NVMe™ and PCIe SSDs
NVMe™ Management Interface

Peter Onufryk
Sr. Director, Product Development
PMC-Sierra

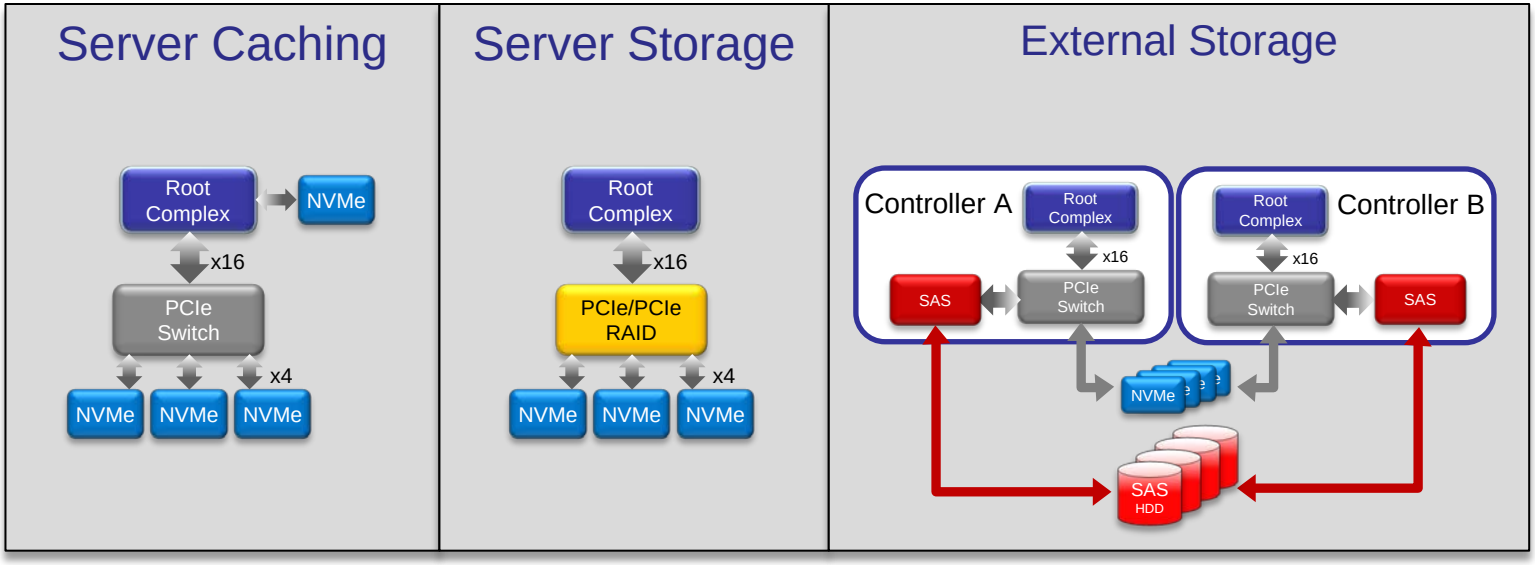
Austin Bolen
Storage Development Principal Engineer
Dell

Special thanks to the NVMe™ Management Interface Workgroup members for contributions & support.

Agenda

- NVMe Management Ecosystem
- In-band vs Out-of-Band Management
- NVMe Out-of-Band Management Stack Overview
  - Transport Layer (MCTP)
  - Protocol Layer (NVMe Management Command Set)
- NVMe Device
  - Management Architectural Model
  - Command Processing
- Mgmt. Controller/Host Communication
- Summary

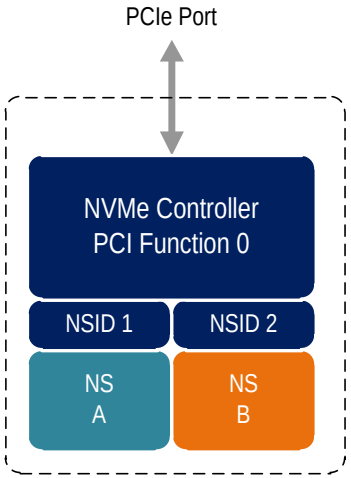
NVMe Storage Device Management



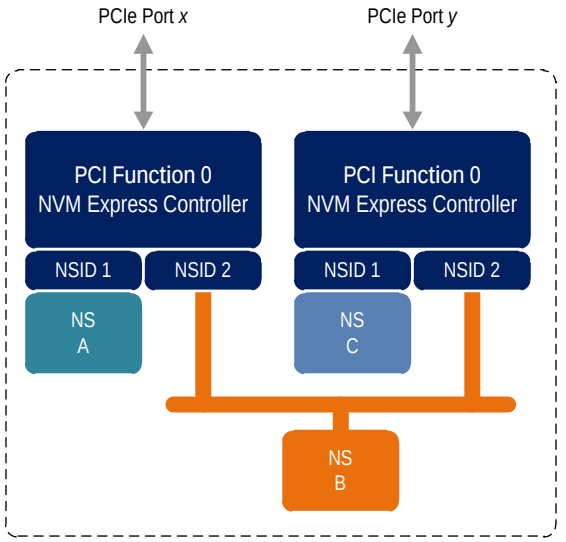
- Example Pre-boot Management
  - Inventory, Power Budgeting, Configuration, Firmware Update
- Example Out-of-Band Management During System Operation
  - Health Monitoring, Power/Thermal Management, Firmware Update, Configuration

