SDC 20
SNIA EMEA
FEBRUARY 4-5, 2020
TEL AVIV, ISRAEL
STORAGE DEVELOPER CONFERENCE

Let's Manage "NVMe over Fabrics"

Slawek Putyrski
Principal Engineer
Intel Corporation

Agenda

- Let’s use NVMe
- Let’s do it over Fabrics
- Let’s manage NVMe over Fabrics
- Let’s go “Fishing"
- Summary

Let’s use NVMe

NVMe Refresher

About NVM Express (The Technology)

- NVM Express (NVMe™) is an open collection of standards and information to fully expose the benefits of non-volatile memory in all types of computing environments from mobile to data center

NVM Express Base Specification

The register interface and command set for PCI Express attached storage with industry standard software available for numerous operating systems. NVMe™ is widely considered the defacto industry standard for PCIe SSDs.

NVM Express Management Interface (NVMe-MI™) Specification

The command set and architecture for out of band management of NVM Express storage (i.e., discovering, monitoring, and updating NVMe™ devices using a BMC).

NVM Express Over Fabrics (NVMe-oF™) Specification

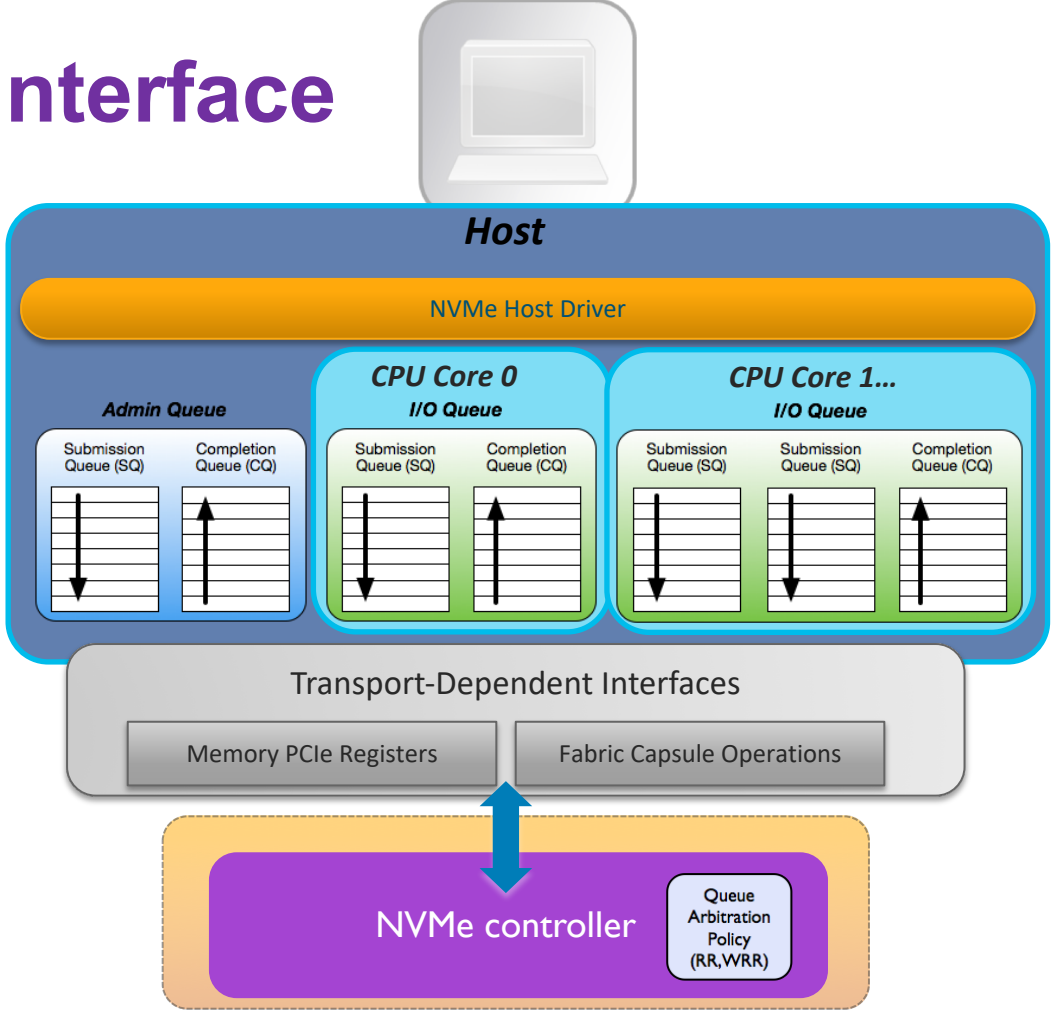
The extension to NVM Express that enables tunneling the NVM Express command set over additional transports beyond PCIe. NVMe over Fabrics™ extends the benefits of efficient storage architecture at scale in the world's largest data centers by allowing the same protocol to extend over various networked interfaces.

NVMe

- Specification for SSD access via PCI Express (PCIe)
- High parallelism and low latency SSD access
- New modern command set with Administrative vs. I/O command separation (control path vs. data path)
- Full support for NVMe for all major OS (Linux, Windows, ESX etc.)

