SDC 18
September 24-27, 2018
Santa Clara, CA
www.storagedeveloper.org

Datacenter Management of NVMe Drives

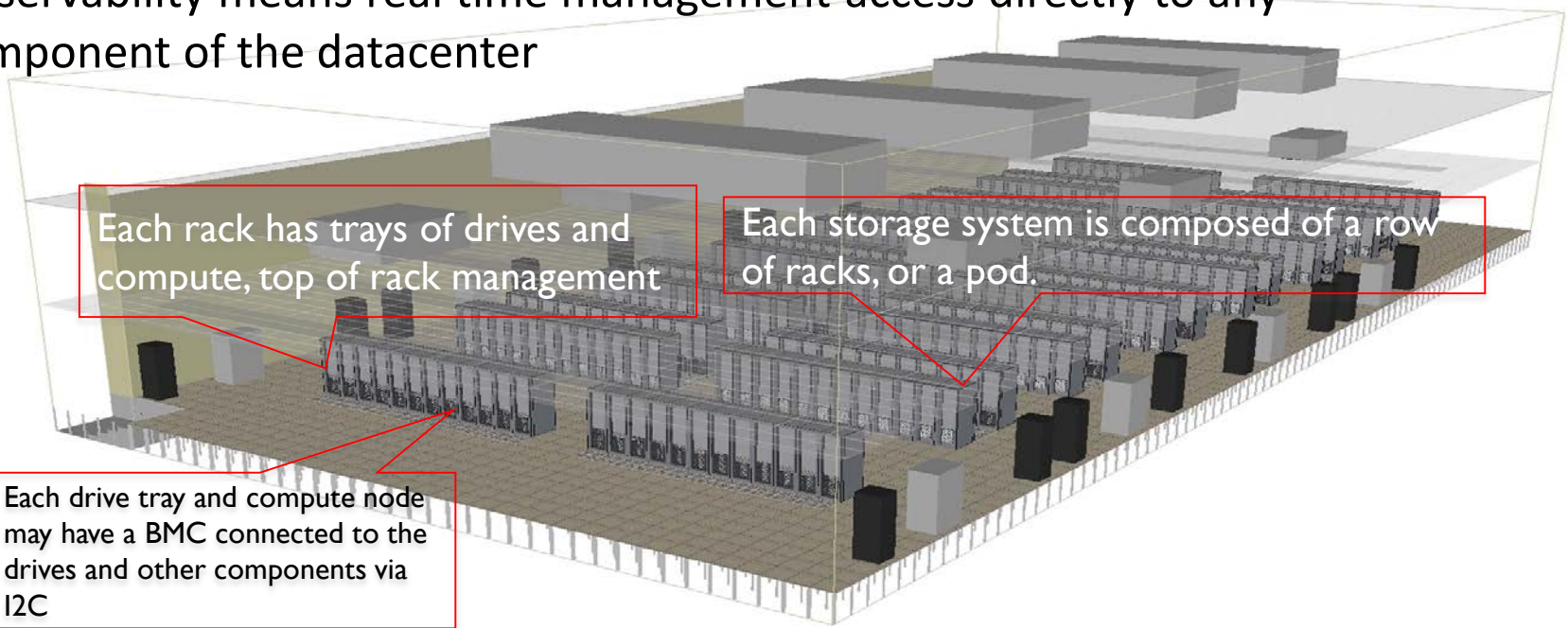
Mark Carlson
Co-chair Object Drive TWG
Toshiba Memory

Abstract

This talk describes work going on in three different organizations to enable scale out management of NVMe SSDs. The soon to be released NVME-MI 1.1 standard will allow management from host based agents as well as BMCs. This might be extended to allow support for Binary Encoded JSON (BEJ) in support of host agents and BMCs that want to support the Redfish Standard. We will also cover work going on in SNIA (Object Drive TWG) and DMTF in support.