NVMe Architecture (review)

- **NVM Subsystem** - one or more controllers, one or more namespaces, one or more PCI Express ports, a non-volatile memory storage medium, and an interface between the controller(s) and non-volatile memory storage medium

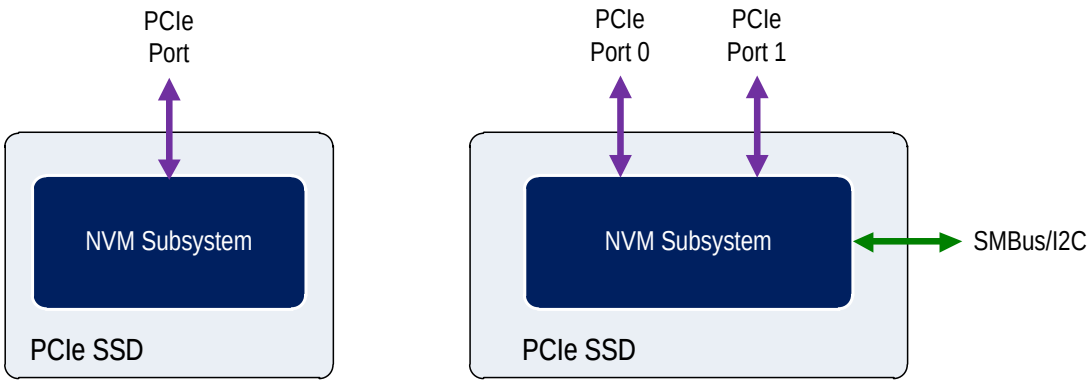


NVM Subsystem with One Controller and One Port



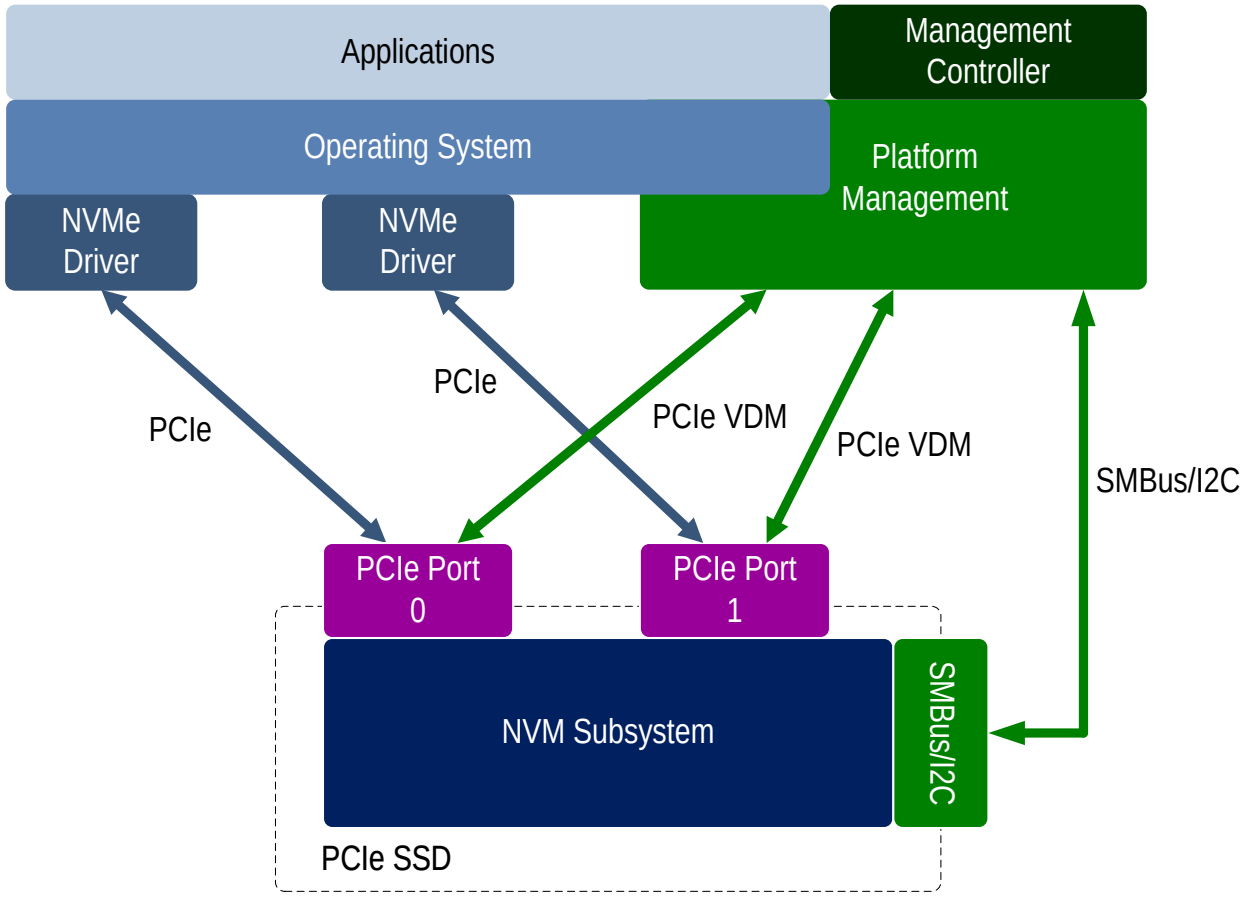
NVM Subsystem with Two Controllers and Two Ports