NVMe Multi-Queue Interface

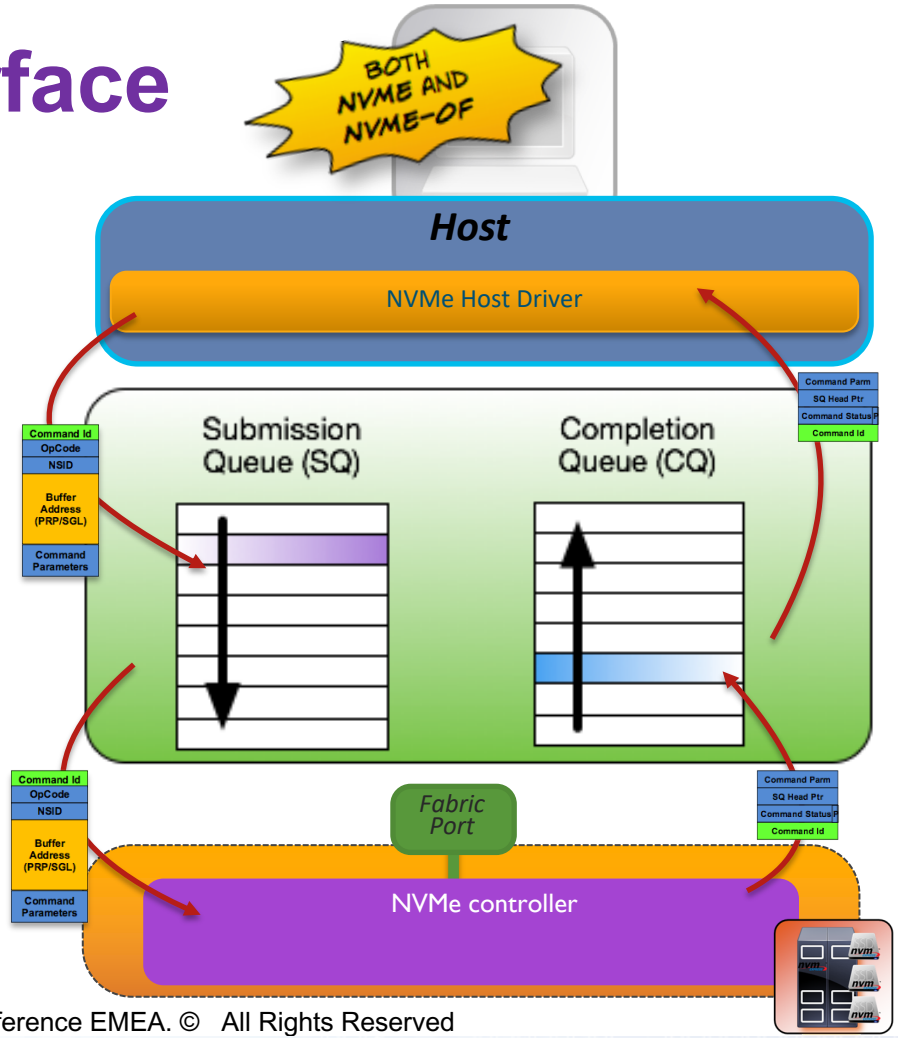
- I/O Submission and Completion Queue Pairs are aligned to Host CPU Cores
  - Independent per-queue operations
  - No inter-CPU locks on command Submission or Completion
  - Per Completion Queue Interrupts enables source core interrupt steering



NVMe Multi-Queue Interface

- Host Driver enqueues the SQE into the SQ
- NVMe Controller dequeues SQE
- NVMe Controller enqueues CQE into the CQ
- Host Driver dequeues CQE

This queuing functionality is always present…
… but ***where*** this takes place can differ