Scale Out

- In a datacenter, management is organized at the tray, then rack, then row, then datacenter levels
- Observability means real time management access directly to any component of the datacenter



Management Agents in the Datacenter

- Host based agents deployed primarily in Hyperscalers' systems
  - This doesn't work for most Enterprises
    - Vendor cannot dictate what the customer "runs"
  - Custom management profile constantly changing as new requirements for telemetry and control are realized
  - Management software might be updated multiple times per day
  - Management traffic is In-band of other networking traffic, needs throttling and QoS
  - Uses in-band commands to the drives
- BMC usage starting to be adopted by Hyperscalers
  - Primarily used for in-box temperature management
  - No external management network
    - Only one network everywhere

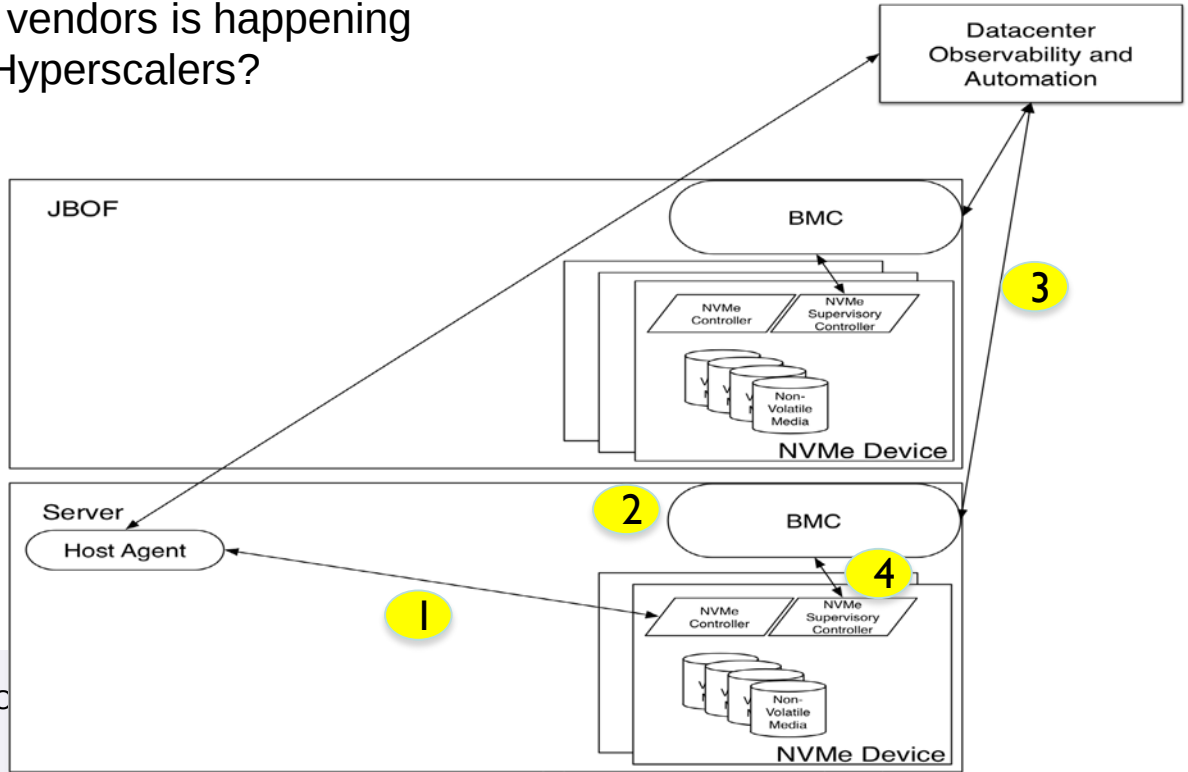
What operations/metrics?

- BMC primarily used to automate box management
  - Temperatures -> Fan control
- Host agent manages a NVMe drive like previous SCSI/SATA drives: in-band
  - Example NVMe Admin commands
    - Firmware update
    - Format/repair the drive
    - Namespace Management

Adopting NVMe-MI

- NVMe-MI is the standard for NVMe Management from a BMC
  - Uses a DMTF standard called MCTP built on top of a cheap network
- Adoption by Enterprise Storage vendors is happening
  - How do we extend this to Hyperscalers?

1. Command line Linux tool for NVMe-MI can be used in their host agent scripting
   - In-band through NVMe-MI send/receive
2. Host agent access to BMC, reporting up only critical issues with box management (tempurature, failures)
3. Using BMC remote protocols to manage the drives
   - Underlying NVMe-MI (4) to the drives
   - Primarily enterprise customers



What is missing?