NVMe Storage Devices

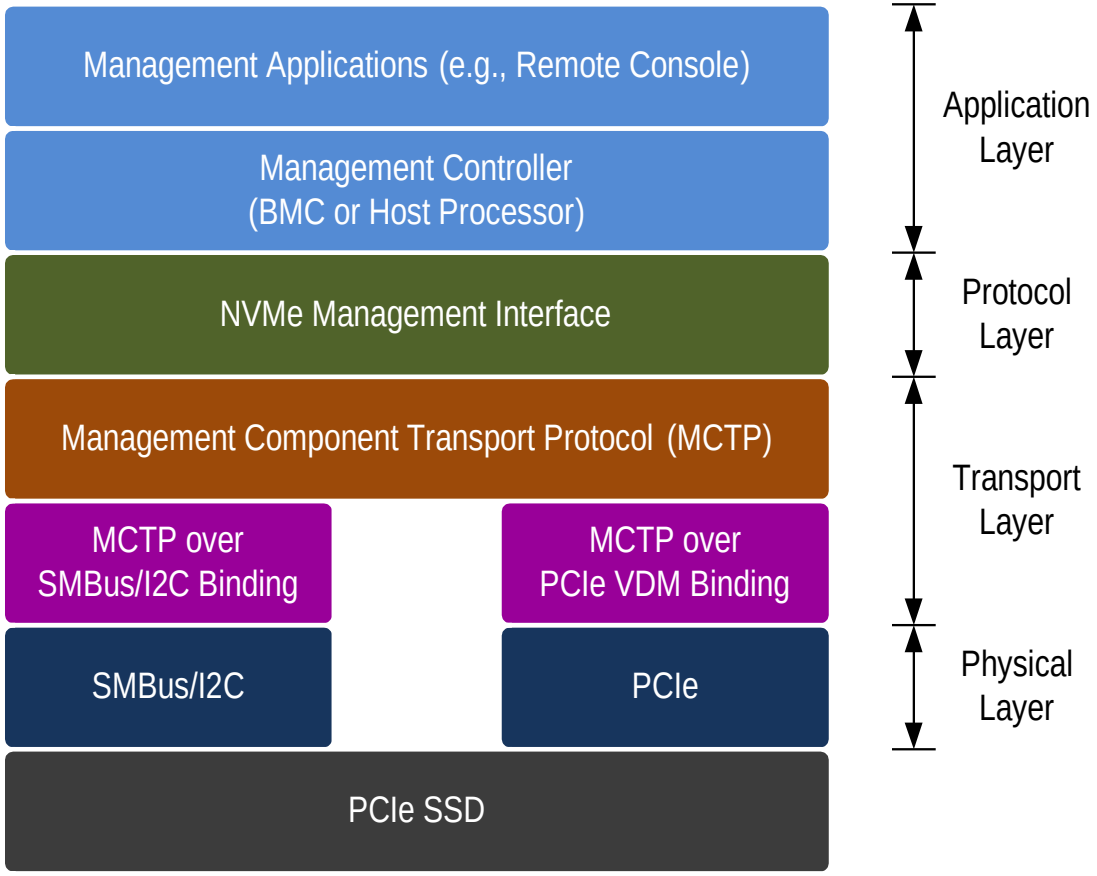


- An NVMe Storage Device consists of one NVM Subsystem with
  - One or more PCIe ports
  - An optional SMBus/I2C interface

