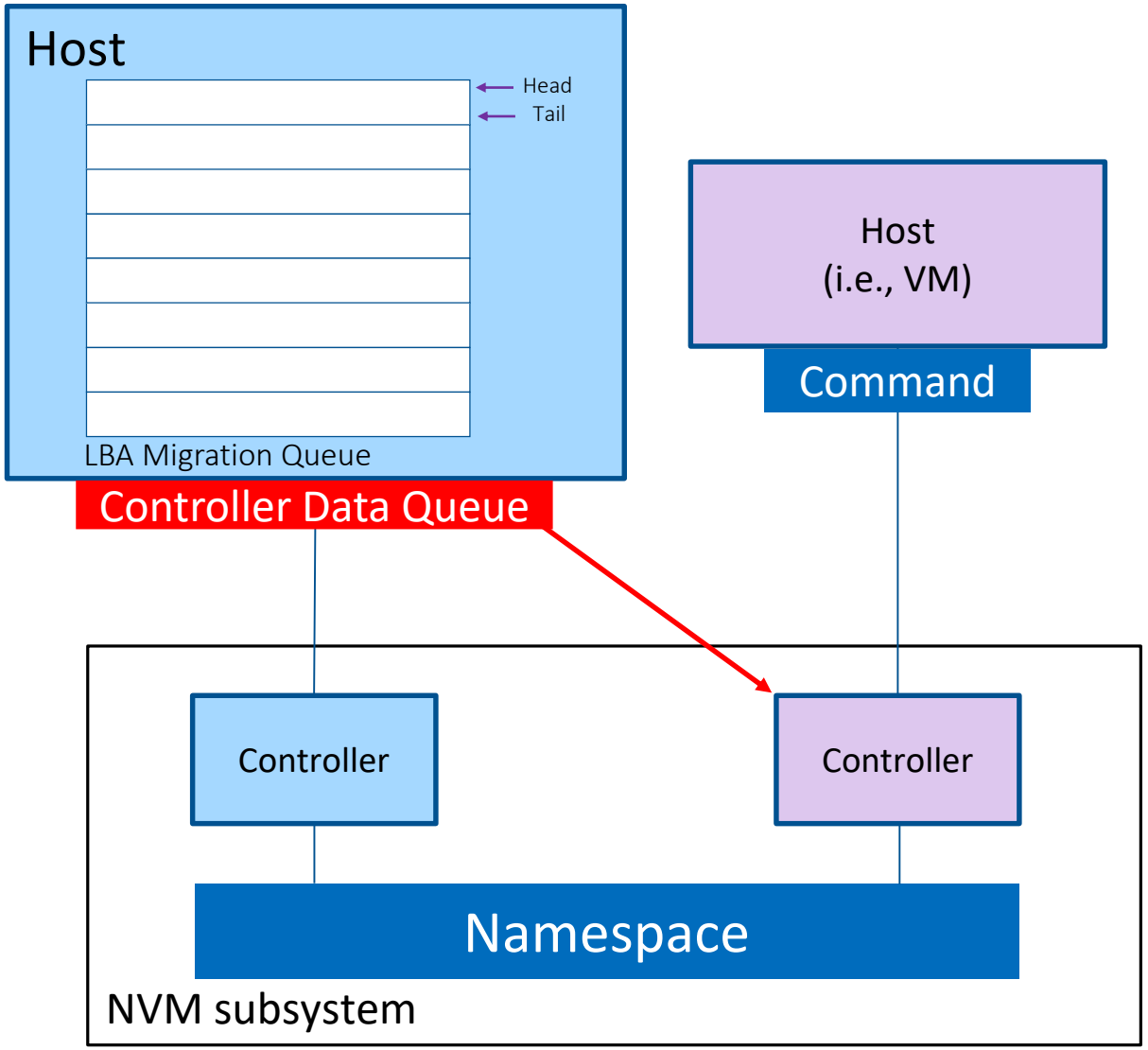
NVMe™ TP4165 Tracking LBA Allocation with Granularity



- For each namespace the controller tracks allocated LBAs at the granularity specified by the **Tracked LBA Allocation Granularity** (TLBAAG) field in the I/O Command Set Specific Identify Namespace data structure for the NVM Command Set
  - All LBAs within the reported granularity are tracked as being allocated if 1 LBA within the granularity is allocated
  - All LBAs within the reported granularity are tracked as being deallocated if all LBAs within the granularity are deallocated
- The **Action Type** field in the Get LBA Status command was modified to allow a HOST to request the allocated LBAs of a namespace
  - Within a specified range of LBAs
  - Across the entire namespace
- Controller returns an **LBA Status Descriptor List data structure**
  - **Number of LBA Status Descriptors**
  - **Completion Condition** to indicate if more data is available to be reported by the controller
  - List of LBA Status Descriptor Entries
    - **Descriptor Starting LBA**
    - **Number of Logical Blocks**

} Aligns to a Read command

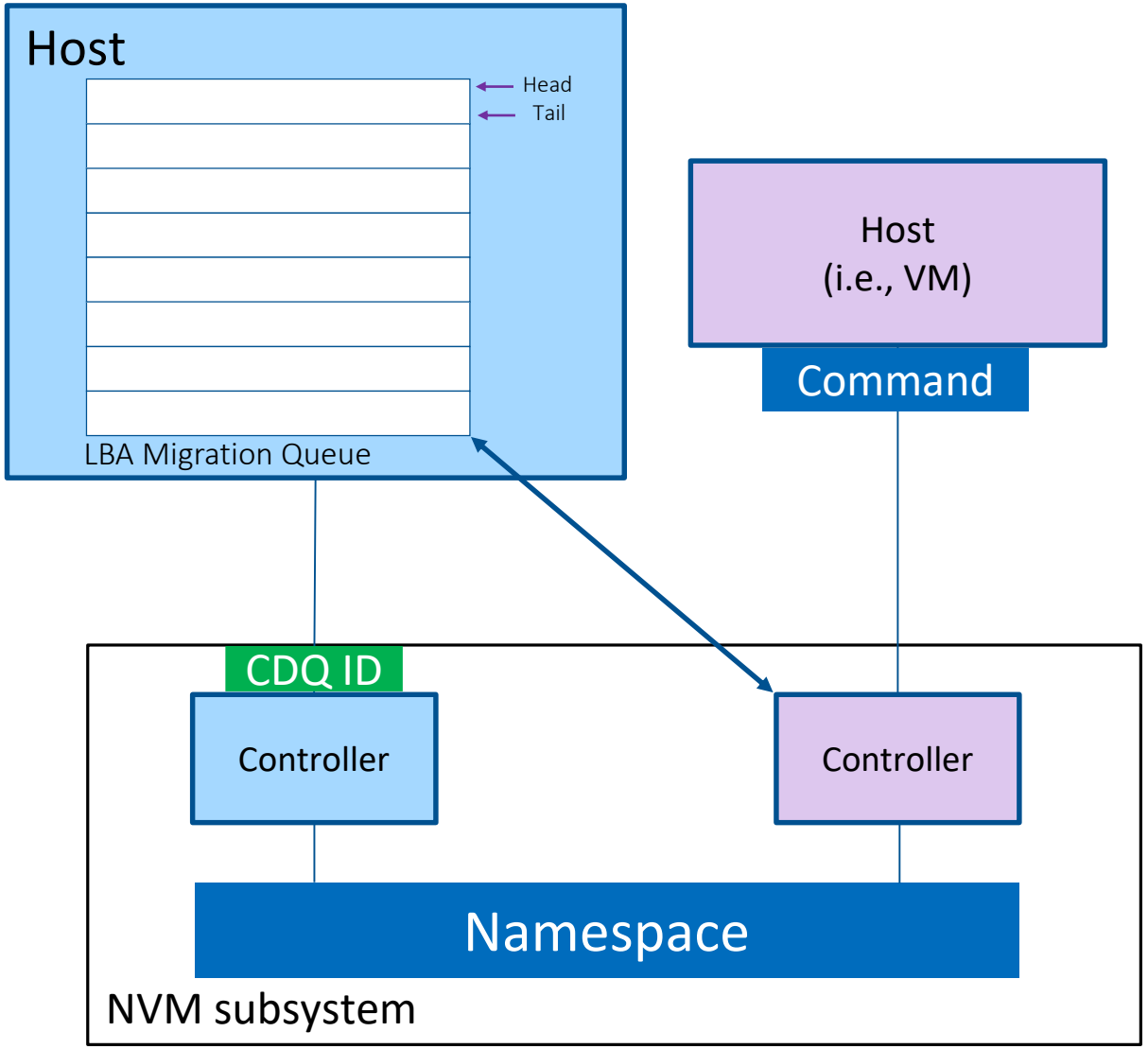
Tracking Modified LBAs



Create an LBA Migration Queue

- Host allocates memory for a LBA Migration Queue
- Host submits a Controller Data Queue command to a controller specifying:
  - Controller Identifier of a controller to track LBA modifications by that controller
  - Log User Data Changes Management Operation
  - Host Memory Address
  - Dword Size