- Host Agent Support
  - Custom “adapter” from internal model to NVMe-MI semantics – extract metrics for Observability
  - Access to the drive: Inband, MCTP: VDM, SMBus
    - What tools and libraries exist? (cmd-line nvme)
- BMC Support
  - MCTP “stack” over existing I^2C links
  - Adapter for external BMC connection
  - Major vendors (AMI, et. Al.) need to support
  - Open Source BMC firmware (OCP) needs to support

Issue: Proliferation of Models

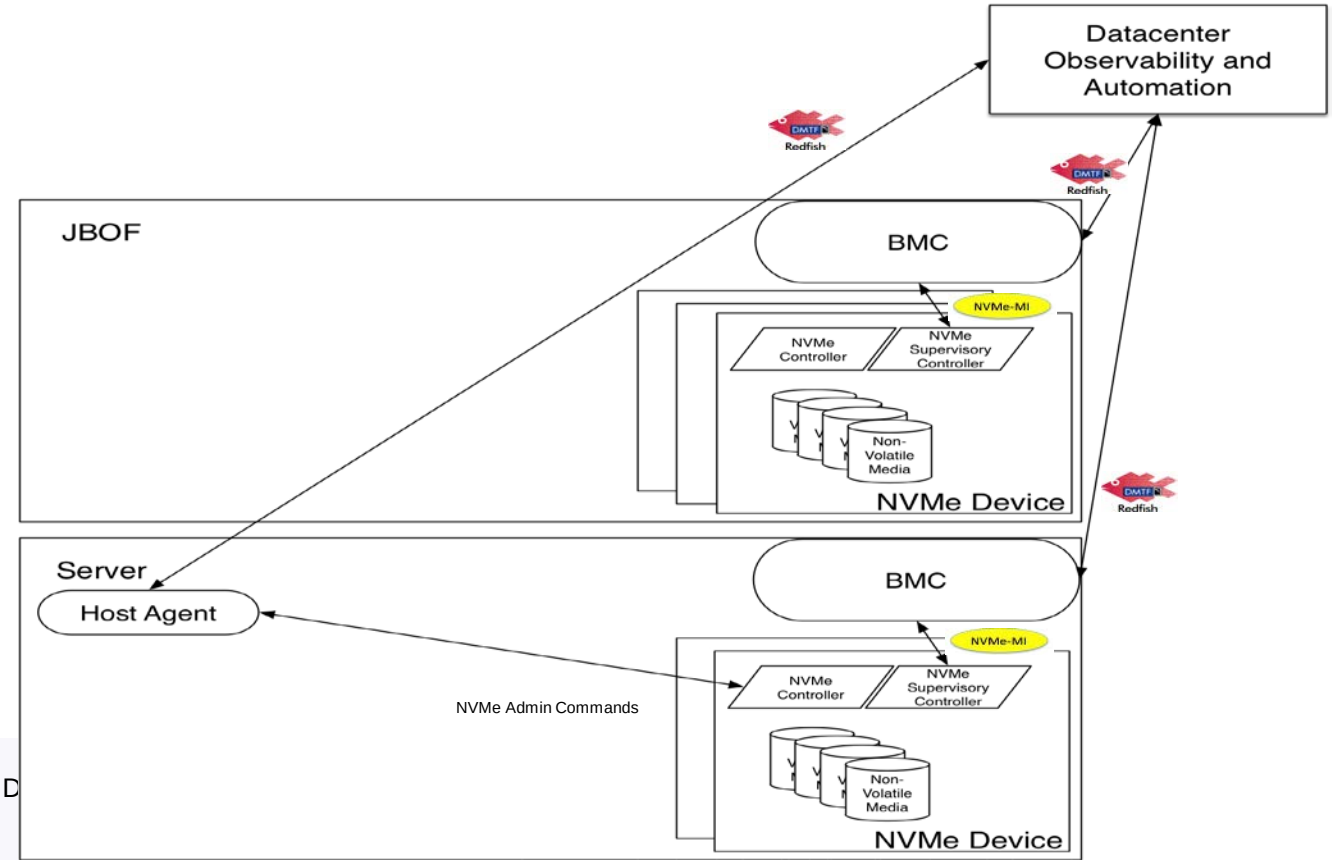
- NVMe-MI can be considered a “model” of the drive and its operation
- Hyperscalers have their own model for telemetry and control – each one has a different one
  - Hyperscalers are the only ones that can write the adapter from NVMe-MI to their own model
- BMCs have differing external interfaces each with their own models
- Outside of the system, can management software be written to an interface in common between a host agent and a BMC?