Driver vs. Out-of-Band Management



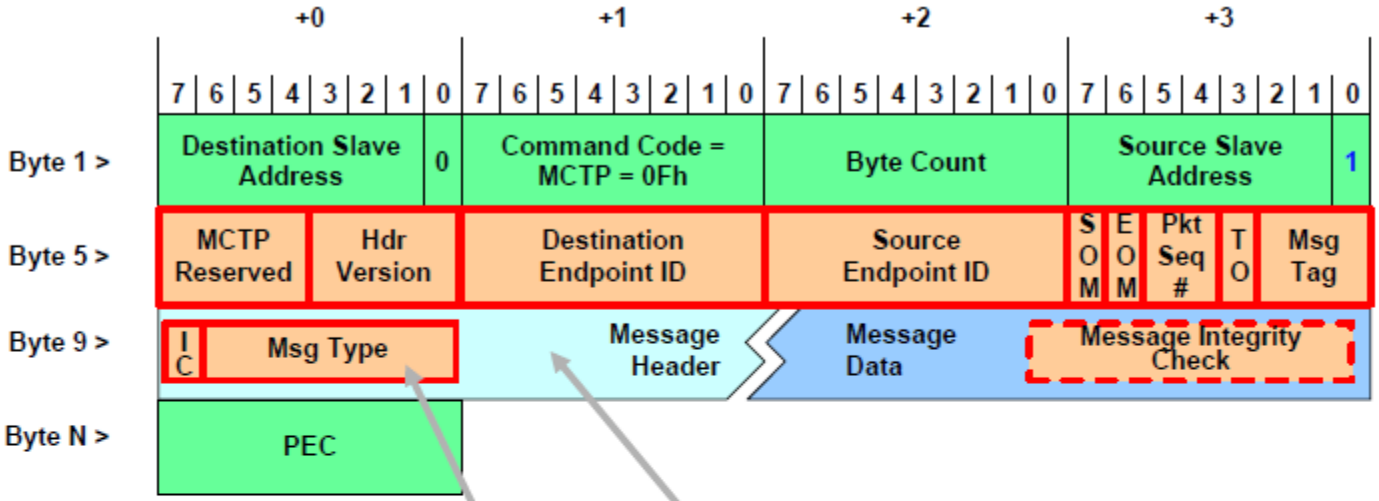
Management Interface Protocol Layering



MCTP Terminology

- MCTP defines a communication model used to transfer data between management entities
- Management Controller (MC): A microcontroller or processor that aggregates management parameters from one or more management devices and makes access to those parameters available to local or remote software
- Management Device: A device managed by a Management Controller
- MCTP Packet: Base unit of transfer in MCTP.
- MCTP Message: One or more MCTP Packets.

MCTP Packet - SMBus/I²C

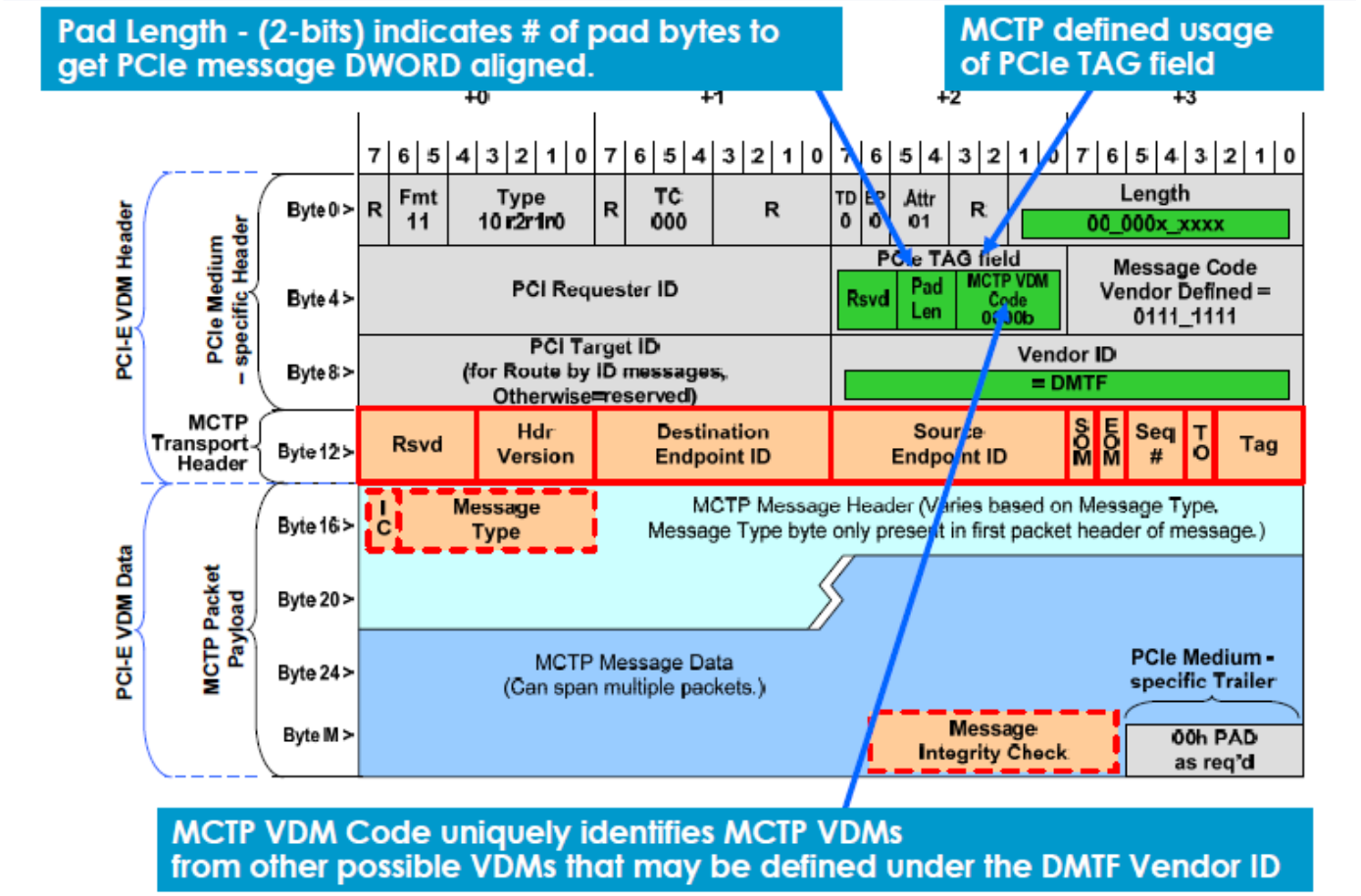


MCTP Message Header
(Varies based on Message Type)

Message Type byte
(only required in first packet header of message.)