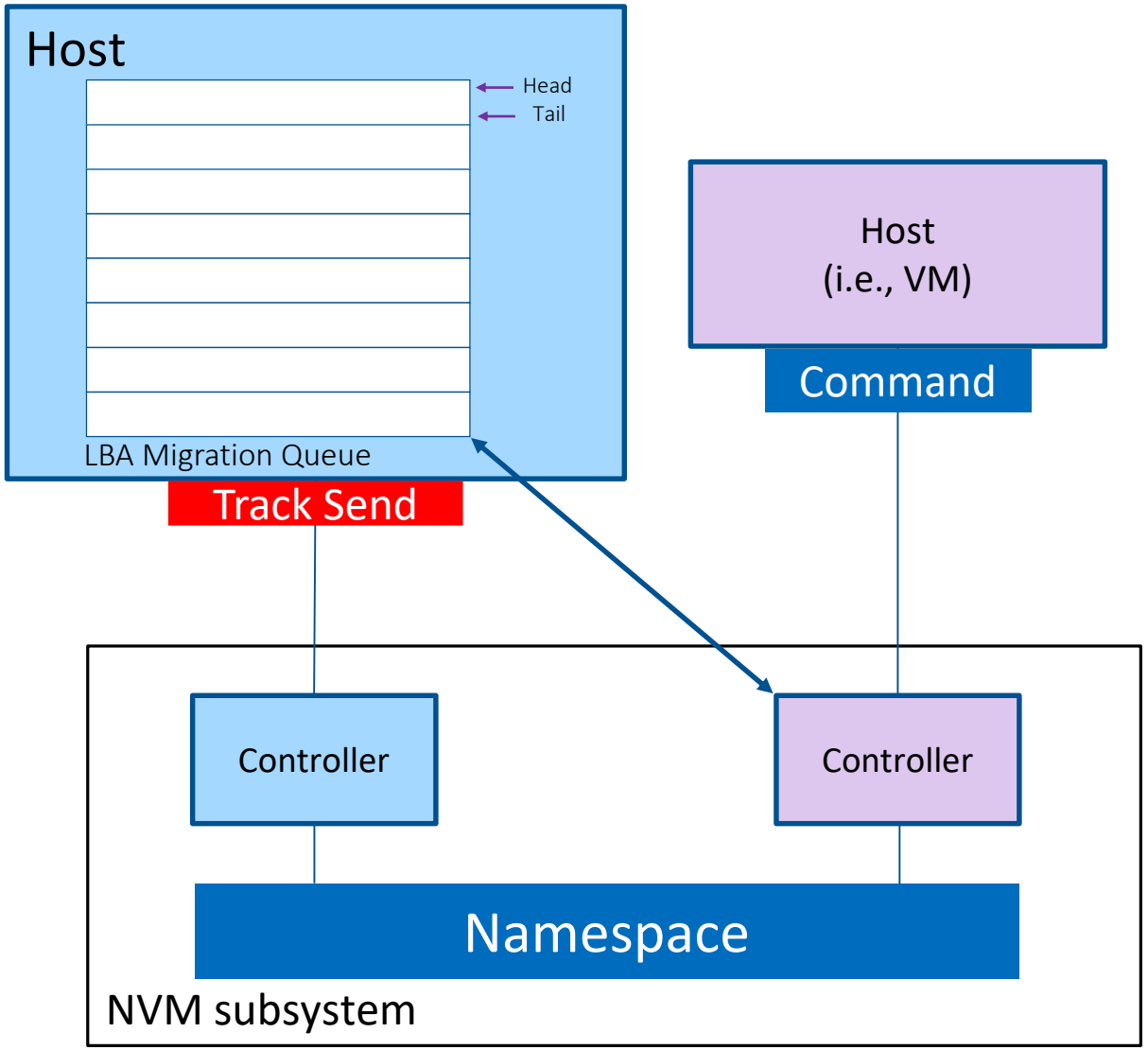
Tracking Modified LBAs



Create an LBA Migration Queue

- Host allocates memory for a LBA Migration Queue
- Host submits a Controller Data Queue command to a controller specifying:
  - Controller Identifier of a controller to track LBA modifications by that controller
  - Log User Data Changes Management Operation
  - Host Memory Address
  - Dword Size
- Controller returns a Controller Data Queue Identifier

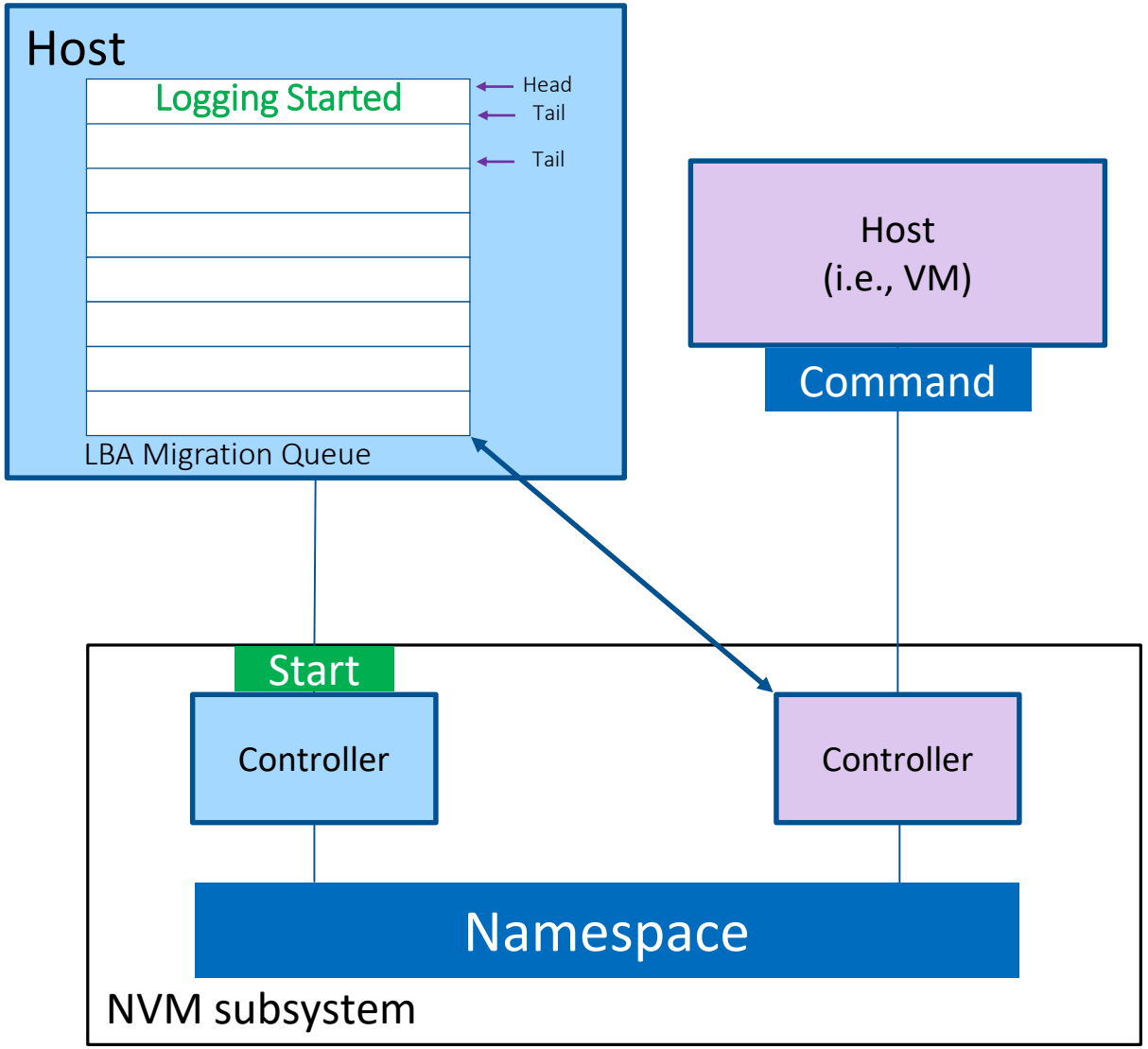
Tracking Modified LBAs



Start logging LBA modification into the LBA Migration Queue

- Host submits a Track Send command to a controller specifying:
  - Log User Data Management Operation
  - Controller Data Queue Identifier of the LBA Migration Queue
  - Start logging modified LBAs
    - LBA Migration Queue entries are allowed to posted after CQE for command is posted