Enter Redfish

- DMTF standard for scale out datacenter management
  - Based on the tried and true RESTful concepts
  - Standard based on other standards
- Basic philosophy is to have a set of URLs to describe all the components in the data center
  - Each component represents it's functionality directly without intervening re-interpretation/adaption
- Widely adopted by Enterprise Storage and Server Vendors
  - Can be leveraged for Hyperscaler Adoption

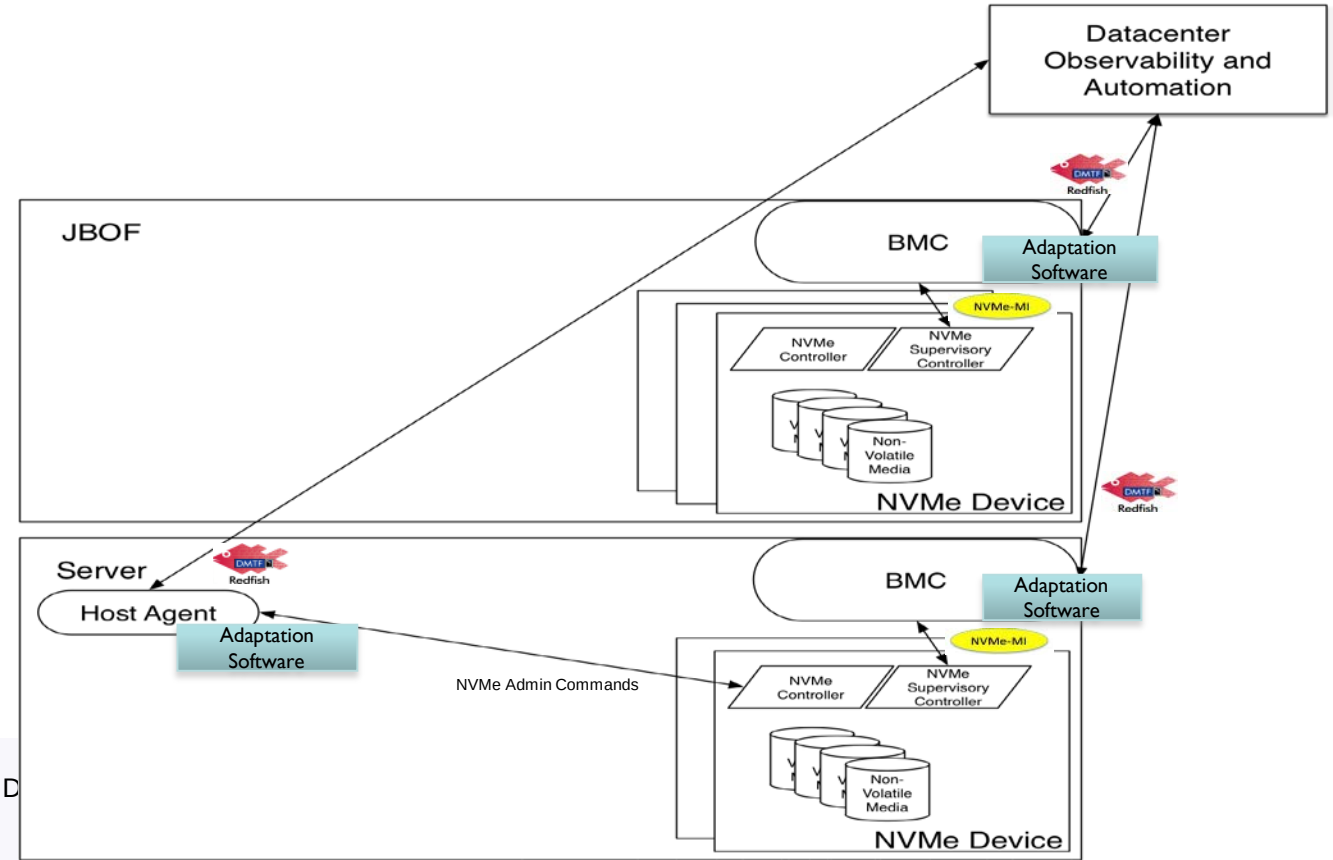
Redfish and NVMe

- NVMe-MI is an “Inside the Box” interface
- Redfish is ideal for an “Outside the Box” interface
- NVMe Admin Commands are In-Band
- What this requires is an Adaptation from NVMe to Redfish