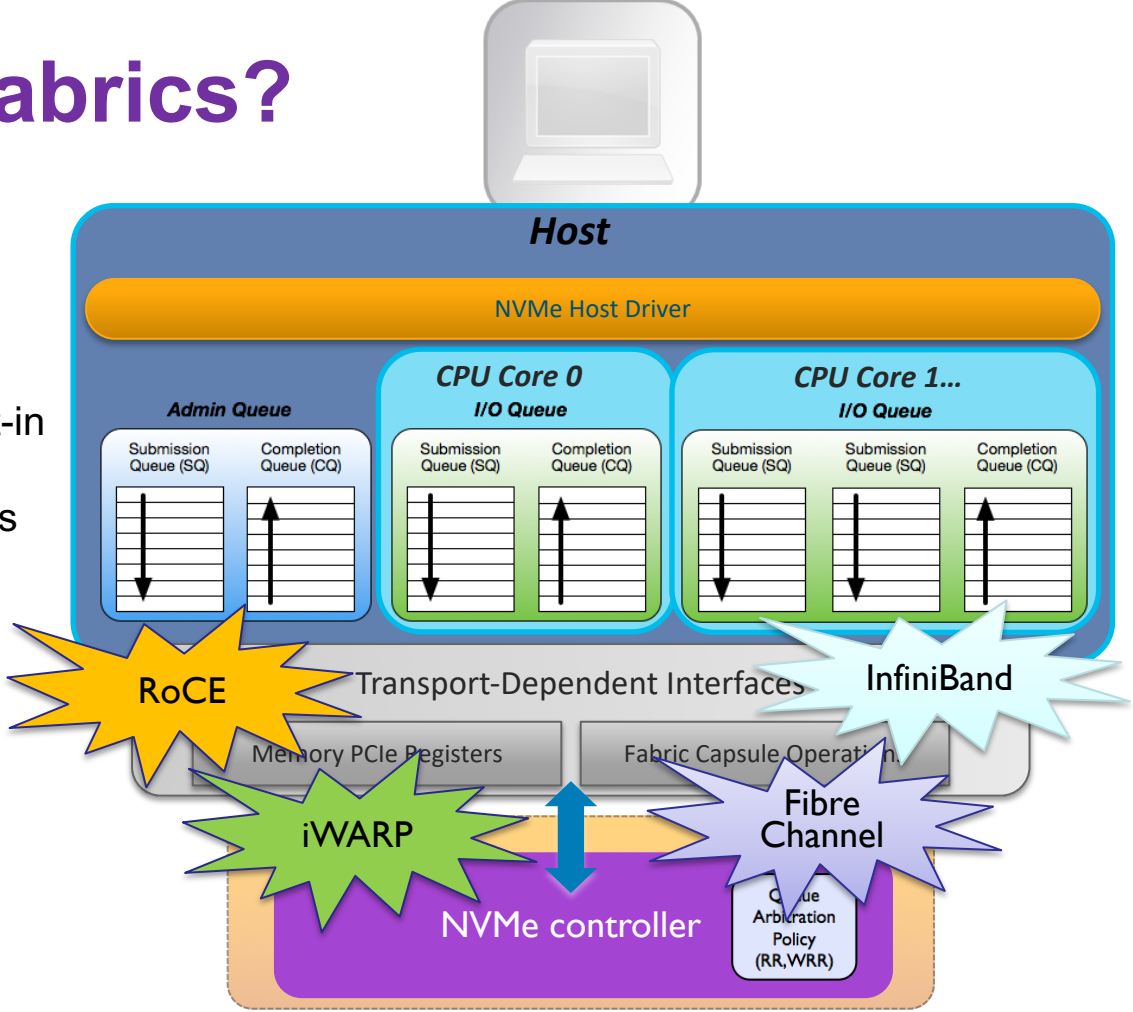


Let’s do it over Fabric

NVMe-oF Refresher

Why NVMe over Fabrics?

- NVMe Functionalities supported
  - Multi-queue model
  - Low latency access
  - Multipathing capabilities built-in
- Optimized NVMe System
  - Same Architecture regardless of transport
  - Extends efficiencies across fabric
- Network Storage
  - Efficient sharing
  - Workload migration support
  - Better capacity utilization

