Goal of Gen-Z Fabric extensions
• Support the management of port-based Gen-Z Fabrics (Initiators/Targets)
• Support Fabric-attached Resources (Targets)

Status
• Redfish has an existing model for host/target mechanism on a port-based fabric
• Formed a DMTF work register to develop the schema and mockups representing Gen-Z fabric management
• Gen-Z schemas v1.0 approved by Redfish Forum
• Schemas included in Redfish schema release v2019.4
  • DSP8010 available for download at http://www.dmtf.org/standard/redfish
Top Level Fabric Gen-Z Extensions

/redfish/v1/Fabrics
Root Resource
Links to all content

/redfish/v1/Fabrics/GenZ
Model of Gen-Z Fabric
Using Switches, Endpoints, and Zones

/redfish/v1/Systems
Collection of Systems
“Logical” view of general purpose systems

/redfish/v1/Systems/<id>/FabricAdapters
Bridge from SoC interface to Gen-Z Fabric

/redfish/v1/Chassis
Collection of Chassis
“Physical” view of the system

/redfish/v1/Chassis/<id>
Chassis
Chassis global physical asset info

/redfish/v1/Chassis/<id>/MediaController
Media controller for Gen-Z access

/redfish/v1/Chassis/<id>/MemoryDomains
Collection of exportable memory regions

/redfish/v1/Chassis/<id>/Memory
Physical Media of resource

/redfish/v1/Chassis/<id>/MediaControllers
Ports
VCAT

/redfish/v1/Chassis/<id>/MemoryChunks
Switch
FabricAdapters

/redfish/v1/Chassis/<id>/Endpoint
Ports
VCAT

/redfish/v1/Chassis/<id>/AddressPools
MediaController
VCAT

/redfish/v1/Chassis/<id>/MemoryChunks
Switch
FabricAdapters

/redfish/v1/Chassis/<id>/Port
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Memory
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories

/redfish/v1/Chassis/<id>/Media
Memories
Fabric model (Gen-Z) to FAM

Initiator Side

- Systems
- Fabric Adapters
- Address Pools
  - Pool1
  - Zone1
- Zones
- Fabrics
  - GenZ
- Endpoints
  - Initiator Endpoint
  - Target Endpoint
- ConnectedEntity/Links.Endpoints

Target Side

- Chassis
- Media Controllers
- Memory Domains
- Memory
- Memory Chunks
- Ports
- Ports
  - AssociatedEndpoint
  - ConnectedPort/ConnectedPort

Resources

- ConnectedEntity/Links.Endpoints
- Navigation Link
- New resource

Copyright © 2019 DMTF

www.dmtf.org
FABRIC MODEL
Fabric model of a Gen-Z Switch

/redfish/v1/Fabrics
Root Resource
Links to all content
Use to track the Fabric health and state

/redfish/v1/Fabrics/GenZ
Model of Gen-Z Fabric
Using Switches, Endpoints, and Zones

Switch<1D>
Collection of Switch Components

Ports
Routes
VCA
Metrics

Control Space

Endpoints
Zones
AddressPools

Copyright © 2019 DMTF
Fabric Model of a Gen-Z Switch

- **Gen-Z Switch Details**
  - Describes Switch Details
  - Contains Ports
  - Enable/Disables Switch functionality
  - Allows for switch-specific configuration

---

```json
{
  "@odata.id": "/redfish/v1/Fabrics/GenZ/Switches/Switch1",
  "@odata.type": "#Switch.v1_3_0.Switch",
  "@Redfish.Copyright": "Copyright 2014-2019 DMTF. For the full DMTF copyright policy, see "
  "Id": "Switch1",
  "Name": "Gen-Z Switch",
  "SwitchType": "GenZ",
  "Manufacturer": "Contoso",
  "Model": "Switch Model XH13",
  "SerialNumber": "JN2014888L",
  "Ports": {
    "@odata.id": "/redfish/v1/Fabrics/GenZ/Switches/Switch1/Ports"
  },
  "Status": {
    "State": "Enabled",
    "Health": "OK"
  },
  "UUID": "1ad59f5-49f9-51fe-9a93-c549f87f00",
  "Actions": {
    "#Switch.Reset": {
      "target": "/redfish/v1/Fabrics/GenZ/Switches/Switch1/Actions/Switch.Reset",
      "ResetType@Redfish.AllowableValues": [
        "ForceRestart"
      ]
    }
  },
  "Links": {"EndPoints": [
    {
      "@odata.id": "/redfish/v1/Fabrics/GenZ/EndPoints/2"
    }
  ], "Owner": {}}
}
```
Fabric Model of Gen-Z Switch Ports

