CXL Fabric Management

Vincent Haché

Director of Systems Architecture, Rambus

Agenda

- CXL Overview
- System Management Requirements
- What is a Fabric Manager?
- Component Command Interface
- Management Architecture
- Management Command Sets
- MLD Management

References

There are 5 key resources covering Fabric Management:

1. CXL 2.0 Specification (Link)
   - Introduces In-band device management via mailbox
   - Introduces FM API and MCTP transport details
2. CXL 2.0 Errata (Link)
   - Critical fixes to asynchronous MCTP event notifications
3. Type 3 Management Using MCTP CCI ECN (Link)
   - Enables MCTP-based device management, and generalizes key concepts that were formerly switch-specific
4. CXL FM API over MCTP Binding Specification (Link)
5. CXL Type 3 Device CCI over MCTP Binding Specification (Link)

CXL Delivers the Right Features & Architecture

Challenges

Industry trends driving demand for faster data processing and next-gen data center performance

Increasing demand for heterogeneous computing and server disaggregation

Need for increased memory capacity and bandwidth

CXL

An open industry-supported cache-coherent interconnect for processors, memory expansion and accelerators

Coherent Interface

Leverages PCIe® with 3 mix-and-match protocols

Low Latency

.Cache and .Memory targeted at near CPU cache coherent latency

Asymmetric Complexity

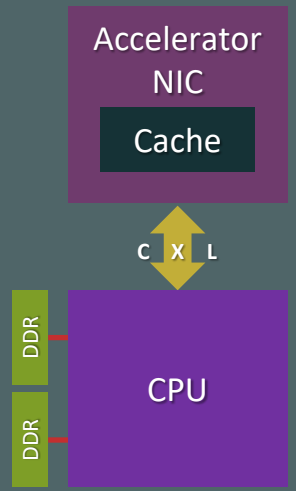
Eases burdens of cache coherent interface designs

CXL 2.0 Usage Models – Recap

Caching Devices / Accelerators

Usages:
- PGAS NIC
- NIC atomics

Protocols:
- CXL.io
- CXL.cache

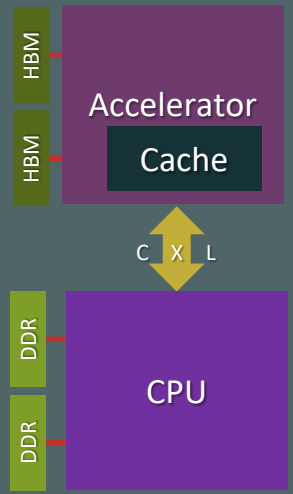


Type 1 Device

Accelerators with Memory

Usages:
- GPU
- FPGA
- Dense Computation

Protocols:
- CXL.io
- CXL.cache
- CXL.memory

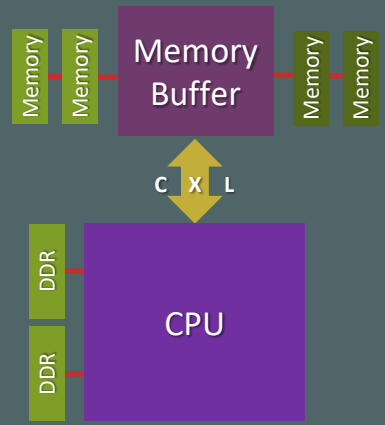


Type 2 Device

Memory Buffers

Usages:
- Memory BW expansion
- Memory capacity expansion
- Persistent Memory

Protocols:
- CXL.io
- CXL.mem

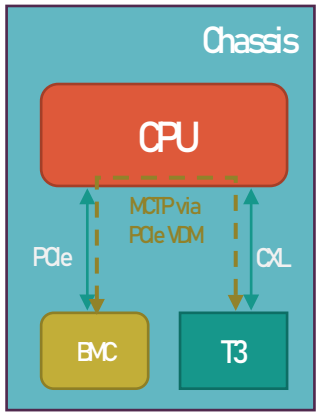


Type 3 Device

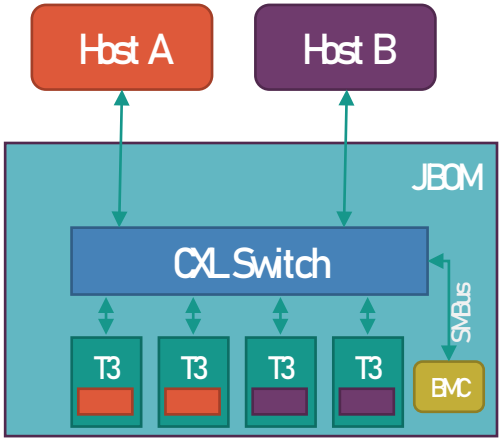
System Management Requirements

Existing Conventions:

Server



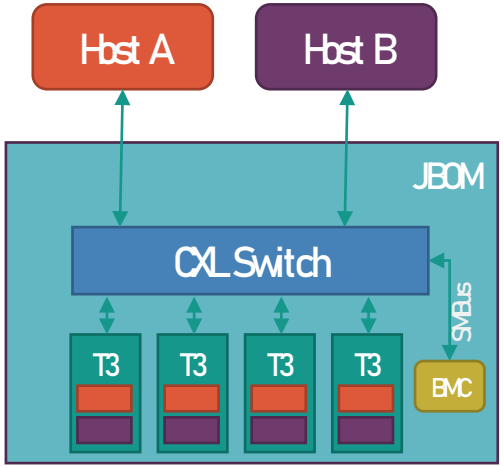
Rack-mount Appliance



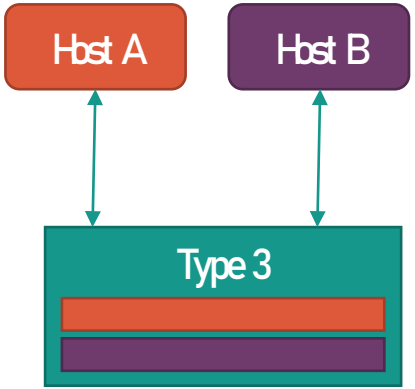
System Management Requirements

New Device Capabilities:

Multi-Logical Device (MLD)



Multi-Headed Devices



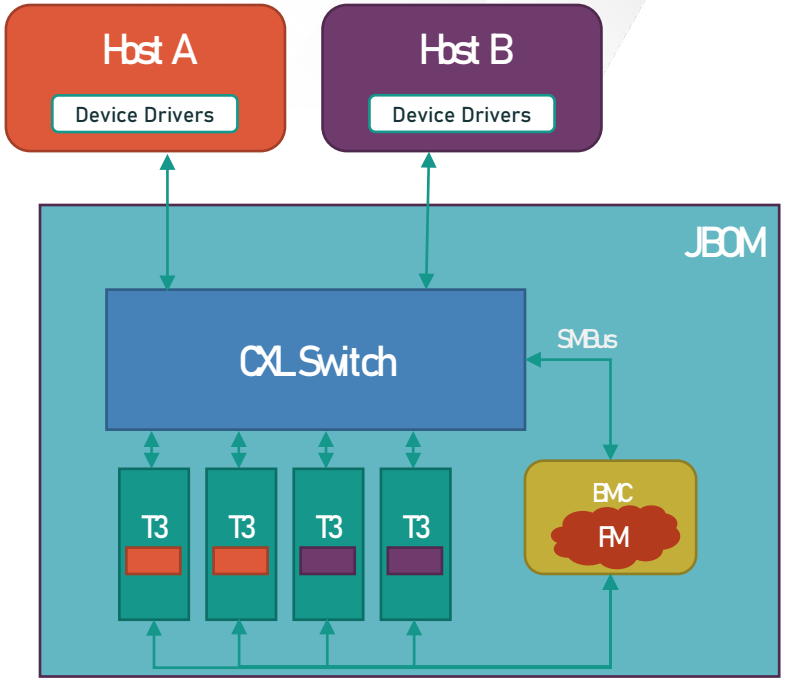
What is a Fabric Manager?

Fabric Manager (FM) is a conceptual term

Refers to the application-specific logic composing systems, allocating pooled resources, managing platforms, etc.

Can take many forms:

- BMC in a rack-mount appliance
- Management software running in a host
- Embedded FW in a CXL Switch



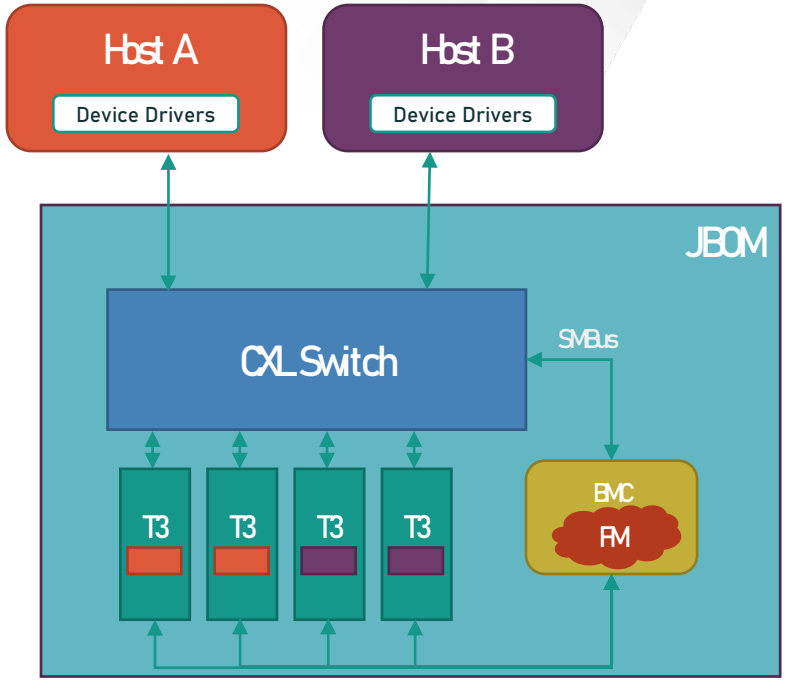
What is a Fabric Manager?