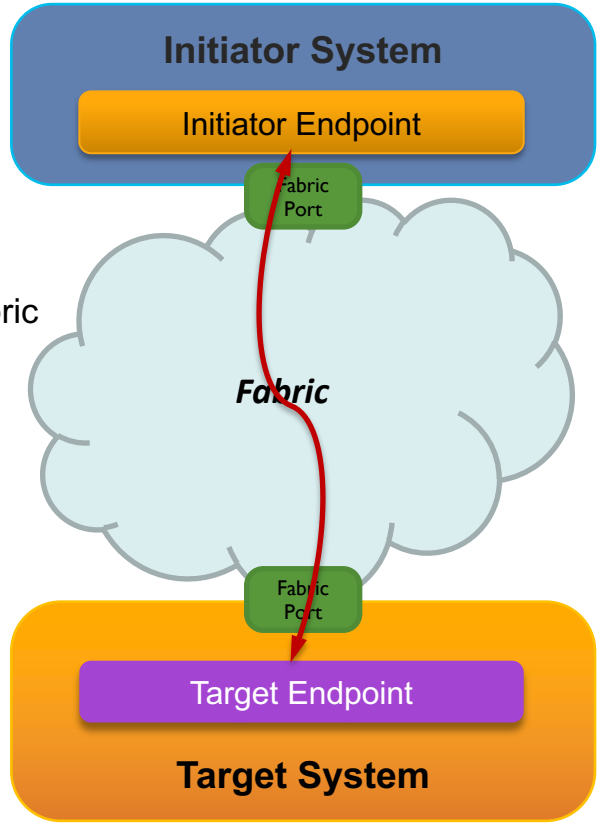


Lets' Manage Fabric

NVMe over Fabrics Management

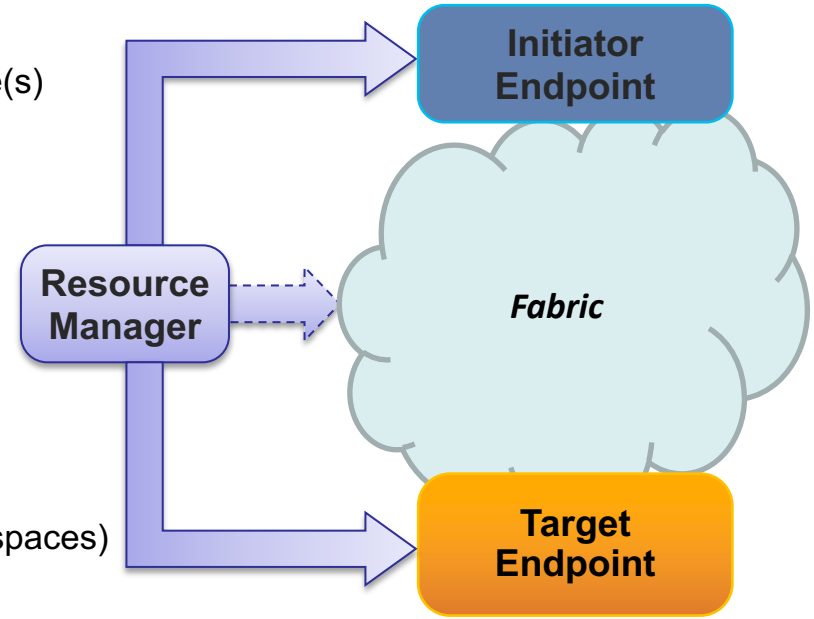
Logical Fabric Management

- Management Points
  - Initiator System
  - Target System
  - Fabric
- Fabric Endpoint
  - Logical representation of physical device(s) accessible through fabric
  - Uniquely identified across fabric instance
  - Contains all information necessary for establishing connections
- Fabric Type independent management model
  - Port-based fabrics (e.g. PCIe)
  - Addressable fabrics (e.g. TCP/IP)
- Connection over Fabric
  - Initiator needs Target Endpoints Data
  - Target needs Initiator Endpoints Data



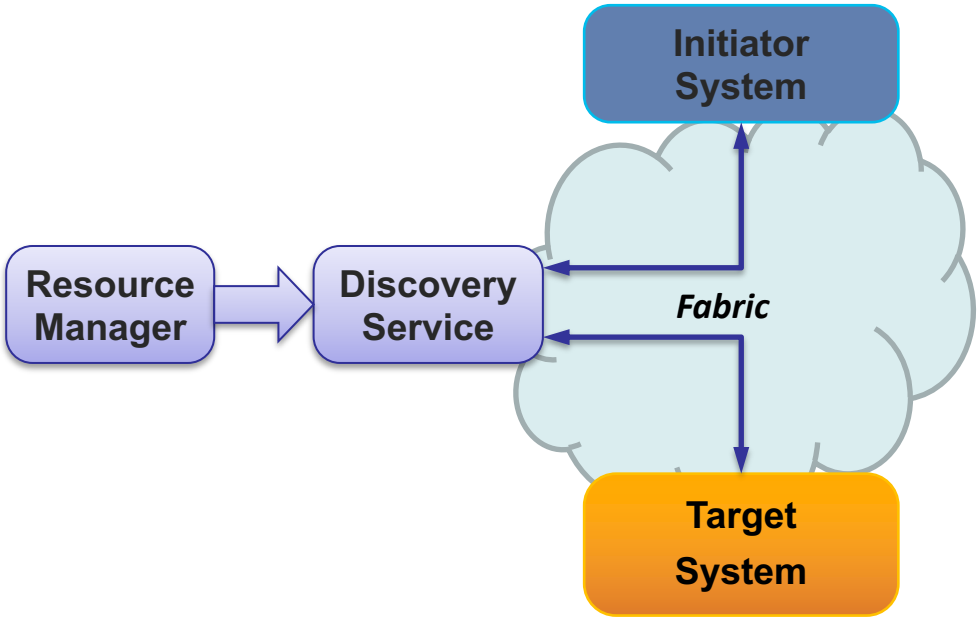
NVMe Over Fabrics

- Management points
  - Initiator Endpoint – Compute System HW or SW
  - Target Endpoint – NVM Subsystem(s) or Namespace(s)
  - Fabric (Optional)
- Initiator configuration
  - Initiator Identifier (NQN)
  - Transport Information (e.g. Protocol, Address)
  - Fabric Port (e.g. Ethernet Interface)
  - Represented Device (e.g. Computer System)
- Target configuration
  - Target Identifier (NQN)
  - Transport Information (e.g. Protocol, Address)
  - Fabric Port (e.g. Ethernet Interface)
  - Represented Device (e.g. NVM Subsystem or Namespaces)



Discovery Service

- Optional Service simplifying NVMe over Fabrics Management
  - Distributed vs. Centralized
- Well Known Service Identifier
  - Defined by NVMe over Fabrics specification
  - Not required to be configured
- Supports various fabric protocols
  - TCP/IP, RDMA
- Persistent discovery service
  - NVMe-oF 1.0 – one-time action
  - NVMe-oF 1.1 – persistent service

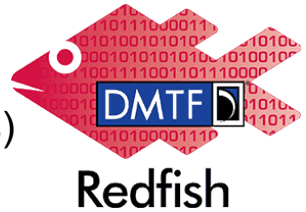


Let’s Go “Fishing”