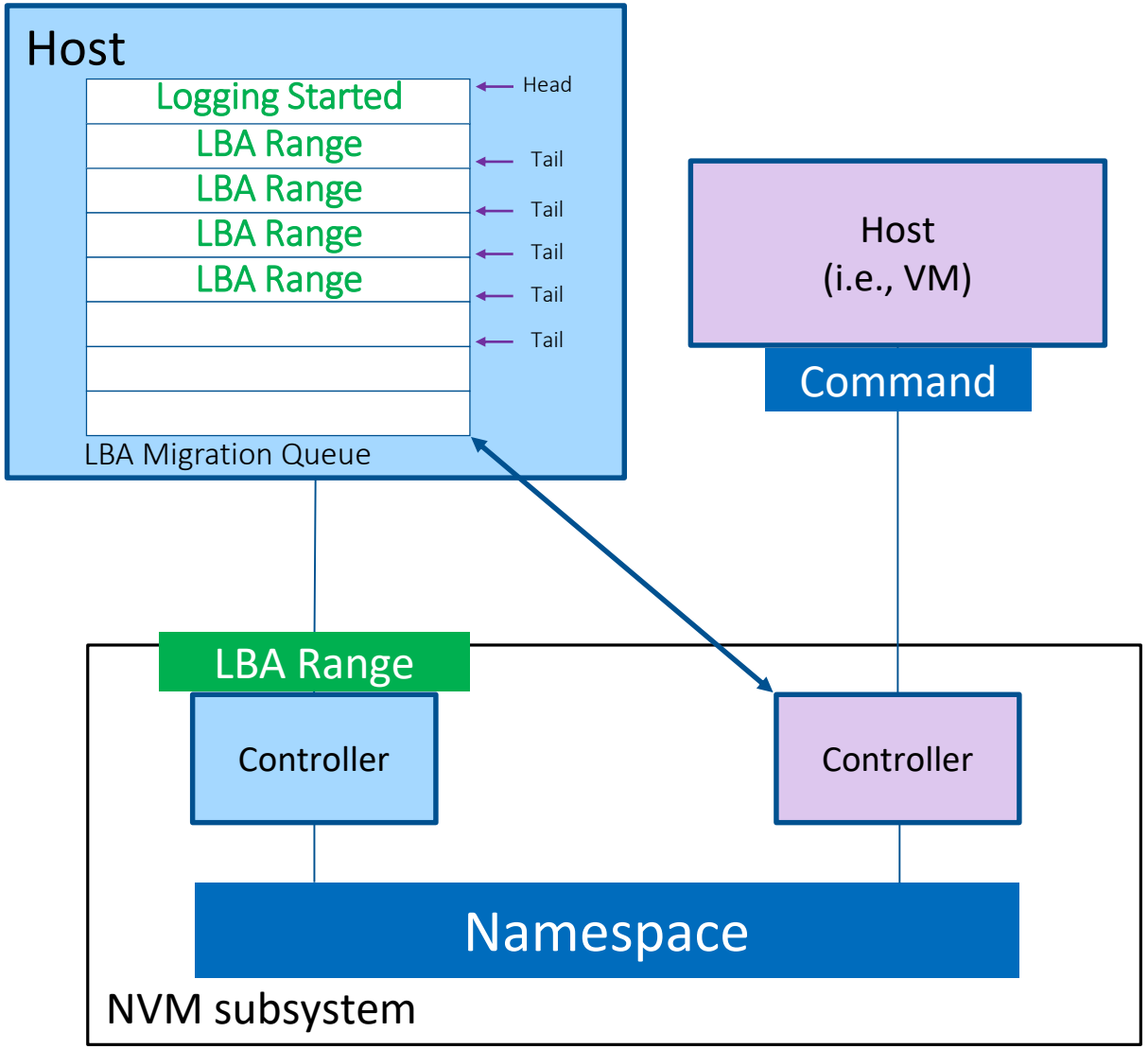
Tracking Modified LBAs



Start logging LBA modification into the LBA Migration Queue

- Host submits a Track Send command to a controller specifying:
  - Log User Data Management Operation
  - Controller Data Queue Identifier of the LBA Migration Queue
  - Start logging modified LBAs
    - LBA Migration Queue entries are allowed to posted after CQE for command is posted
- The controller that created the LBA Migration Queue to post entries specifying:
  - Changes logging
    - Start, stop, full, controller suspended
  - LBA modifications by the controller associated to the LBA migration queue

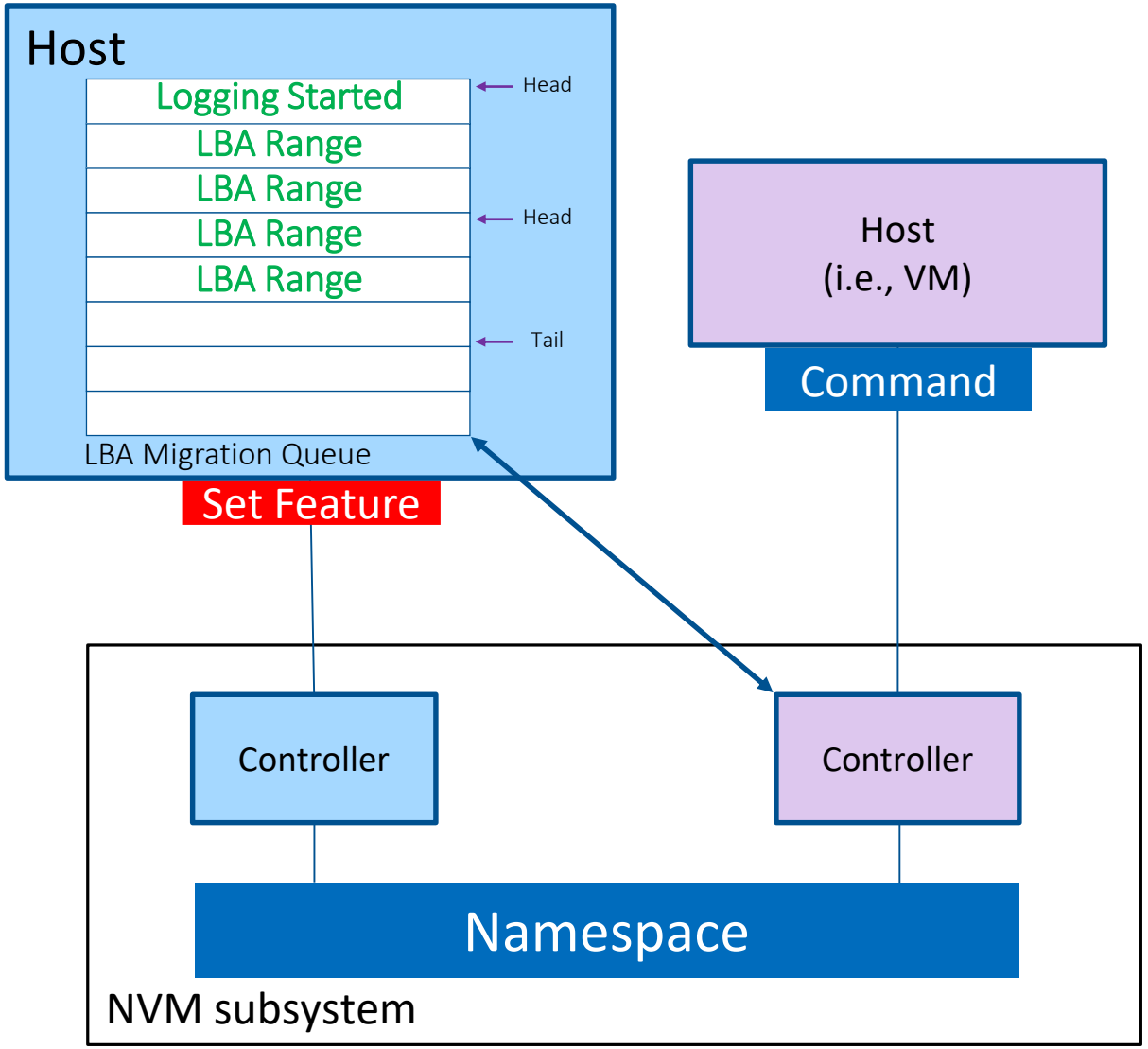
Tracking Modified LBAs



Start logging LBA modification into the LBA Migration Queue

- Host submits a Track Send command to a controller specifying:
  - Log User Data Management Operation
  - Controller Data Queue Identifier of the LBA Migration Queue
  - Start logging modified LBAs
    - LBA Migration Queue entries are allowed to posted after CQE for command is posted
- The controller that created the LBA Migration Queue to post entries specifying:
  - Changes logging
    - Start, stop, full, controller suspended
  - LBA modifications by the controller associated to the LBA migration queue