Framework is flexible by design to enable a wide variety of applications (embedded, automotive, hyperscale…)

Most management capabilities are optional

FM is required for advanced system operations:

- Use of MLDs – FM is responsible for assigning LDs to hosts
- Memory pooling – FM is responsible for binding switch ports to host hierarchies



Component Command Interface

Intro to Component Command Interface

- Commands are processed by a Component Command Interface (CCI)
- Two types
  - Mailbox CCI – presented through memory registers
  - MCTP-based CCI – presented as an MCTP EP
- Not a queued interface
- Lengthy operations run as “Background Operations”
- A component may support multiple, with varying capabilities
- Command opcodes are 2B: 1B command set, 1B command
- Supported command list is reported through “Command Effects Log”

Mailbox CCI

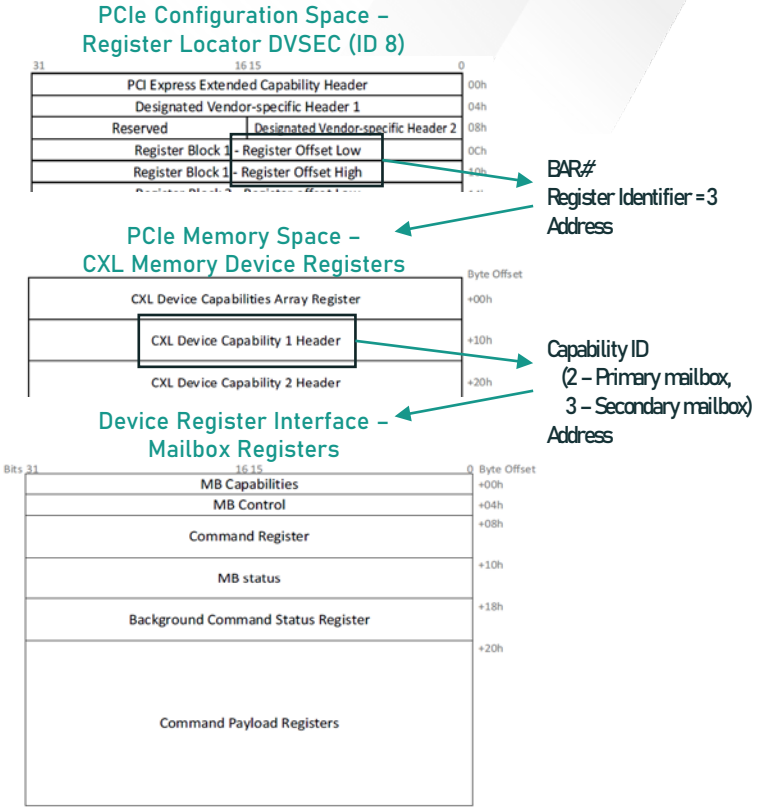
Located in PCIe MMIO Space

Two types of mailbox:

- Primary – designed for use by driver; intended for privileged operations
- Secondary (optional) –designed for log/event record access; no interrupt or background operation support

Command inputs written to Command Payload Registers, outputs read from same region