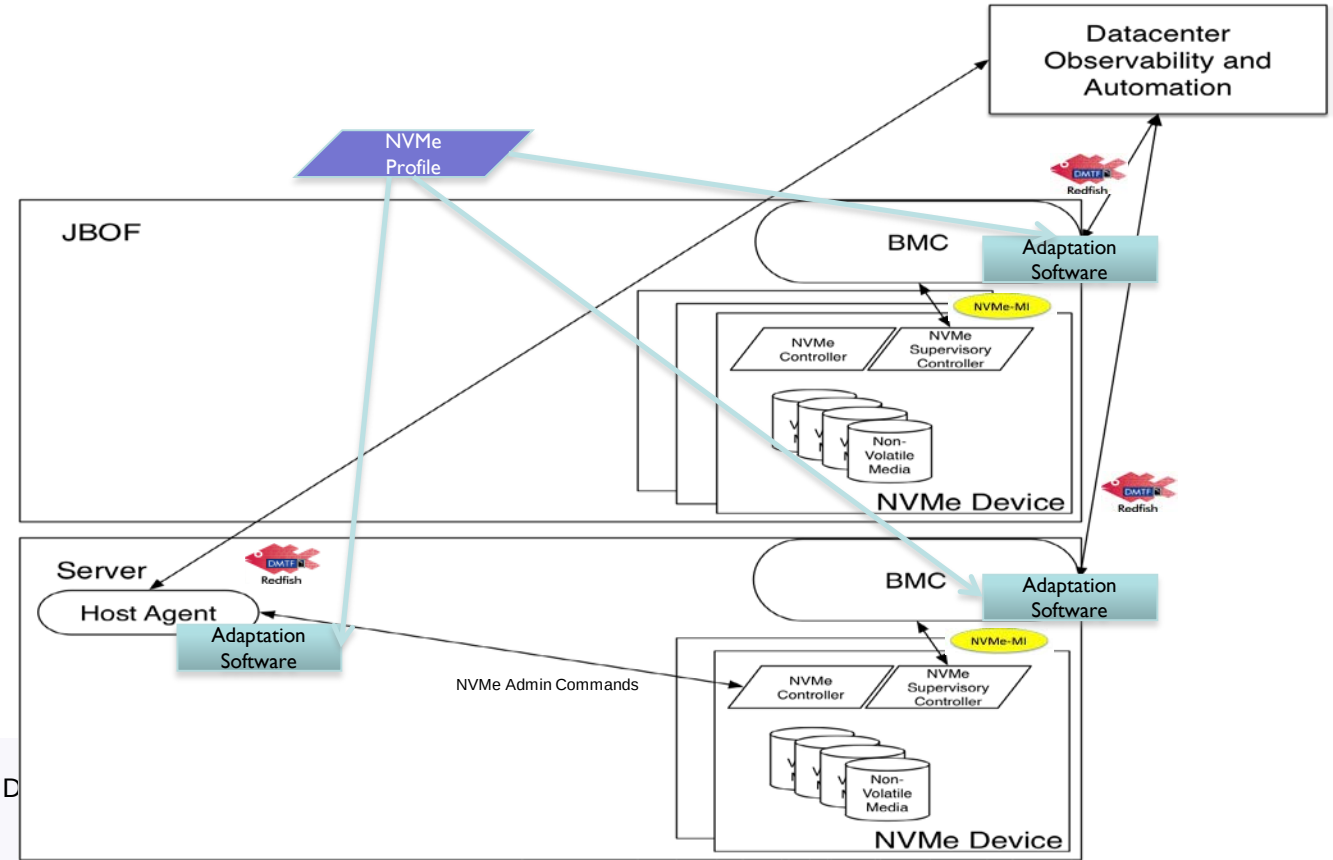
Redfish/NVMe Adaptation

- Adaptation software should be an open source implementation
- Should work for both Host Agents and BMCs
- Even if not using Redfish, can serve as example code for supporting NVMe for an internal model