Redfish and Swordfish

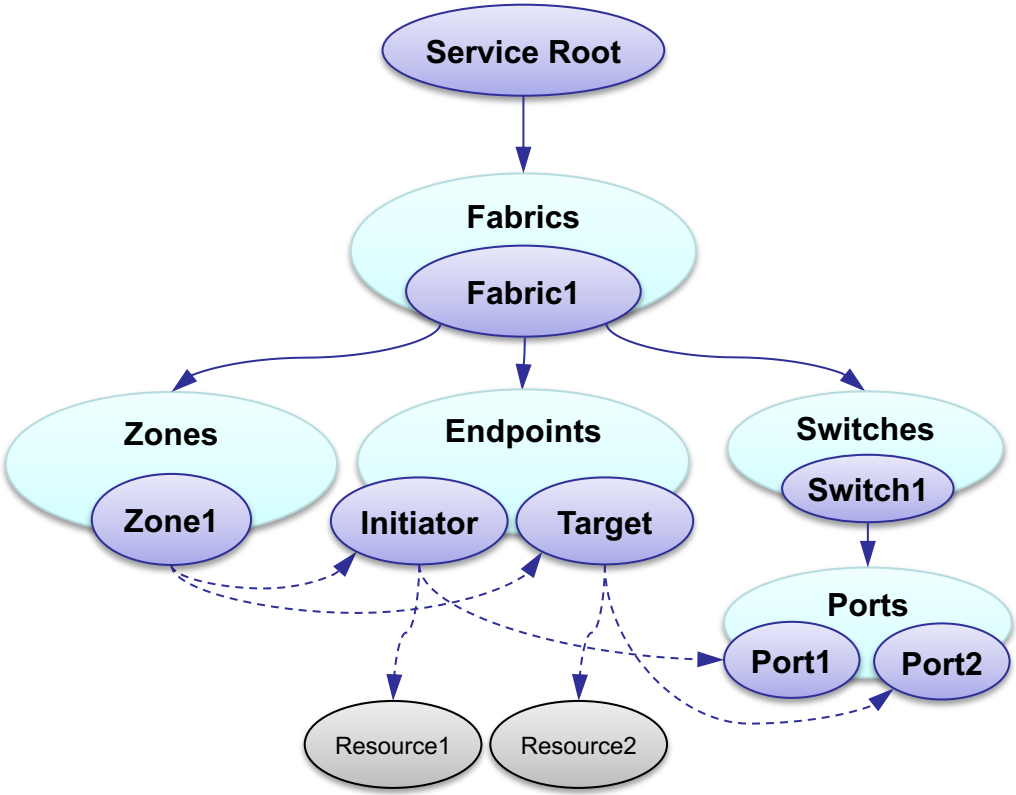
Management Standards

- Redfish ®
  - DMTF Infrastructure Management Standard
  - Hierarchical Management Model with JSON structures and OData Schemas
  - Secure REST API separating management model from transport (https)
  - IPMI Successor
    - Extended Management Scope
- Swordfish
  - SNIA Storage Management Standard
  - Uses and Extends Redfish Management Model
  - Focuses on Storage Management
    - Logical Storage (Block, Object, File)
    - Storage Quality of Services



Fabric Management Model

- Fabric
  - Configuration Umbrella
  - Defines Fabric Type (PCIe, Fiber Channel, iSCSI, NVMe-oF, etc.)
- Switches
  - Fabric Infrastructure configuration (Switches, Ports, etc.)
- Endpoints
  - Represent Resources within Fabric domain
  - Contains identification and access information
- Zones
  - Defines connectivity boundaries and access rules between endpoints



Initiator Endpoint

- Identifier – NQN
- Role – Initiator
- Device Type – Computer System
- Fabric Port – Ethernet Interface in initiator computer system
- Transport Protocol – RoCEv2 RDMA
- Fabric Address – IP Address / Port
- Represented Device Link