= common fields for all MCTP messages

MCTP Packet - PCIe VDM

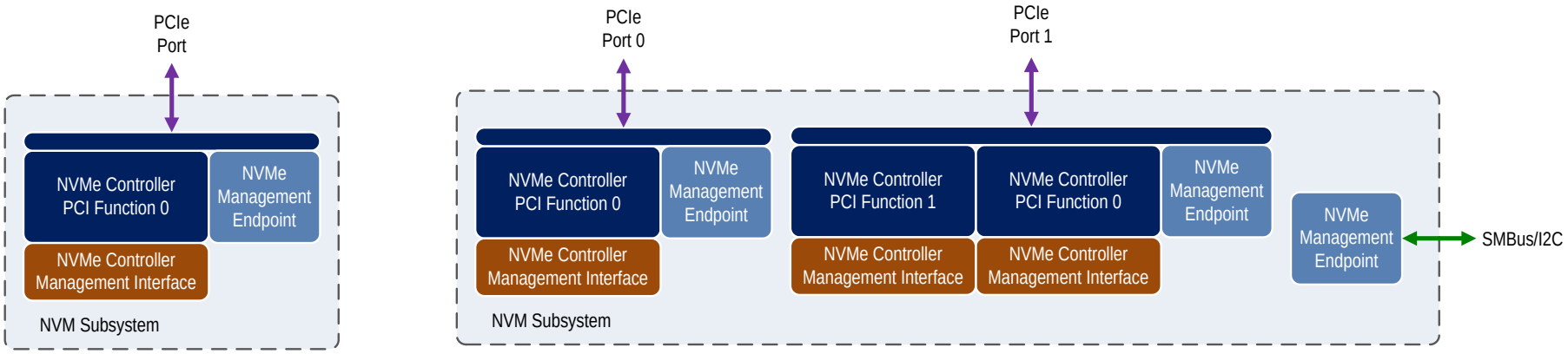


NVMe Management Interface Command Set Overview (preliminary)

Command Type	Command
NVMe Management Interface Specific Commands	Controller Inventory
	Read / Write VPD
	Run Diagnostics
	Health Status
	Command Flow Control
	Exception Handling
	…
PCIe Command	Configuration Read
	Configuration write
	I/O Read
	I/O Write
	Memory Read
	Memory Write
	…

Command Type	Command
NVMe Commands	Get Log Page
	Identify
	Set Feature
	Get Feature
	Firmware Activate
	Firmware Image Download
	Vendor Specific
	Format NVM
	Security Send
	Security Receive
	…