Adaptation Standardization

- A Redfish profile determines the properties and functions that are required to be supported
- Open Source tools can generate test clients from this profile
- Vendors and Customers can create their own profiles that extend the NVMe Profile



NVMe Redfish Profile

- Open Compute Project is creating a profile for NICs
- SNIA has created a standard profile for IP based drives
  - Has experience with Redfish for drives
- SNIA Object Drive TWG will create an NVMe Redfish Profile
  - Educate SNIA on NVMe
  - Document profile and review with [management-if]
  - Register with DMTF for broad usage

Redfish Device Enablement (RDE)

- DMTF PMCI WG developing a standard to enable a server Management Controller to present a Redfish-conformant management of I/O Adapters in a server without building in code specific to each adapter family/vendor/model.

- Support adapter “self-contained, self-describing” including value-add (OEM) properties
- New managed devices (and device classes) do not require Management Controller firmware updates
- Support a range of capabilities from primitive to advanced devices (lightweight/low bandwidth options)
- Leveraging PLDM, a provider architecture is being specified that can binary encode the data in a small enough format for devices to understand and support.
- MC acts as a proxy to encode/decode the data to/from the provider
- PLDM works over I2C & PCIe VDM. Additional mappings under consideration.

RDE Operations

HTTP Operation	RDE Operation
GET	Read
PUT	Replace
PATCH	Update
POST	Action or Create (Collections)
DELETE	Delete (Collections)
HEAD	Read Headers

15

The RDE API – Discovery and Registration

Command	Usage
NegotiateRedfishParameters	Concurrency and feature support
NegotiateChannelParameters	Asynchrony support and chunk transfer size
GetSchemaDictionary	Dictionary retrieval
GetSchemaURI	Formal schema identification
GetSchemaInstanceETag	Get a digest of schema data

16

Binary Encoded JSON (BEJ)

- Compact binary format for self-describing encoding of multiformat data
- Separates the string labels of the data from the data itself
  - Property names and enumeration strings are static, don't need to be part of the encoding as long as we can systematically restore them later
  - We'll put them into a separate "dictionary" and refer to them by "sequence numbers"
  - Requires that sequence numbers be well-defined. JSON not ordered, so we'll have to impose a canonical ordering on it
- Comparable to ASN.1, but simpler to deal with
  - Providing sequence number AND format for each tuple eliminates need for context-aware decoding
  - Adding counts for sets and arrays enables preallocation of memory for decoded contents without requiring an additional pass through the data
  - Less state required to decode
  - Directly tied to JSON, so we can skip some of the exotic ASN.1 formats and reuse high bits in the format byte as flags
  - Easier to implement: 2 days for BEJ encoder/decoder vs a month for ASN.1

17

Can an NVMe Device support BEJ?