Optionally generates MSI/MSI-X interrupts



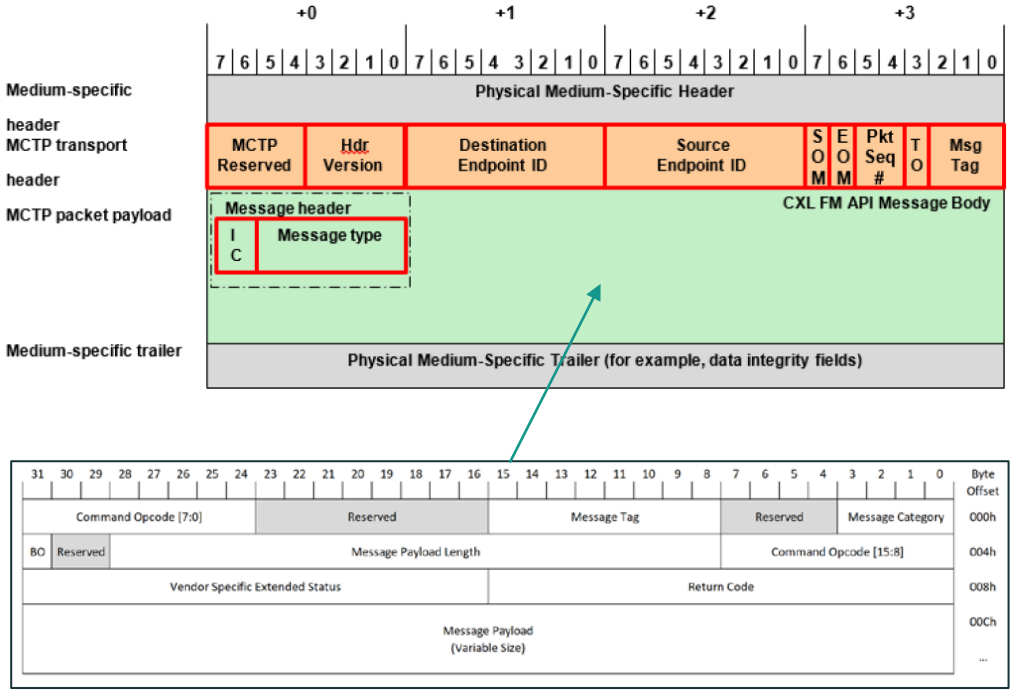
MCTP-based CCI

FM will first discover all MCTP EPs using MCTP spec-defined discovery

CCIs will advertise support for CXL Message Types

- Type 07h for FM API commands
- Type 08h for General and Memory Device Commands

Supported over any physical interface for which an MCTP binding spec is defined

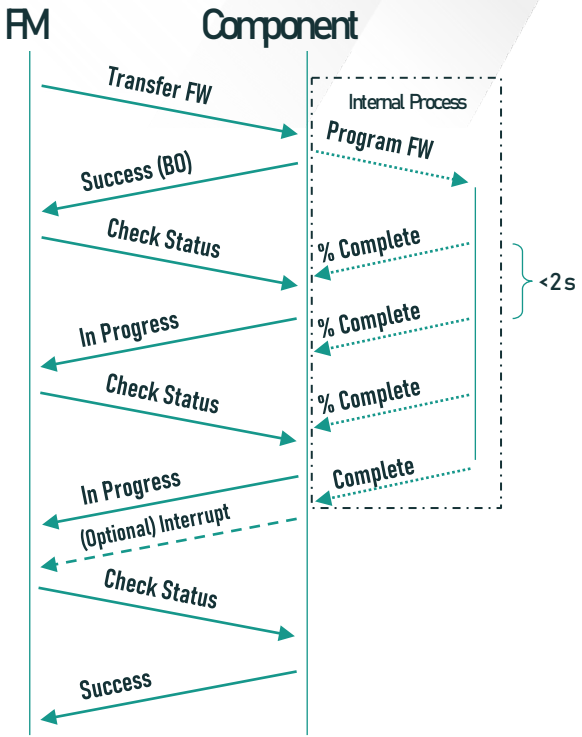


Background Operations

Time-consuming management operations are defined as “Background Commands”

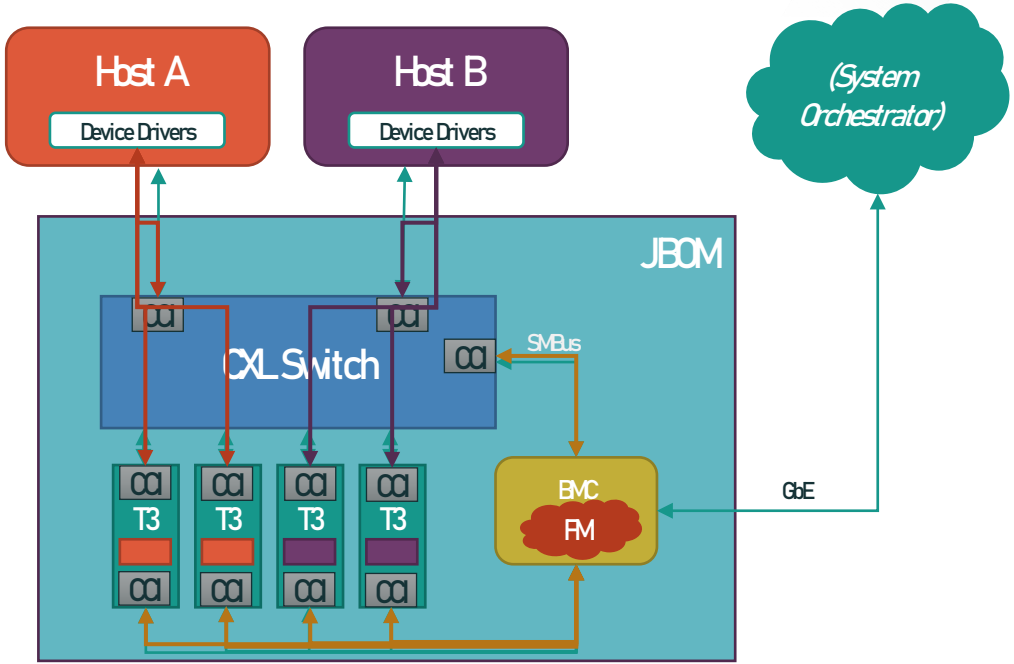
Up to one per CCI supported, but recommended to only support those commands on one interface