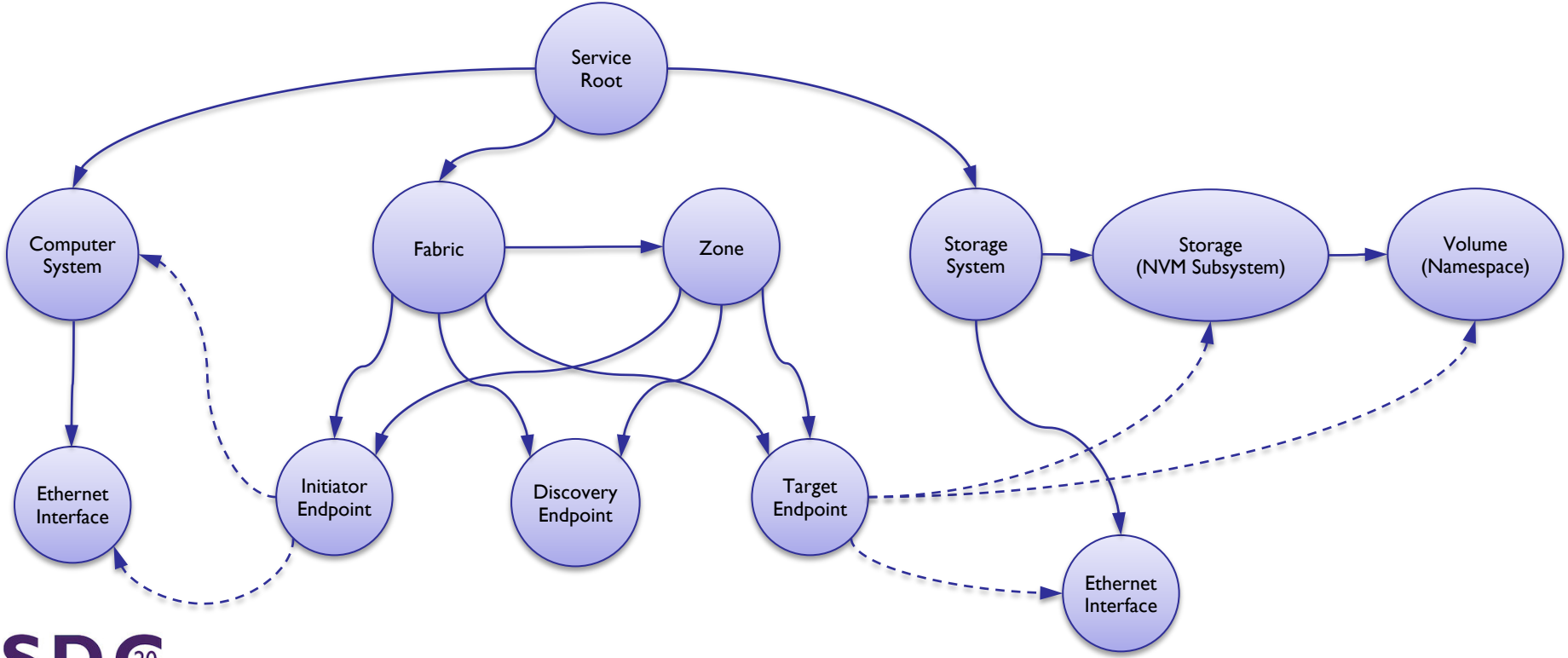

```
{
    "@Redfish.Copyright": "Copyright 2014-2019 DMTF. All rights reserved.",
    "@odata.id": "/redfish/v1/Fabrics/NVMe-oF/Endpoints/Initiator1",
    "@odata.type": "#Endpoint.v1_4_0.Endpoint",
    "Id": "Initiator1",
    "Name": "NVMe-oF initiator 1",
    "Description": "NVMe-oF initiator implemented by the computer system",
    "EndpointProtocol": "NVMeOverFabrics",
    "Identifiers": [
        {
            "DurableName": "host.corp.com:nvme:nvm-subsys-sn-4635",
            "DurableNameFormat": "NQN"
        }
    ],
    "ConnectedEntities": [
        {
            "EntityType": "RootComplex",
            "EntityRole": "Initiator",
            "EntityLink": {
                "@odata.id": "/redfish/v1/Systems/1"
            },
        {
            "EntityType": "NetworkController",
            "EntityRole": "Initiator",
            "EntityLink": {
                "@odata.id": "/redfish/v1/Systems/1/EthernetInterface/1"
            }
        }
    ],
    "IPTransportDetails": [
        {
            "TransportProtocol": "RoCEv2",
            "IPv4Address": {
                "Address": "10.3.5.131"
            },
            "Port": 13244
        }
    ]
}
```