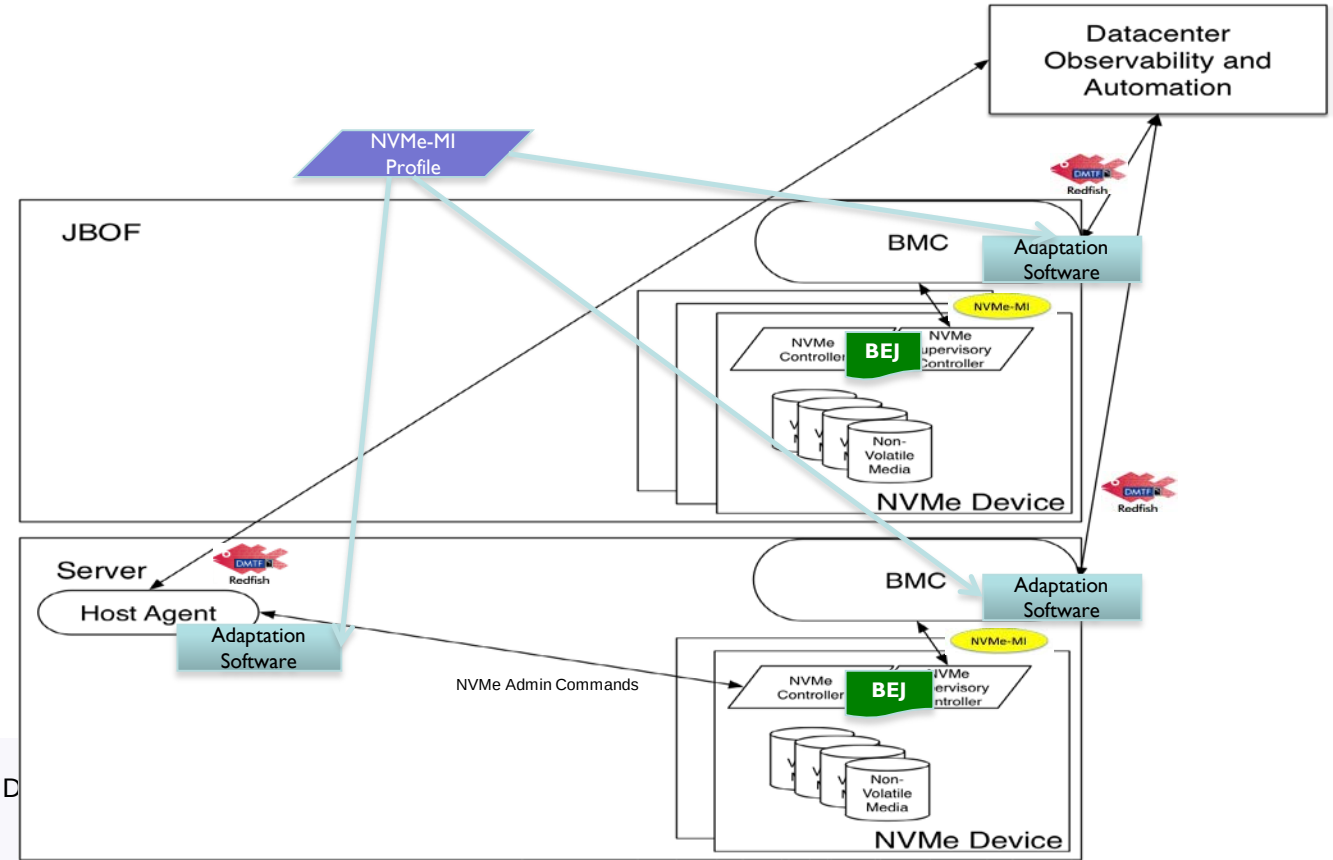
- Just a different encoding of what is already implemented (no new instrumentation)
- Additional support needed in the command set
  - Send/Receive BEJ, Dictionary
- Vendor specific functions and properties are simple extensions to the profile
  - Do not need adaptation to Redfish
- Vastly simplifies the adaptation
  - One generic adaptation for all NVMe devices
  - In the future: Not just Storage Devices!

Computational Storage

- As other disaggregate infrastructure components such as accelerators and FPGAs take on a NVMe interface how are they managed?
  - Do BMCs even make sense still for some ?
  - Why not run the Agent in a computational component?
    - Sized and authorized for the task
- With NVMe Admin commands for Redfish, different components can produce their own profiles - specific to their functions – through this standard interface
  - The management can then interoperably compose these component resources on the fly, tuning them to each purpose

Generic NVMe Adaptation

- Adaptation Software is a generic transposition from BEJ <-> Redfish
- Still needs to talk NVMe-MI or NVMe Admin
- Still NVMe specific