Command receives immediate response to indicate BO has started



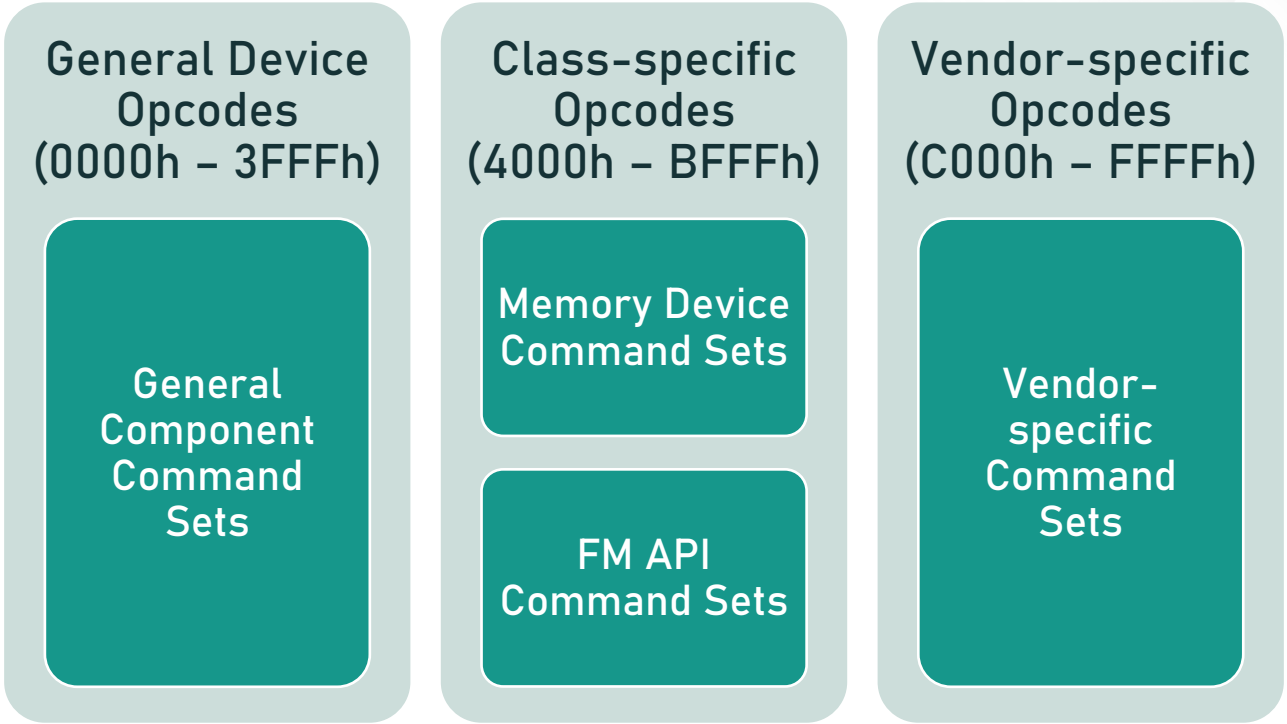
Management Architecture

Example Rack-mount Appliance with Spec Terminology:



Management Command Sets

Command Set Categories



General Component Command Sets

Uses command opcodes 0000h to 3FFFh

Applicable to all classes of devices (Type 1/2/3 devices and switches)

Includes generic capabilities used in the discovery and management of all classes of devices

Information and Status (00h)

- Identify and CCI status

Events (01h)

- Read/clear event records and interrupt configuration

Firmware Update (02h)

- Program and activate

Timestamp (03h)

Logs (04h)

- Supported command list (Command Effects Log)

General Component Command Sets

Table 173. CEL Entry Structure

Byte Offset	Length	Description
0	2	**Opcode**: The command opcode.
2	2	**Command Effect:** Bit mask containing one or more effects for the command opcode • Bit [0]: **Configuration Change after Cold Reset** - When set, this opcode makes a driver visible change to the configuration of the device or data contained within persistent memory regions of the device. The change does not take effect until a device cold reset. • Bit [1]: **Immediate Configuration Change** - When set, this opcode makes an immediate driver visible change to the configuration of the device or data contained within persistent memory regions of the device. • Bit [2]: **Immediate Data Change** - When set, this opcode makes an immediate driver visible change to the data written to the device. • Bit [3]: **Immediate Policy Change** - When set, this opcode makes an immediate change to the policies utilized by the device. • Bit [4]: **Immediate Log Change** - When set, this opcode makes an immediate change to a device log. • Bit [5]: **Security State Change** - When set, this opcode results in an immediate driver visible change in the security state of the device. Security state changes that require a reboot to take effect do not use this effect. • Bit [6]: **Background Operation** - When set, this opcode is executed in the background. • Bit [7]: **Secondary Mailbox Supported** – When set, submitting this opcode via the secondary mailbox is supported, otherwise this opcode will return Unsupported Mailbox or CCI if issued on the secondary mailbox. Only valid when returned on the primary or secondary mailbox. This bit is reserved if the CEL is being returned from any CCI other than the primary or secondary mailbox. • Bits[15:8]: Reserved, shall be set to zero.