Tracking Modified LBAs



Host processes the entries in the LBA Migration Queue

- Host submits a Set Features command specifying:
  - Controller Data Queue feature
  - Controller Data Queue Identifier of the LBA Migration Queue
  - Head Pointer Location
  - Optional Tail Pointer trigger to cause a Controller Data Queue Tail Pointer event
- New One-Shot Event Type
  - Host ARMs each instance of the event
    - Host handles event in flight if re-arming
  - Removal of pending events
    - Controller Data Queue feature
      - Set Features command specifying the Controller Data Queue feature
      - Track Send command specifying the LBA Migration Queue

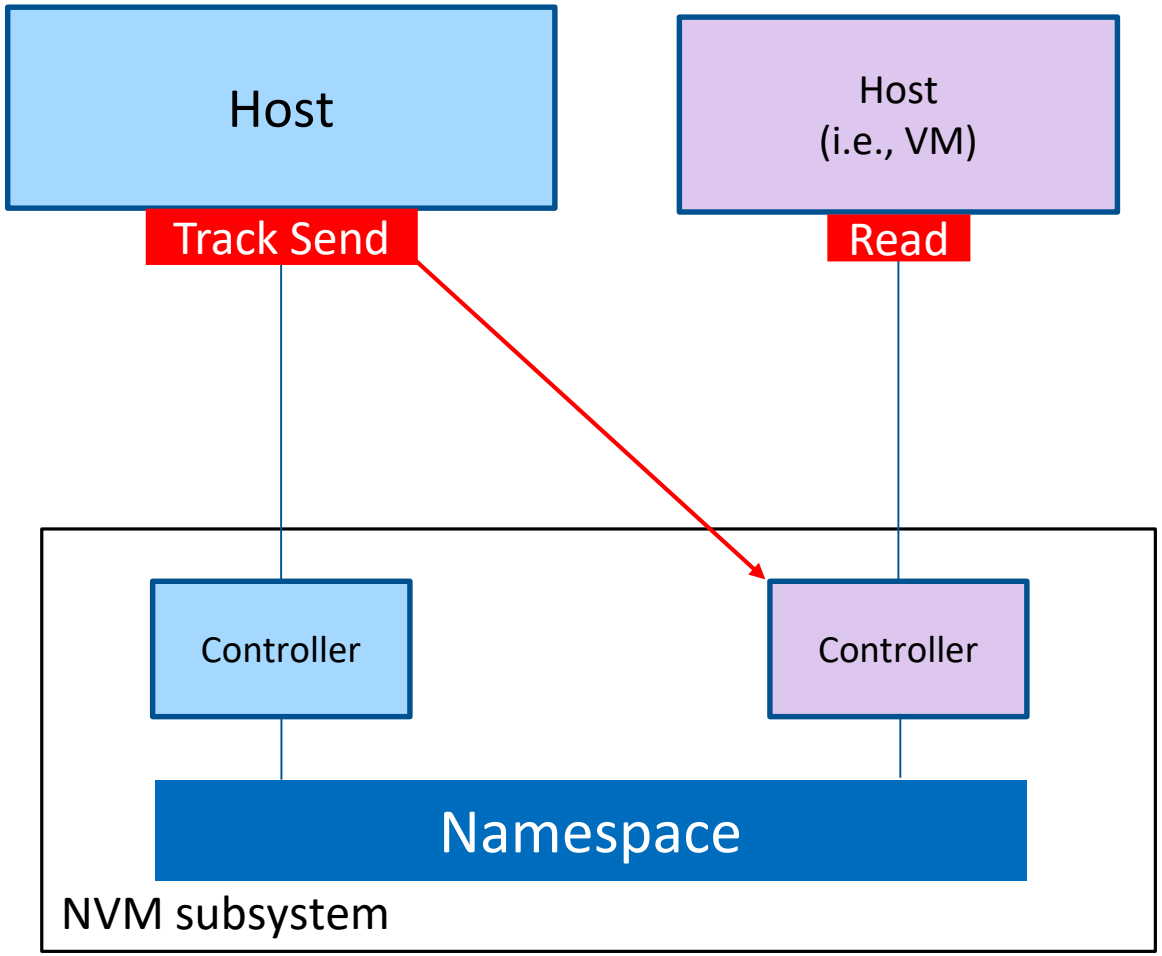
LBA Migration Queue Entry

- LBA Queue Entry only defined by the NVM Command Set
- LBA Queue Entries specify the **Entry Sequence Attribute**
  - The first entry since tracking LBAs was enabled or the controller resumed operation while tracking LBAs was enabled
    - Does not have to be posted into the queue before the completion queue entry is posted by the controller for the command that enabled tracking LBAs
  - The last entry due to
    - A host request
    - Controller being suspended
    - Queue becoming FULL
      - Causes tracking to be disabled, so hosts need to size the queue appropriately and remove entries form the queue in the timely manner
  - An entry that is not the first and last
- A single namespace indicating LBA range or Entire Namespace, Phase Tag, and deallocation only entry

Affects of Resets on Tracking Modified LBAs

- All tracking by a controller of modified LBAs ceases on that controller as a result of a Controller Level Reset
  - Every LBA Migration Queue is deleted by the MMC

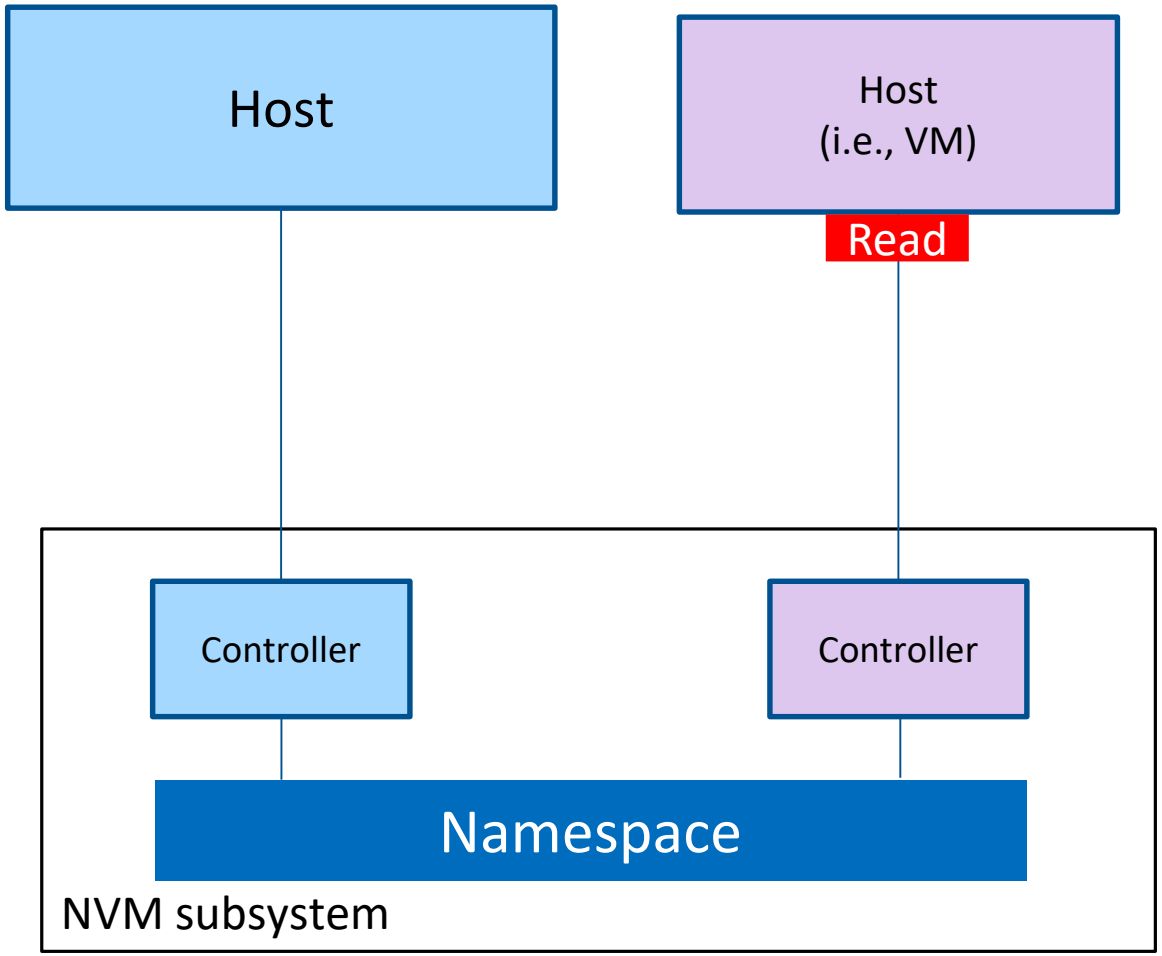
Tracking Modified Host Memory



Start tracking the host memory modifications by another controller due to processing NVMe® commands