- Gen-Z Port Details
  - Describes Port details
  - Describes Routing Info
  - Describes Virtual Channels
Fabric Model of Gen-Z Switch Port Routes

- Gen-Z Ports have 2 route tables
  - Linear Packet Relay Table (LPRT)
  - Multi-subnet Packet Relay Table (MPRT)
- LPRT has 4k possible route table entries
- MPRT has 64k possible route table entries
  - Patch each route entry to set route info
  - Use RawEntryHex to patch the entire entry at once
  - Can patch specific route data if required
• Each port has 4k LPRT entries, 64k MPRT entries and 32 VCAT entries
  • Lot of data to transmit to HW
  • High radix switches have many ports to patch!
• Propose to utilize a “Deep Patch” method to patch many entries at once
  • Entries can be sparse
  • Single connection to server for patching many entries
  • Reduces amount of data being transmitted to the HW
  • Use relative identifier “Id” to reduce amount of characters being sent to HW
• Deep Patch support expected in next release of Redfish Specification

Ex. PATCH /redfish/v1/Fabrics/GenZ/Switches/Switch1/Ports/1 [ {“LPRT”:{ “Members”:[ { “Id”: 1, “RawEntryHex”: “0x12dfeb” } ] } } ]
Example Deep Patch (Proposed)

- PATCH /redfish/v1/Fabrics/GenZ/Switches/Switch1/Ports/1

  
  ```
  [  
    { "LPRT": {  
      "Members": [  
        { "Id": 1, "RawEntryHex": "0x12dfeb" },  
        { "Id": 2, "RawEntryHex": "0x334ddf" },  
        { "Id": 9, "RawEntryHex": "0x073e5d"}  
      ]  
    },  
    { "MPRT": {  
      "Members": [  
        { "Id": 1, "RawEntryHex": "0x31124fd"},  
        { "Id": 7, "RawEntryHex": "0x1b458df"}  
      ]  
    }  
  ]
  ```
FABRIC ATTACHED TARGET MODEL
Fabric model to a Fabric Attached Target

/redfish/v1/Fabrics
Root Resource
Links to all content

/redfish/v1/Fabrics/GenZ
Model of Gen-Z Fabric
Using Switches, Endpoints, and Zones

/redfish/v1/Chassis
Collection of Chassis
“Physical” view of the system

/redfish/v1/Chassis/<id>
Chassis
Chassis global physical asset info

/redfish/v1/Chassis/<id>/MediaControllers
Media controller for Gen-Z access

/redfish/v1/Chassis/<id>/MemoryDomains
Collection of exportable memory regions

/redfish/v1/Chassis/<id>/Memory
Physical Media of resource

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory

/redfish/v1/Chassis/<id>/MediaControllers

/redfish/v1/Chassis/<id>/MemoryDomains

/redfish/v1/Chassis/<id>/Memory
Model of Fabric Attached Memory

- FAM resides in Chassis
  - Physical description of media
  - Not associated with a specific system
    - Not attached to an SoC
  - Can be composed with multiple systems
- Add MediaControllers to describe FAM controller
- Use Memory and MemoryDomains to describe assignable media
Model of MediaController

- MediaController physical device
  - Has Gen-Z Ports, same as Fabric Switches
- Can configure MediaController settings
- Describes status of controller
- Links to associated endpoint and MemoryDomain
Fabric Endpoint describes MediaController and Memory Chunks

Each MemoryChunk describes a region of FAM
- Includes attributes of the region like Region Key

Each Endpoint describes the Global Component Identifier (GCID)
- Logical Fabric Identifier
- Used in Gen-Z Address from Initiator
FABRIC ADAPTER MODEL
Fabric model of a Gen-Z Bridge (Initiator)

/redfish/v1/Fabrics
Root Resource
Links to all content

/redfish/v1/Fabrics/FabricsExtension
Use to track the Fabric health and state

/redfish/v1/Systems
Collection of Systems
“Logical” view of general purpose systems

/redfish/v1/Systems/<id>/FabricAdapter
Bridge from SoC Interface to GenZ Fabric

/redfish/v1/Fabrics/GenZ
Model of Gen-Z Fabric
Includes Switches, Endpoints, and Zones