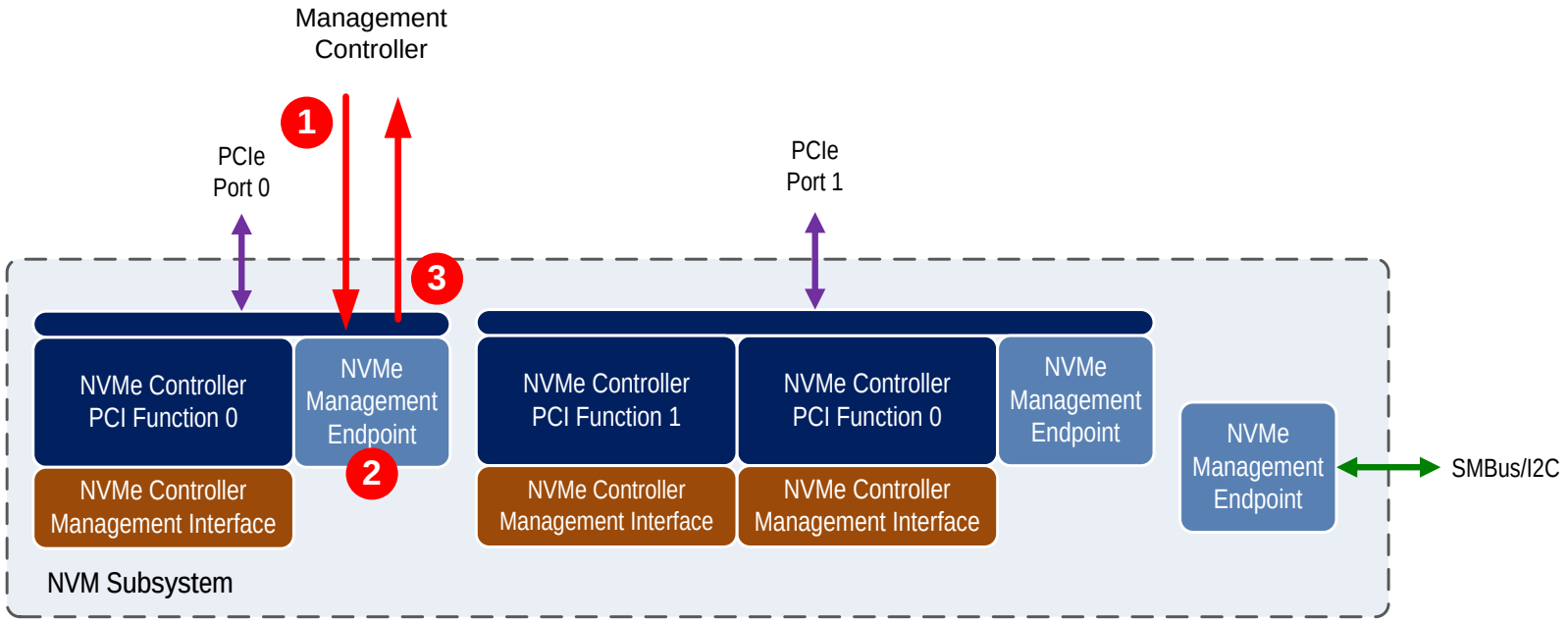
NVM Subsystem Architectural Model



- **NVMe Management Endpoint** – An MCTP endpoint that is the terminus and origin of MCTP packets/messages and which processes MCTP and NVMe Management Interface commands

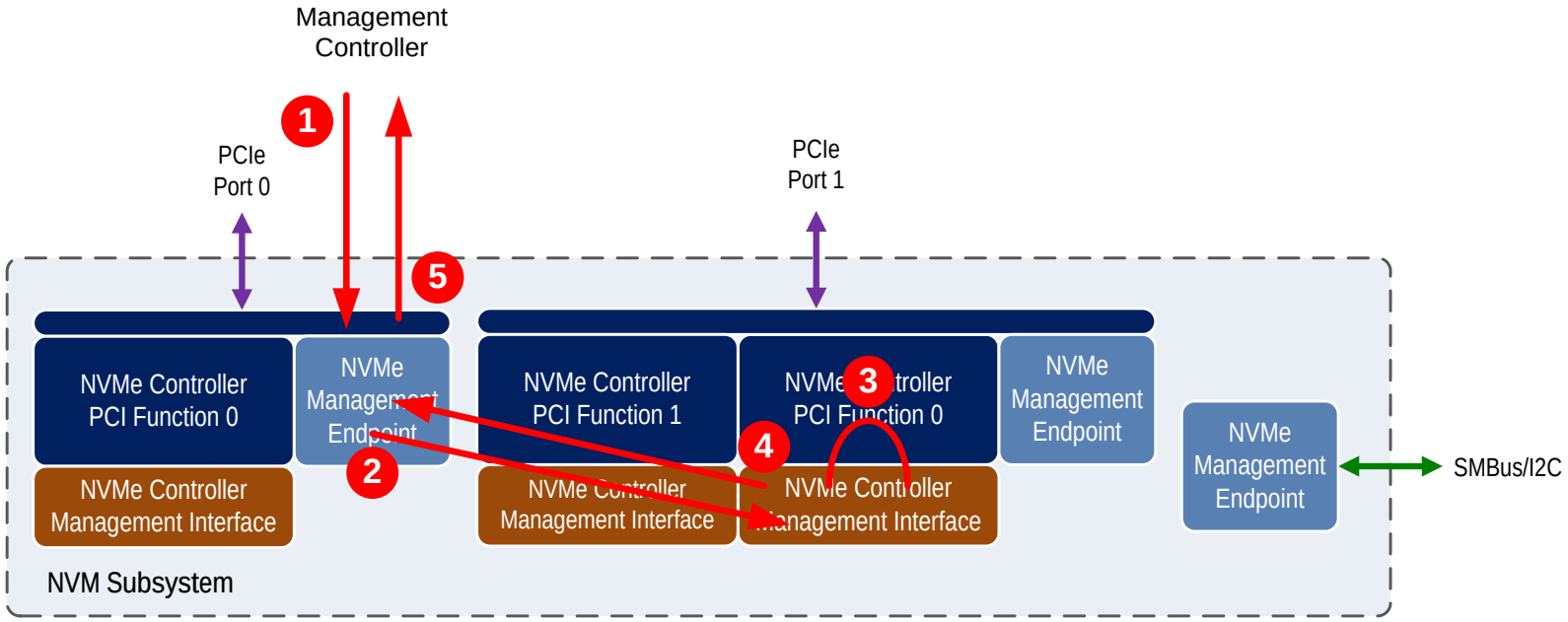
- **NVMe Controller Management Interface** – An interface associated with each NVMe controller in the NVM subsystem that is responsible for executing management operation on behalf on an NVMe Management Endpoint

NVMe Management Interface Specific Command Processing



1. Management Controller sends request message to NVMe Management Endpoint
2. Management Endpoint processes command
3. NVMe Management Endpoint sends response message to Management Controller