- Host submits a Track Send command to a controller specifying:
  - Controller Identifier of a controller to track host memory modifications by that controller
  - Memory ranges

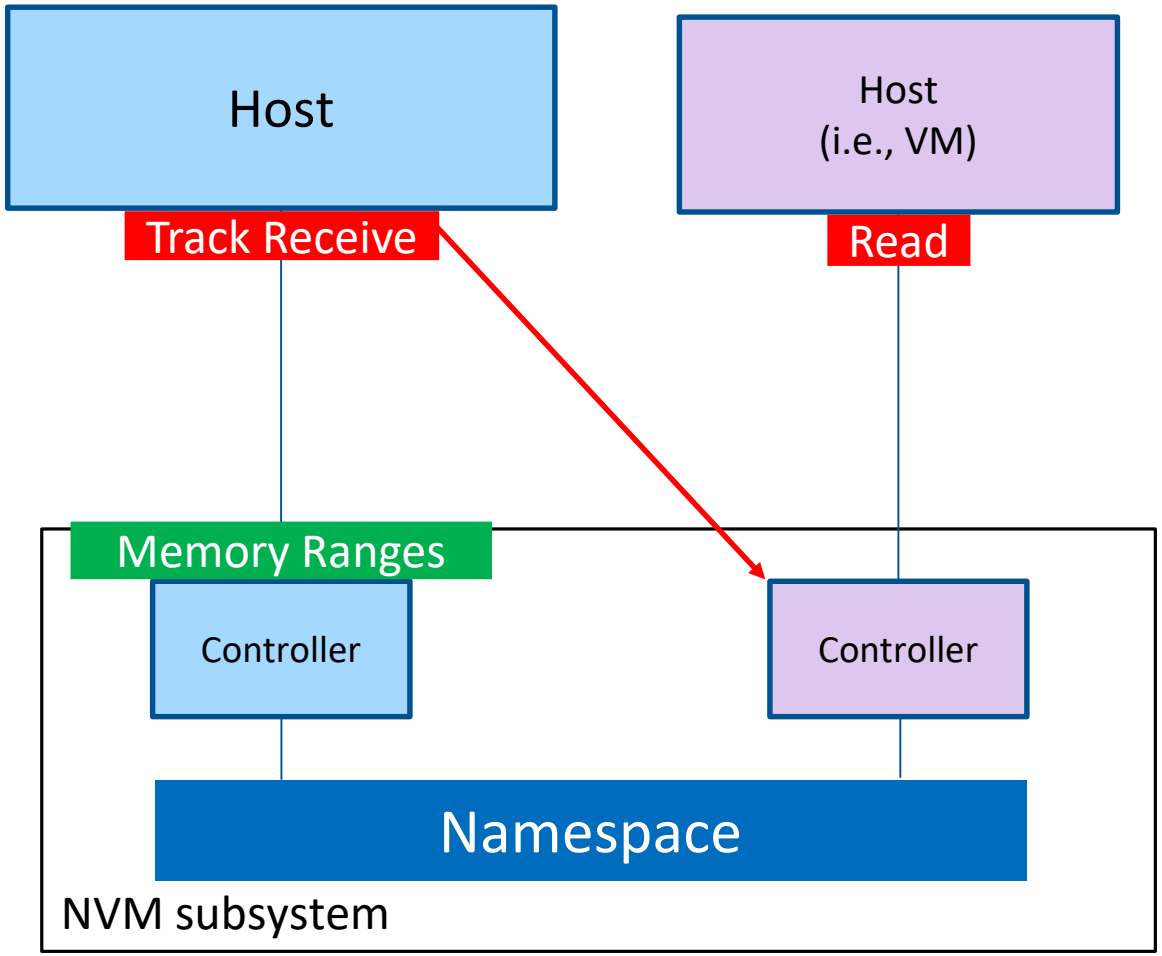
Tracking Modified Host Memory



Start tracking the host memory modifications by another controller due to processing NVMe® commands

- Host submits a Track Send command to a controller specifying:
  - Controller Identifier of a controller to track host memory modifications by that controller
  - Memory ranges
- As controller processes command
  - Remembers all host memory modifications within those Memory ranges

Tracking Modified Host Memory



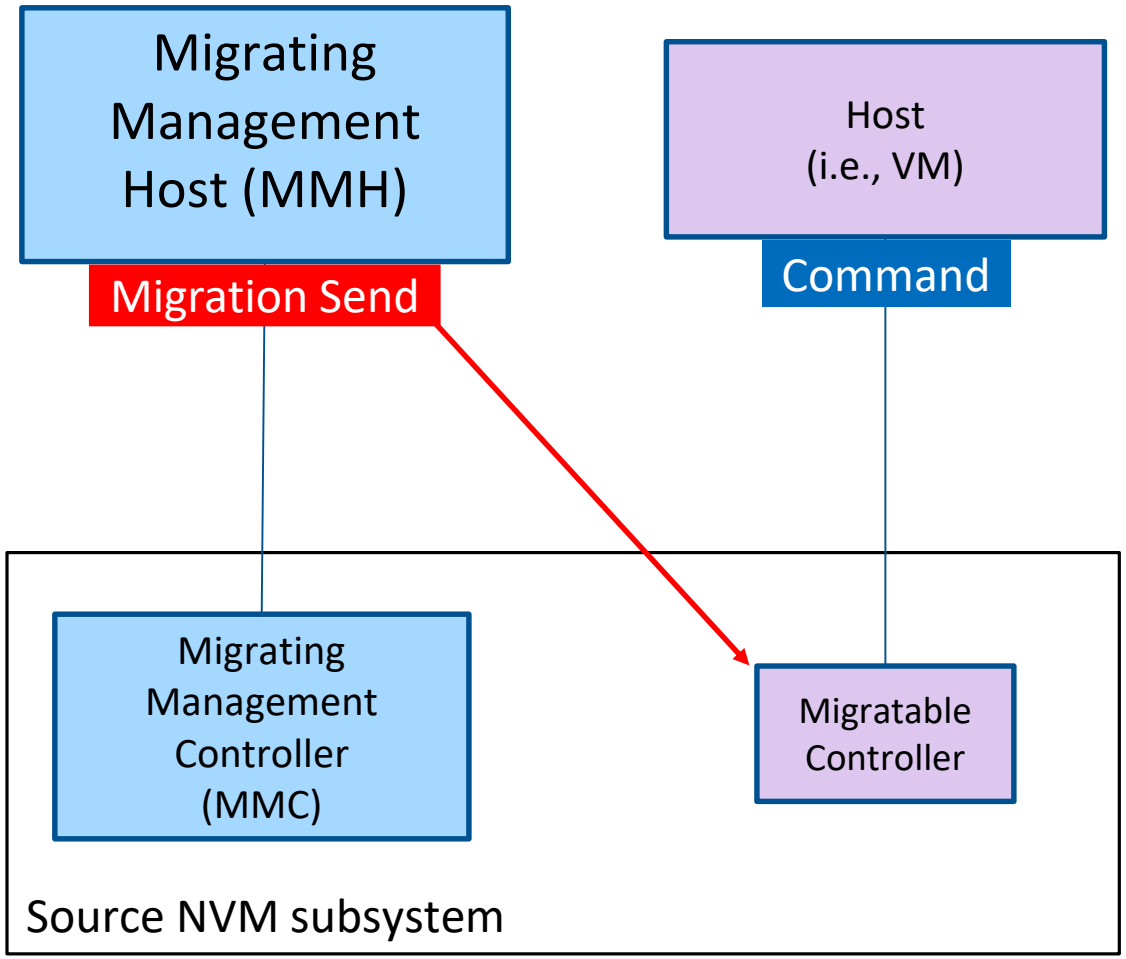
Obtain the host memory medications

- Host submits a Track Receive command to a controller specifying:
  - Tracked Memory Ranges Management Operation
  - Size of the returned data
  - Controller Identifier of a controller to track host memory modifications by that controller
- the controller returns:
  - Suspended state of the specified controller
  - If more data is ready to be reported
  - Zero or more modified host memory ranges