Information and Status (00h)

- Identify and CCI status

Events (01h)

- Read/clear event records and interrupt configuration

Firmware Update (02h)

- Program and activate

Timestamp (03h)

Logs (04h)

- Supported command list (Command Effects Log)

Memory Device Command Sets

Uses command opcodes 4000h to BFFFh

Applicable to Type 2/3 devices

Includes all commands specific to management of memory media

Used by System FW during boot and kernel drivers after boot

- **Identify (40h)**
  - Identify memory device capabilities
- **Capacity Config and Label Storage (41h)**
  - Manage labels for persistent memory
- **Health Info and Alerts (42h)**
  - Media state, temperature, health alerts
- **Media and Poison Management (43h)**
- **Sanitize (44h)**
  - Secure clearing of memory
- **Persistent Memory Data-at-Rest Security (45h)**
  - Set security parameters, lock, unlock, etc.
- **Security Passthrough (46h)**
  - Passthrough for SFSC commands
- **SLD QoS Telemetry (47h)**

FM API Command Sets

Uses command opcodes 4000h to BFFFh

Applicable to CXL switches and MLDs

Includes binding commands, LD assignment, and port control

Used by FM to manage switch-attached, disaggregated resources

Physical Switch (51h)

- Identify, port status, port resets

Virtual Switch (52h)

- Binding and unbinding in multi-VCS switches

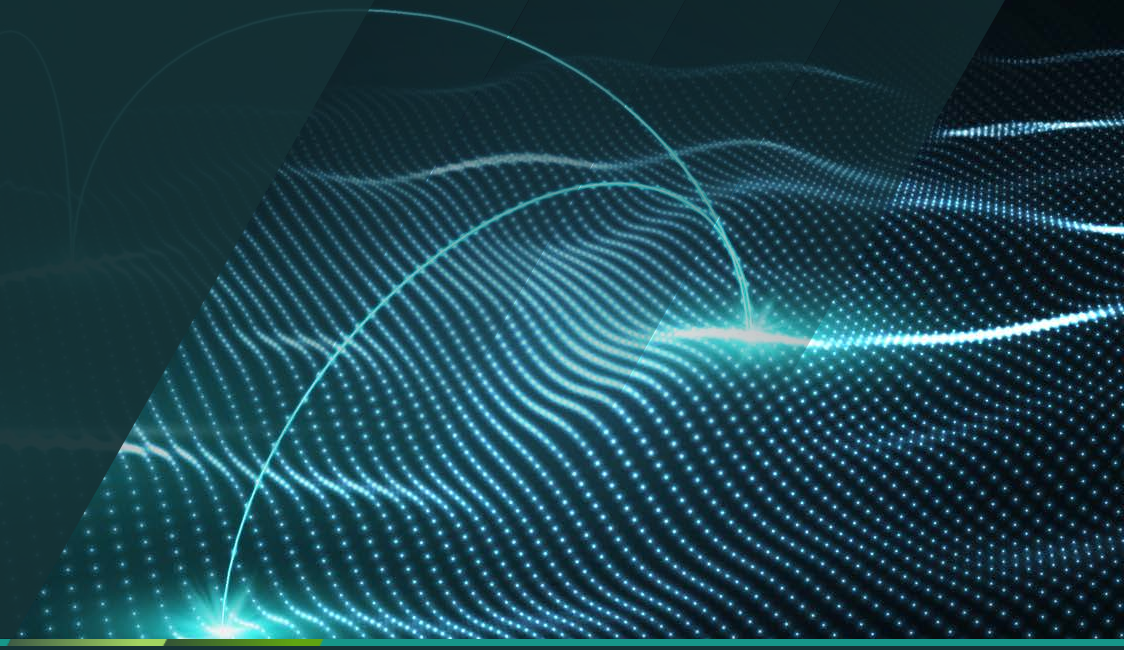
MLD Port (53h)

- Command tunneling

MLD Component (54h)