Switches
Endpoints
Zones
AddressPools

Routes
VCAT
Metrics

Switch
MediaController
FabricAdapter

Ports
VCAT
Metrics

SSDT
MSDT
REQ-VCAT
RSP-VCAT

Routes

Copyright © 2019 DMTF
Model of a Gen-Z Fabric Adapter (Initiator)

- Fabric Adapter bridges Gen-Z Fabric and SoC Interface
- Fabric Adapter has a Gen-Z Requestor to inject packets into the Fabric
  - Uses Routing tables to determine path to target
  - Describes the details of the Adapter HW
- Fabric Adapter may also contain an Integrated switch
  - May contain ports for packet relaying
- Fabric Adapters may also be a responder on the Gen-Z Fabric
Some Fabric Adapters may contain an embedded switch

Embedded Switches have Gen-Z Ports

- If ports can relay traffic, they would have route tables
- If Ports do not relay traffic, ports would not require routing tables
- Ports also describe Virtual Channels (VCAT) like Switch ports
• **Embedded Switch Ports**
  - Same as switch ports in Fabrics
  - Ports can relay traffic, so they have route tables
  - Ports also describe Virtual Channels (VCAT)
• Fabric Adapter with multiple ports has route tables
  • Single Subnet Destination Table (SSDT)
  • Multi-subnet Destination Table (MSDT)
  • Describes which egress port to get to each destination
  • Describes which virtual channel to use on a particular egress port
  • Each Entry can have a set of possible egress routes
Model of a Fabric Adapter Virtual Channels

- Fabric Adapter has Virtual Channel Action Tables
  - For Requestor and Responder (REQ-VCAT, RSP-VCAT)
  - Describes which virtual channels to use
Address Pools to provide constraints

- Provides constraints on the values in Endpoints
  - Ex. Limiting CID values for a Gen-Z Address
- Typical constraints are minimum/maximum
- All endpoints within that pool would adhere to the pool constraints
- Address Pool would be specific to a particular fabric type
Address Pools for Endpoints

• Elements Within an Address Pool
  • Endpoints
  • Zones

• Allows for constraints for a group of endpoints
Address Pools for Zones

• Elements Within an Address Pool
  • Endpoints
  • Zones

• Allows constraining all endpoints within a specific zone

• Provides scalability for managing address ranges
Zones need scalability

- Zones represent connections
  - Which resources are allowed to communicate with other resources
- Large scale fabrics can have thousands of zones with thousands of endpoints within those zones
- Allowing one zone to start communicating with another group would require another zone
- Not scalable, would have to create another zone with all the previous zone endpoints within that zone
- Hard to manage
  - Removing a single endpoint would require DELETEing from all zones containing that endpoint
Zone of Zones

• Created a new Zone of Zones
• Allow for endpoints in one zone to communicate within another zone
• Reduces number of required IOs from a client
• Scalable for large fabrics
• Uses ZoneType to differentiate Zone of Zones from Zone of Endpoints
Example Zones

```json
{
    "@odata.id": "/redfish/v1/Fabrics/Gen2/Zones/2",
    "@odata.type": "#Zone.v1_4_0.Zone",
    "@redfish.Copyright": "Copyright 2014-2019 DMTF. For the full DMTF copyright policy, see https://www.dmtf.org/
    "ID": "2",
    "Name": "Gen-2 Zone 2",
    "Description": "Gen-2 Zone 2",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "ZoneType": "ZoneOfEndpoints",
    "Links": {
        "Endpoints": [
            {
                "@odata.id": "/redfish/v1/Fabrics/Gen2/Endpoints/2"
            }
        ],
        "AddressPools": [
            {
                "@odata.id": "/redfish/v1/Fabrics/Gen2/AddressPools/API"
            }
        ],
        "ContainedBy": [
            {
                "@odata.id": "/redfish/v1/Fabrics/Gen2/Zones/4"
            }
        ],
        "Oem": {}
    }
}
```
EVENTS AND NOTIFICATIONS
Unsolicited Event Packets

- Gen-Z UEPs have a table of descriptions for each Unsolicited Event
- Need to generate a Message Registry based on the UEP table defined in the Gen-Z Core Specification to describe each event type
  - Table 6-40 in the Gen-Z Core Spec. has the list of events to turn into individual messages
  - Take the UEP format to provide parameters for the messages
Questions/ Comments?