Target Endpoint

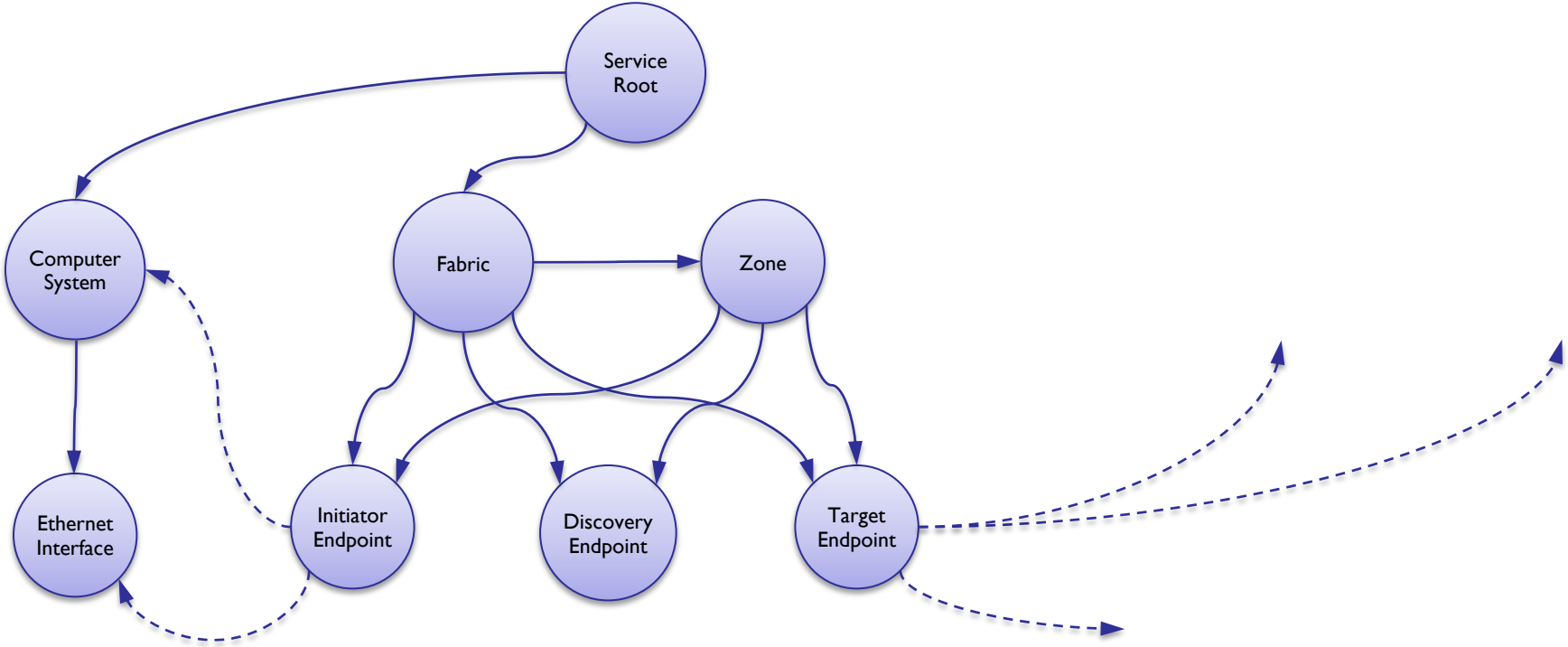
- Identifiers – NQN
- Role – Target
- Device Type – Volume (Namespace)
- Fabric Port – Ethernet Interface in target storage system
- Transport Protocol – RoCEv2 RDMA
- Fabric Address – IP address / port
- Represented Device Link


```
{
    "@Redfish.Copyright": "Copyright 2014-2019 DMTF. All rights reserved.",
    "@odata.id": "/redfish/v1/Fabrics/NVMe-oF/Endpoints/Target1",
    "@odata.type": "#Endpoint.v1_4_0.Endpoint",
    "Id": "Target1",
    "Name": "NVMe-oF target 1",
    "Description": "NVM Namespace",
    "EndpointProtocol": "NVMeOverFabrics",
    "Identifiers": [
        {
            "DurableName": "nqn.corp.com:nvme:nvm-subsys-sn-5381",
            "DurableNameFormat": "NQN"
        }
    ],
    "ConnectedEntities": [
        {
            "EntityType": "Volume",
            "EntityRole": "Target",
            "EntityLink": {
                "@odata.id": "/redfish/v1/StorageSystems/1/Storage/NVMeSubsystem/Volumes/1"
            }
        },
        {
            "EntityType": "NetworkController",
            "EntityRole": "Target",
            "EntityLink": {
                "@odata.id": "/redfish/v1/StorageSystems/1/EthernetInterface/1"
            }
        }
    ],
    "IPTransportDetails": [
        {
            "TransportProtocol": "RoCEv2",
            "IPv4Address": {
                "Address": "10.3.5.132"
            },
            "Port": 13244
        }
    ]
}
```