- Capacity allocation and QoS management

MLD Management



MLD Management

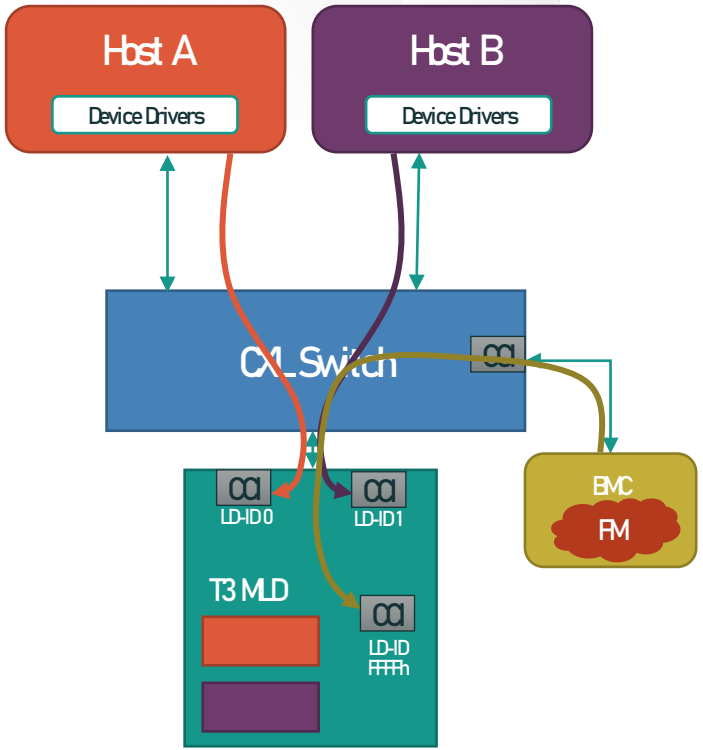
MLDs are accessed by LD-ID, with up to 16 LDs available to hosts (0h to Fh)

LD-ID FFFFh is mandatory and reserved as the 'FM-owned LD', a management target with no memory resources

FM-owned LD is only .io accessible, as .mem and .cache only include 4 LD-ID bits

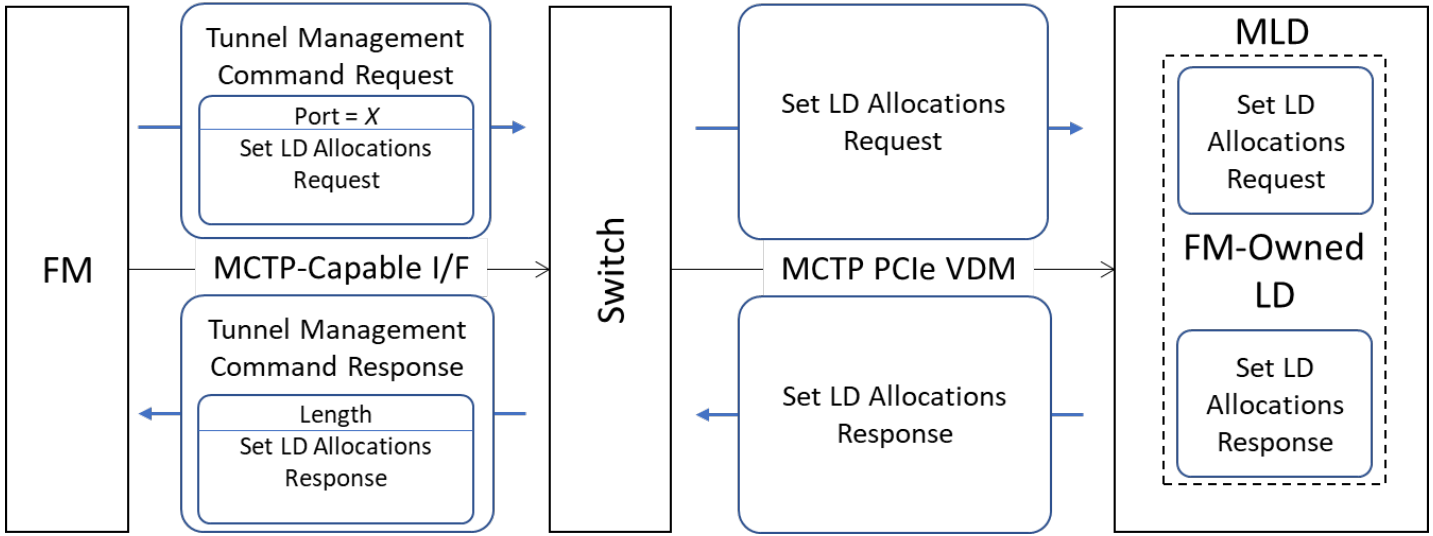
MLD must implement a CCI for each LD plus one for the FM-owned LD