Tracking Modified Host Memory

- All tracking by a controller of modified host memory ceases on that controller as a result of a Controller Level Reset

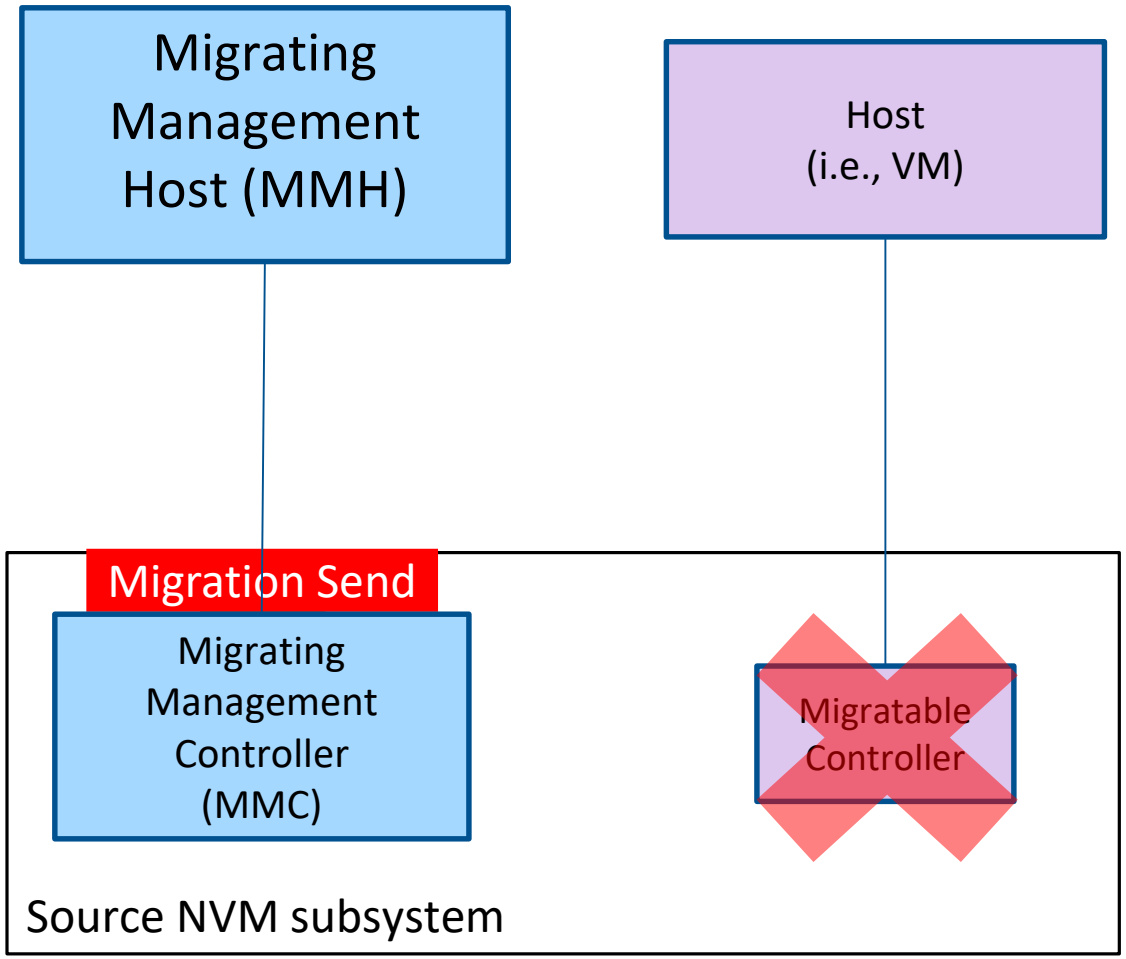
Suspending a Migratable Controller



Migrating Management Host (MMH) suspends the controller to be migrated

- Submit a Migration Send command to a Migrating Management Controller (MMC) specifying:
  - Controller Identifier of the Migratable Controller
  - Suspend Management Operation
- Migratable Controller:
  - Halts fetching commands
  - Completes the processing of all previously fetched commands

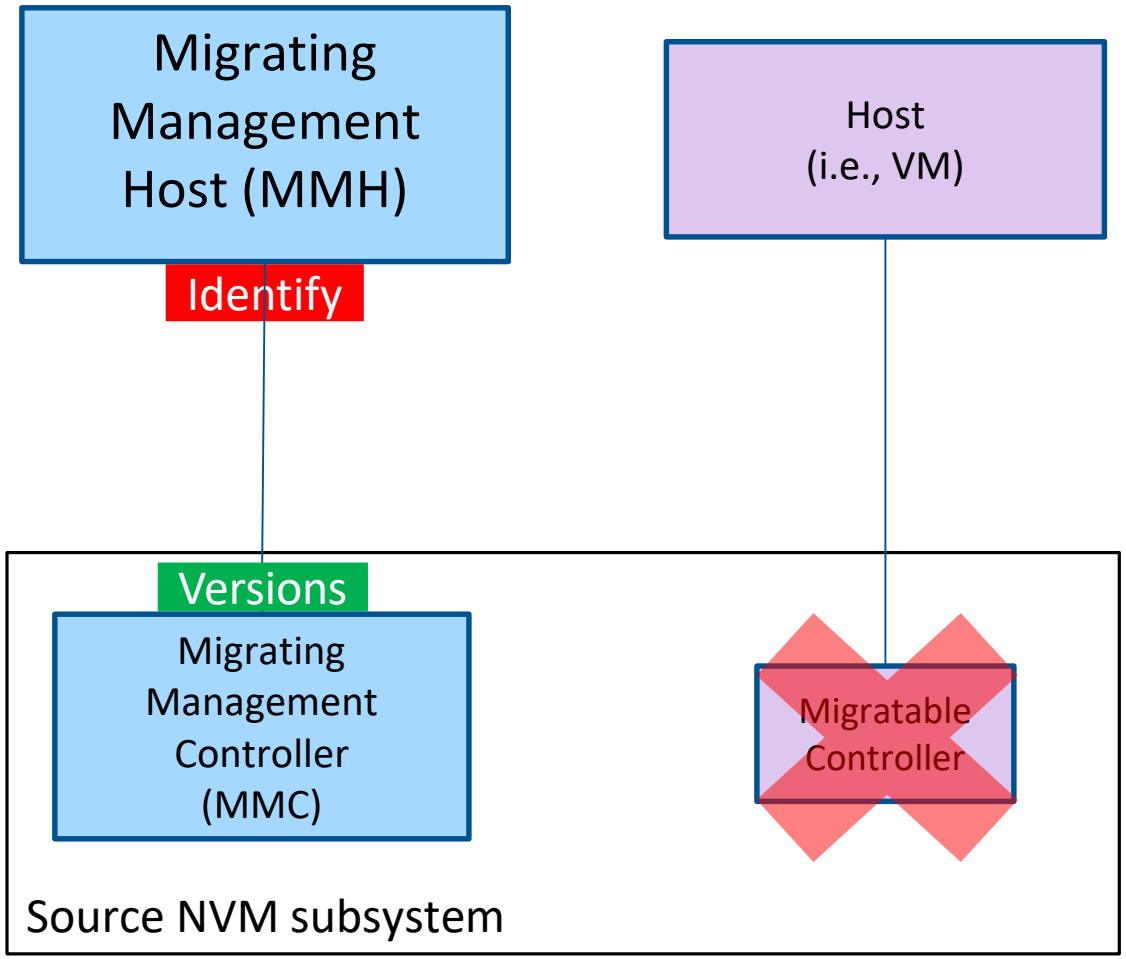
Suspending a Migratable Controller



Migrating Management Host (MMH) suspends the controller to be migrated

- Submit a Migration Send command to a Migrating Management Controller (MMC) specifying:
  - Controller Identifier of the Migratable Controller
  - Suspend Management Operation
- Migratable Controller:
  - Halts fetching commands
  - Completes the processing of all previously fetched commands