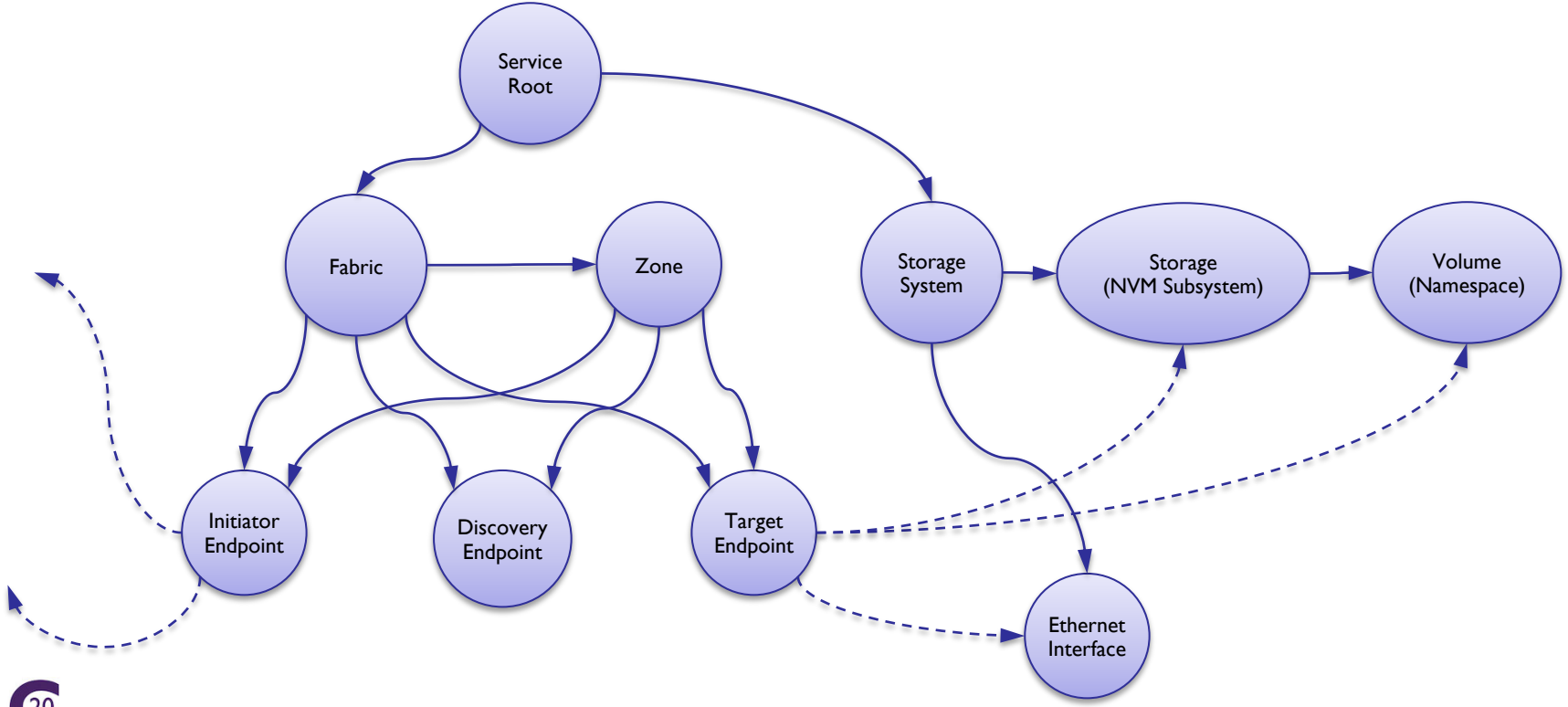
NVMe-oF Management Model



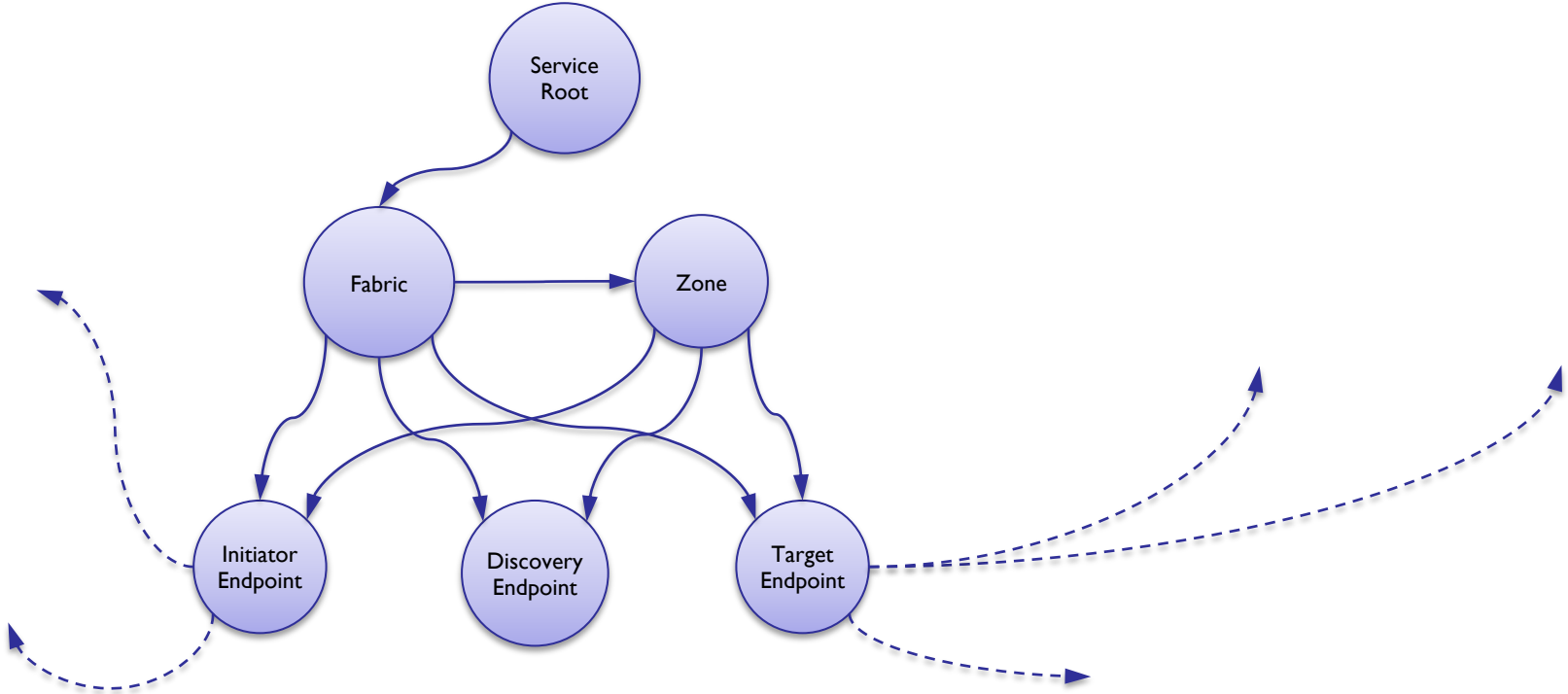
NVMe-oF Initiator Configuration



NVMe-oF Target Configuration



NVMe-oF Discovery Service Config



Summary

- Redfish and Swordfish specifications define logical fabric management model that can be used for NVMe over Fabrics management
- Single model allows management of various NVMe over Fabrics types
  - Ethernet RDMA, TCP/IP, Fiber Channel
- Same model can be used for management of all fabric connected system and services
  - Initiators, Targets, Discovery Services

Let's go "Fishing"

at dmtf.org/Redfish and snia.org/Swordfish

Questions?