PCIe and NVMe Management Command Processing



1. Management Controller sends request message to NVMe Management Endpoint
2. NVMe Management Endpoint forwards request to appropriate NVMe Controller Management Interface
3. NVMe Controller Management Interface executes command on NVMe Controller
4. NVMe Management Endpoint sends response back to NVMe Management Endpoint
5. NVMe Management Endpoint sends response message to Management Controller

Mgmt. Controller (MC)/Host Communication

- MC needs data/notification from host OS, driver, or app
  - Driver version
  - Software/OS RAID information
  - OS name of device (e.g. “/dev/nvme0n1” or “\\.\PhysicalDrive1”)
  - Host changes device configuration (UI)

- MC needs to send data/notification to host OS, driver, or app (UI)
  - MC initiated f/w update is in progress
  - Prepare device for hot removal

- Synchronize access to a shared resource on the NVMe device (UI)
  - Changing power states
  - Setting thresholds (temperature, spare blocks)

Sending Data from Host to MC

- Use existing NVMe Set/Get Features commands
- New Feature Identifiers reserved in NVMe spec.
- Format of each Feature Identifier defined in NVMe Management Interface spec.
- Management Feature Identifiers:
  - NVMe Controller Metadata
  - NVMe Namespace Metadata

Host Data Format

Type-Length-Value (TLV) Element Structure

Type + Version (2 bytes)	Length (2 Bytes)	Value (Length Bytes)
Enumerated value that identifies the type of data in this element. Bits[15:12] = Version Bits[11:0] = Type	Length in bytes of the Value.	Value of this element.

- TLV elements
  - Stored as a list in Get/Set Features Data Structure Element
  - First element at offset 0, second element at offset 4 + Length of first element, etc.

- A value of '0' for the Type is used as a terminator value to the end the TLV element list

Controller Metadata

Type	Value
0h	**End of TLV Elements**
1h	**Feature ID Specific Data**
2h	**Operating System Controller Name**
3h	**OS Driver Name (ODN)**
4h	**OS Driver Version (ODV)**
5h	**Pre-boot Driver Name (PDN)**
6h	**Pre-boot Driver Version (PDV)**
7h	**Current State** (Offline, Online, Prepared for Removal, etc.)

Namespace Metadata

Type	Value
0h	**End of TLV Elements**
1h	**Feature Identifier Specific Data**
2h	**Operating System Namespace Name**
3h	**RAID Information**
4h	**Caching Information**

Sample Controller Metadata

Offset	Contents	Description
0	[15:12] = 0 [11:00] = 5	TLV Element 1 Revision TLV Element 1 Type (Preboot Driver Name)
2	16	TLV Element 1 Length
4	UEFI NVMe Driver	TLV Element 1 Value
20	[15:12] = 0 [11:00] = 6	TLV Element 2 Revision TLV Element 2 Type (Preboot Driver Version)
22	7	TLV Element 2 Length
24	1.2.3.4	TLV Element 2 Value
31	[15:12] = 0 [11:00] = 0	TLV Element 3 Revision TLV Element 3 Type is 0. End of list.

Summary

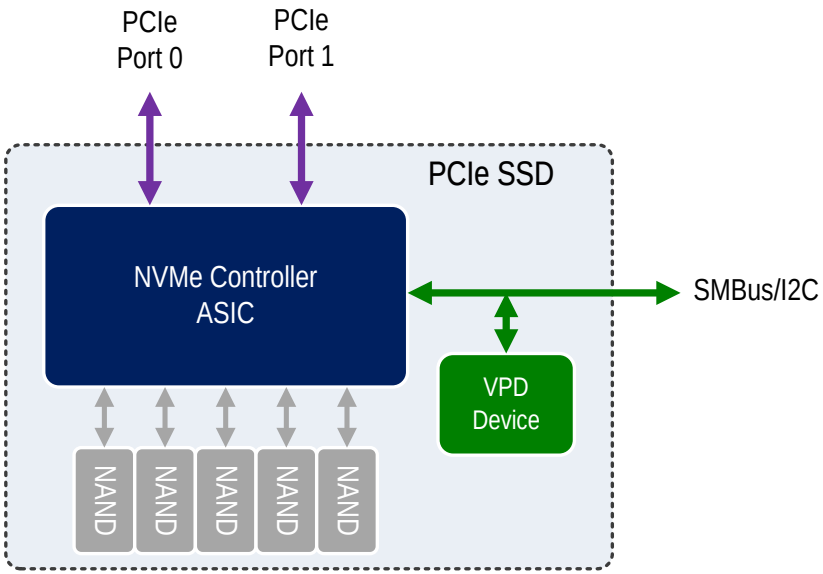
- We are standardizing out-of-band management interface for NVMe storage devices
  - PCIe VDM and SMBus/I2C
- The NVMe management interface is leveraging other management specifications/standards
  - Complementary and not a replacement
- The specification is planned to be completed at the end of this year