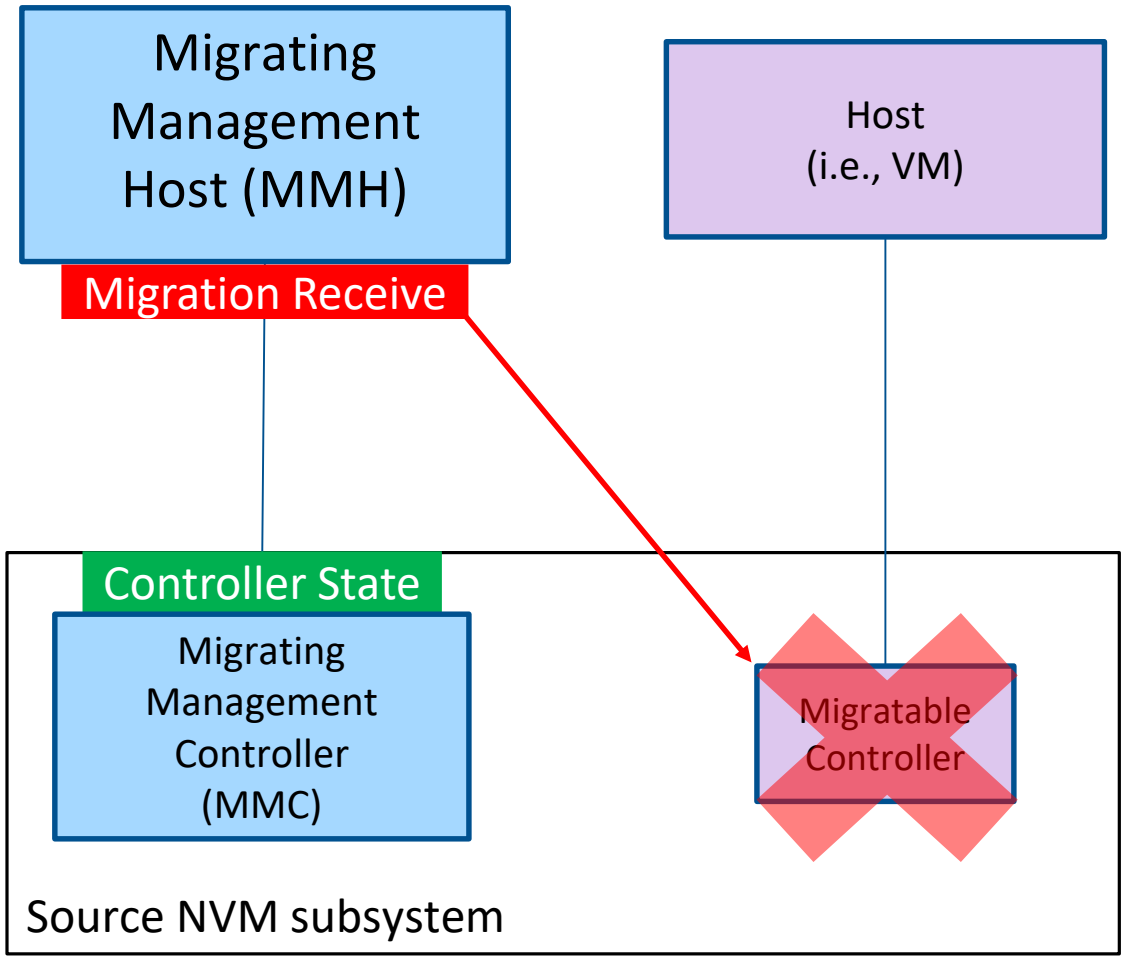
Getting the Migratable Controller State



MMH obtains the controller state of the suspended controller being migrated

- Submit an Identify command specifying a Controller Namespace Structure (CNS) value 20h
- MMC returns supported versions:
  - NVMe® technology defined state (#)
  - Vendor specific state (UUID)

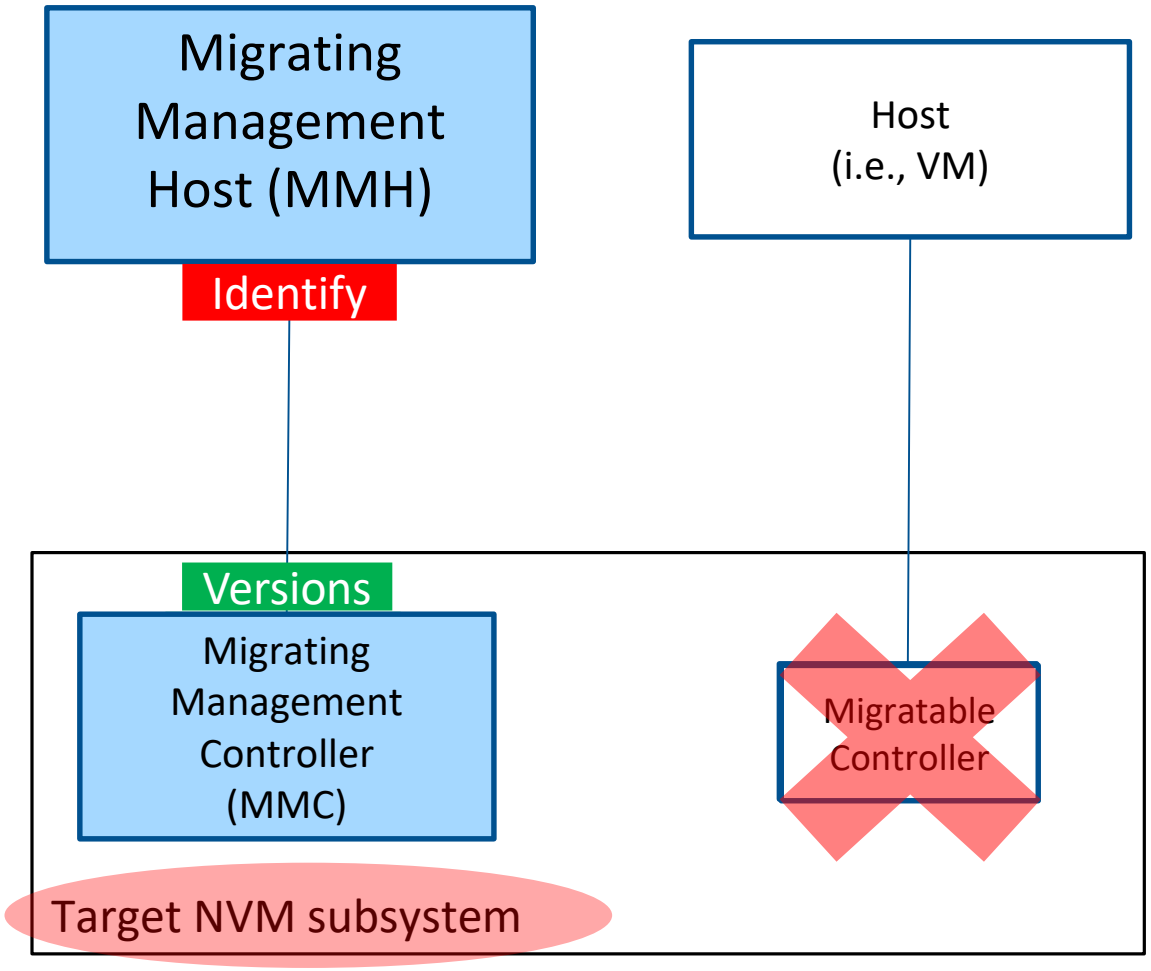
Getting the Migratable Controller State



MMH obtains the controller state of the suspended controller being migrated

- Submit an Identify command specifying a Controller Namespace Structure (CNS) value 20h
- MMC returns supported versions:
  - NVMe® technology defined state (#)
  - Vendor specific state (UUID)
- Submits one or more Migration Receive Send commands to a MMC specifying:
  - Controller Identifier of the Migratable Controller
  - Specific controller state versions
  - Offset
  - Size
- MMC returns the controller state