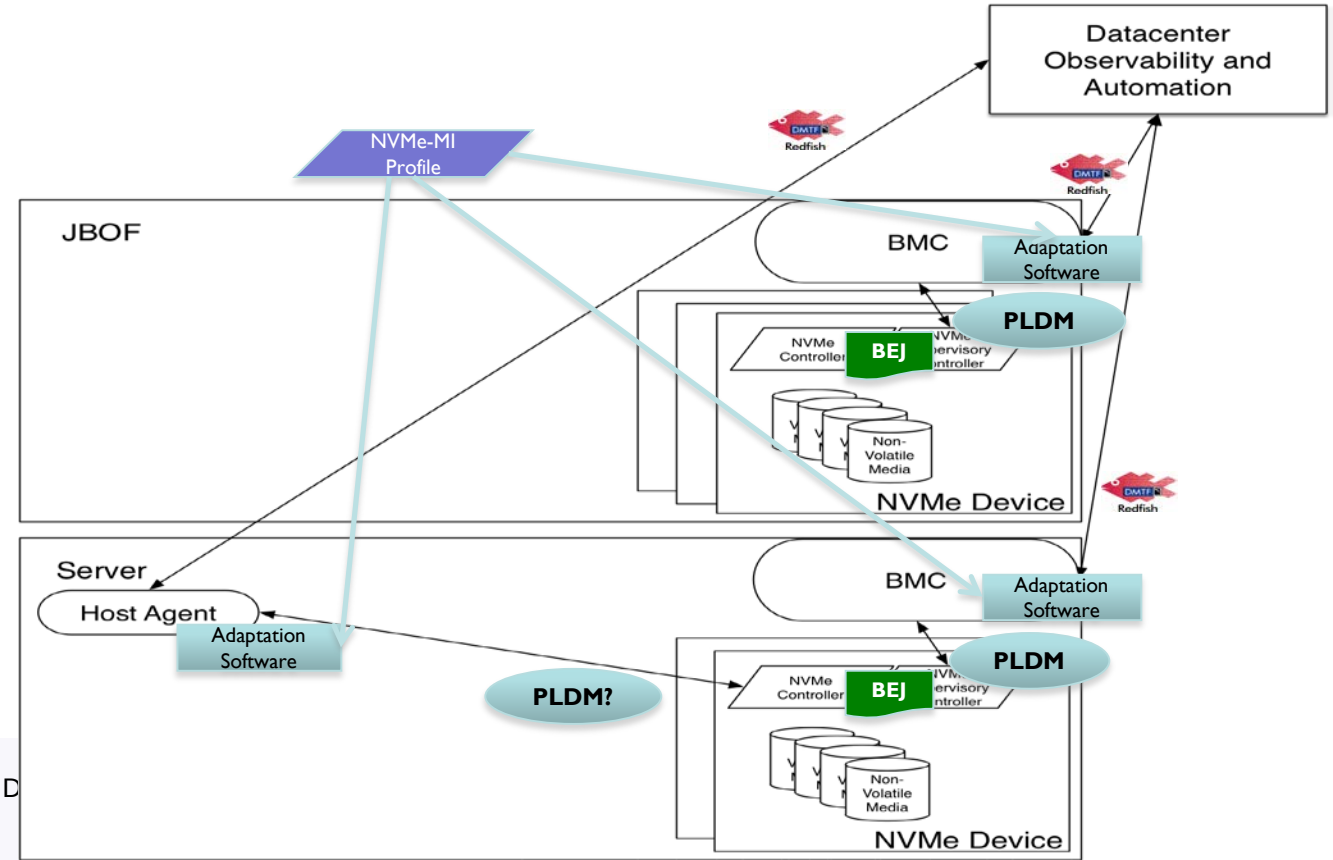


The Future of BMC based Management

- DMTF has a standard Platform Level Data Model (PLDM) DSP0420
  - RDE is based on this standard
  - Uses MCTP like NVMe-MI
- Enterprise System vendors pushing their device vendors in this direction – NICs first

PLDM based NVMe Adaptation

- Leverages RDE over PLDM standard
- Works over all existing control paths (MCTP, VDM, etc.)
- Adaptation Software is now generic for any type of device
- Adjusts to any Profile or Dictionary
- All firmware management is identical
- Roll out as a side by side interface with existing -MI



Cross Association Partnership

- These problems need to be solved cooperatively
  - Hyperscaler and Enterprise vendors are key stakeholders
- SNIA, NVM Express and DMTF organizations have created an Alliance to work in this space
  - Drive and component vendors are engaged
  - They don’t want multiple approaches from each customer, there should always be a core standard that can easily be extended for each major customer

Ethernet NVMe Drives

- Would like to have a RESTful management interface
  - Just implement the NVMe Redfish profile directly via HTTP
  - Add TCP/IP I/F to existing management implementation of the profile
    - Doesn't need to have TCP/IP in the data path
- Scale out management enabled without host or BMC intermediary required

7:00 PM – 8:00 PM Cypress

Birds of a Feather: SNIA NVMe BoF and Meetup

Birds of a Feather

Join the local Bay Area NVMe Meetup group in a session dedicated to NVMe technology. A panel of experts will discuss

Mark Carlson
Toshiba Memory Corpora

Bill Martin
Samsung

J Metz
Cisco Systems

Stephen Bates
Eideticom

Christoph Hellwig

Tom Friend
SK hynix memory solutio

Questions?

Thanks to the SNIA Object Drive TWG for review