References

- MCTP Overview:
  http://dmtf.org/sites/default/files/standards/documents/DSP2016.pdf
  http://www.mandevcon.com/2007/presentations/ts_mctp.pdf
- MCTP Base Spec:
  http://www.dmtf.org/sites/default/files/standards/documents/DSP0236_1.2.0.pdf
- MCTP SMBus/I2C Binding:
  http://www.dmtf.org/sites/default/files/standards/documents/DSP0237_1.0.0.pdf
- MCTP over PCIe VDM Overview:
  http://www.pcisig.com/developers/main/training_materials/get_document?doc_id=6ea959c29d4cd2cdd77667d4d260d64f24374a4d
- MCTP PCIe VDM Binding:
  http://www.dmtf.org/sites/default/files/standards/documents/DSP0238_1.0.1.pdf
- IPMI Platform Management FRU Information Storage Definition:
  http://www.intel.com/content/www/us/en/servers/ipmi/information-storage-definition.html

Backup

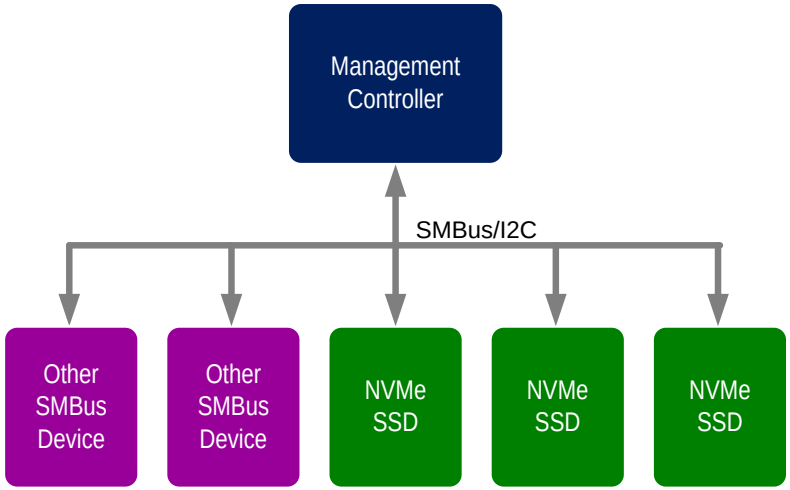
Vital Product Data (VPD)



- Vital Product Data (VPD) contains information about the storage device
  - Examples:
    - Manufacturer
    - Serial number
    - Device configuration
    - Power requirements
  - See IPMI FRU information
- VPD accessible using I2C serial EEPROM read/write operations and NVMe Management Interface commands over MCTP
- Two I2C addresses
  - I2C serial EEPROM access (VPD device)
  - MCTP Endpoint (NVMe controller ASIC)
- VPD accessibility during power modes
  - During Auxiliary Power
    - I2C serial EEPROM read/write
  - During Main Power
    - I2C serial EEPROM read/write
    - NVMe Management Interface commands

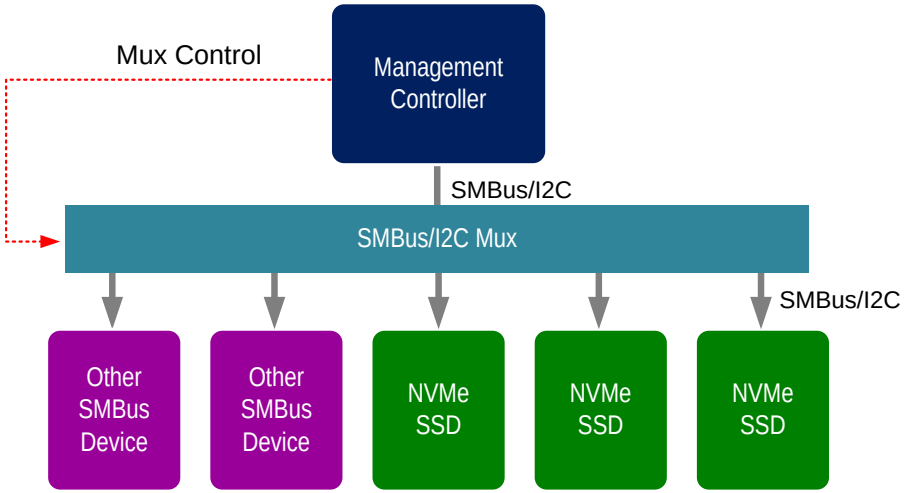
SMBus/I2C Topologies

Shared SMBus/I2C



Requires Unique SMBus/I2C addresses

Segmented SMBus/I2C



Repeated SMBus/I2C Addresses Supported

SMBus/I2C Addressing

- During Auxiliary Power (if supported)
  - I2C serial EEPROM read/write access at default SMBus/I2C address 0xA6, but may be modified using ARP
- During Main Power
  - MCTP Endpoint at default SMBus/I2C address 0xD4, but may be modified using ARP
  - I2C serial EEPROM read/write access
    - If auxiliary power was provided, then SMBus/I2C address shall be maintained if modified using ARP; otherwise, the default address is 0xA6
    - SMBus/I2C address may be modified using ARP
- Supports both shared and segmented SMBus/I2C environments