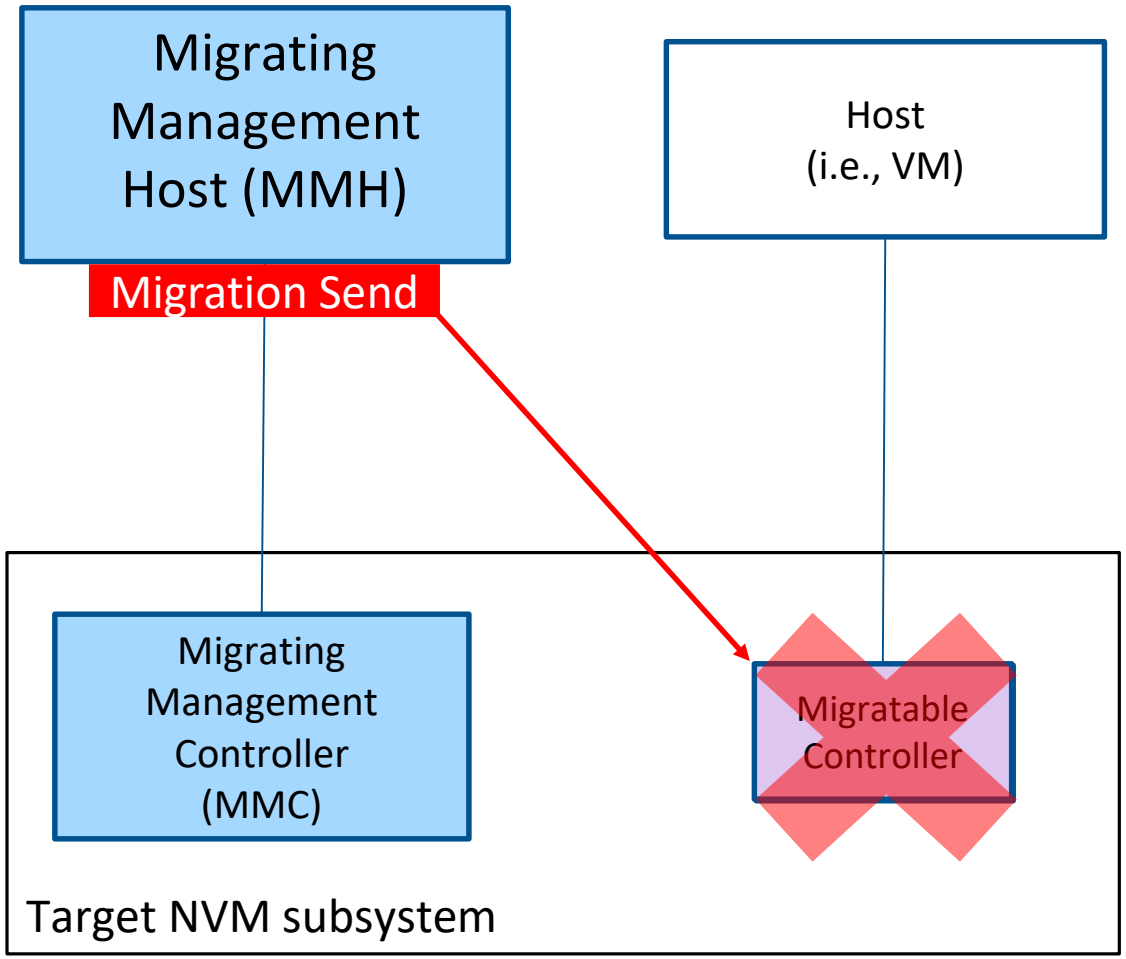
Setting the Migratable Controller State



MMH puts the controller state of the suspended controller being migrated into a different controller

- Submit an Identify command specifying a CNS value 20h
- MMC returns controller state versions:
  - NVMe® technology defined state
  - Vendor specific state

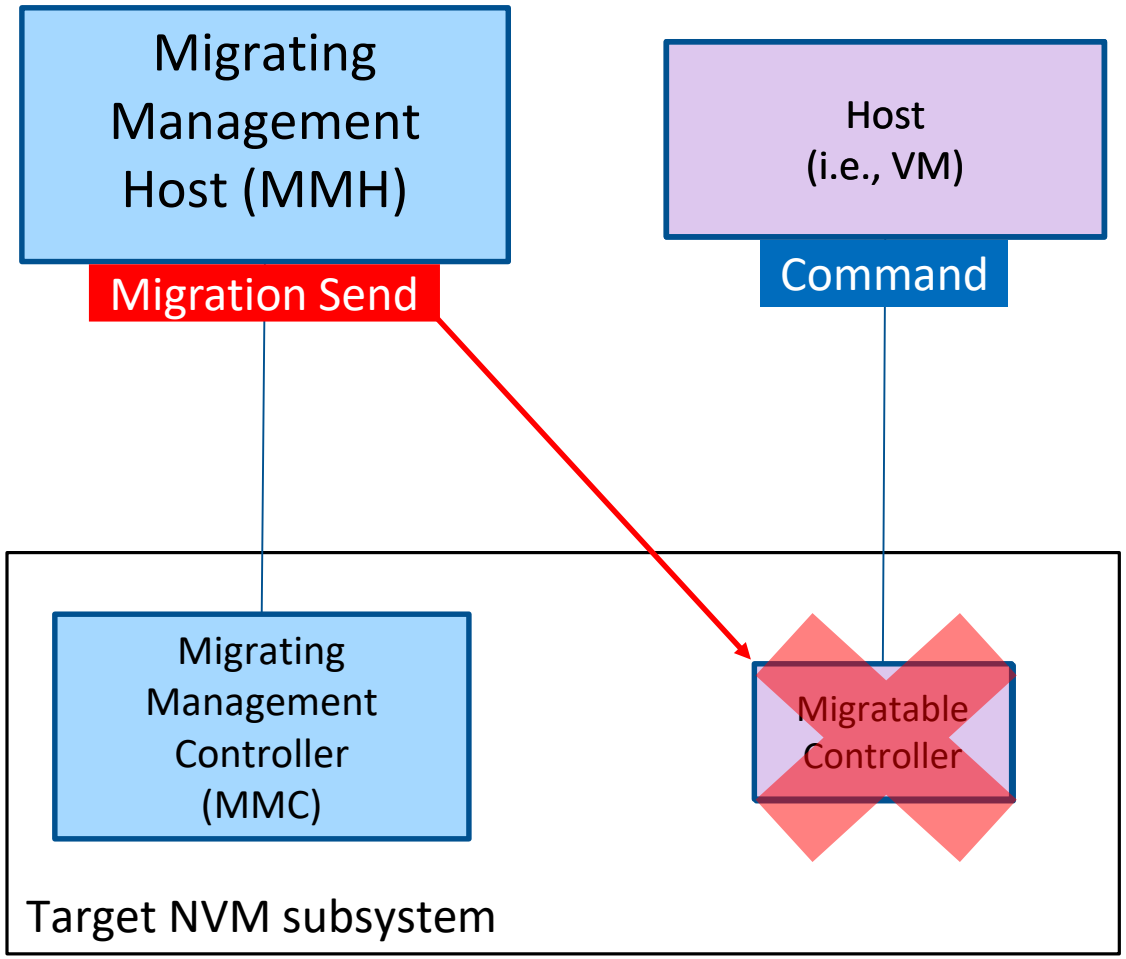
Setting the Migratable Controller State



MMH puts the controller state of the suspended controller being migrated into a different controller

- Submit an Identify command specifying a CNS value 20h
- MMC returns controller state versions:
  - NVMe® technology defined state
  - Vendor specific state
- Submits one or more Migration Send commands to a MMC specifying:
  - Controller Identifier of the different controller
  - Get Controller State Management Operation
  - Specific controller state versions
  - Order of commands
    - First, middle, last

Resuming the Migratable Controller



MMH responsibility to migrate the VM

MMH resumes the controller that has migrated

- Submit a Migration Send command to a Migrating Management Controller (MMC) specifying:
  - Controller Identifier of the Migratable Controller
  - Resume Management Operation
- Migratable Controller:
  - Starts fetching commands

Miscellaneous Information

- Identify Controller data structure
  - Controller Support
    - This controller is a MMC (**HMLMS** bit)
    - Support for Track User Data changes (i.e. LBA Migration Queue)
    - Support for Track host memory changes
  - Controller Restrictions
    - The number of Memory Range Tracking Descriptors (controller and NVM subsystem)
    - The Memory Range Tracking granularity (minimum and maximum)
    - Memory Range Tracking lengths in power of 2
    - Maximum number of LBA Migration Queues (controller and NVM subsystem)
    - Number of separated memory ranges defining the LBA Migration Queue
    - Number of PRPs allowed to define a LBA Migration Queue
- Controller Data Queue added to the FID Scope field in the Feature Identifiers Supported and Effects log page
- A sanitize operation is prohibited is a controller is suspended in the NVM subsystem

NVMe™ Live Migration Summary

- New Commands
  - Controller Data Queue
    - Created and Delete a LBA Migration Queue
  - Track Send/Receive
    - Manage the logging of LBA changes in the specified controller
    - Manage the tracking of host memory changes due to a controller processing commands
  - Migration Send/Receive
    - Manage the suspending a controller, getting and setting controller state, and resuming a controller
- New Features
  - Controller Data Queue
- New Identify command CNS
  - Obtain the supported NVMe and Vendor Specific Controller State versions

Please take a moment to rate this session.

Your feedback is important to us.