FM may tunnel commands to MLDs through switch, as needed



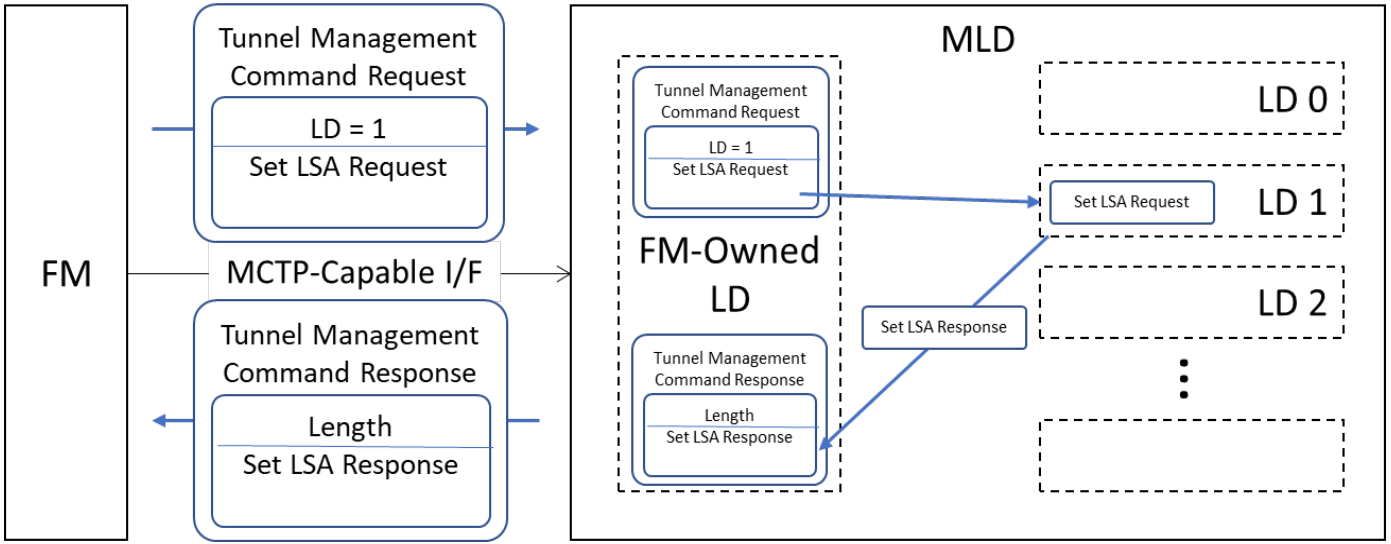
Command Tunneling

Configuring FM-owned LD through a switch:



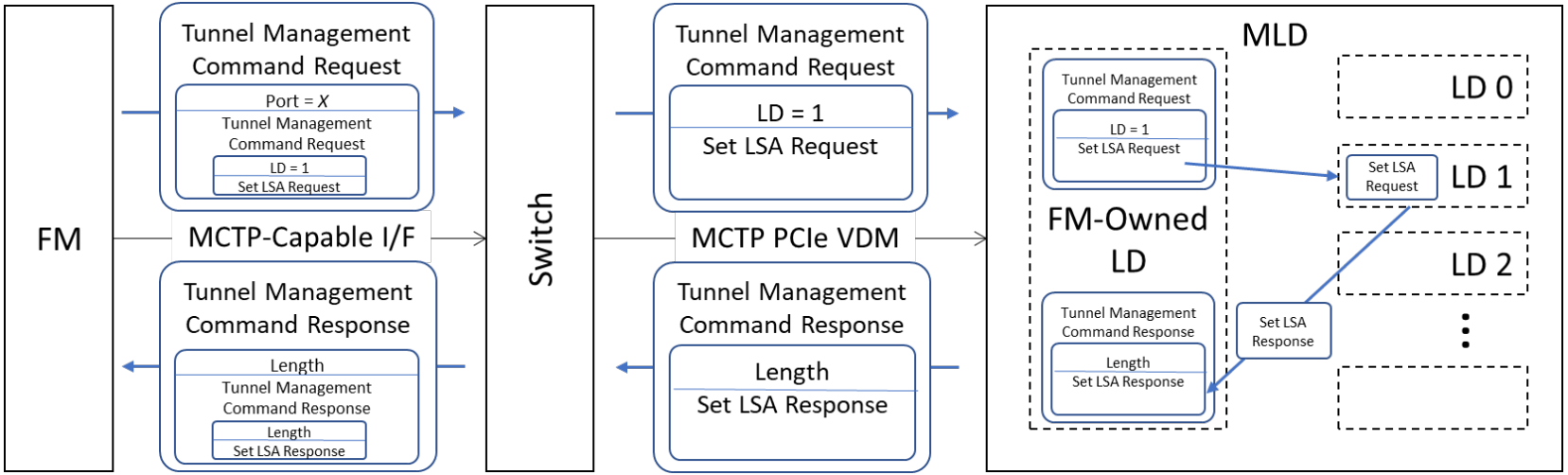
Command Tunneling

Provisioning individual LDs via tunneling:

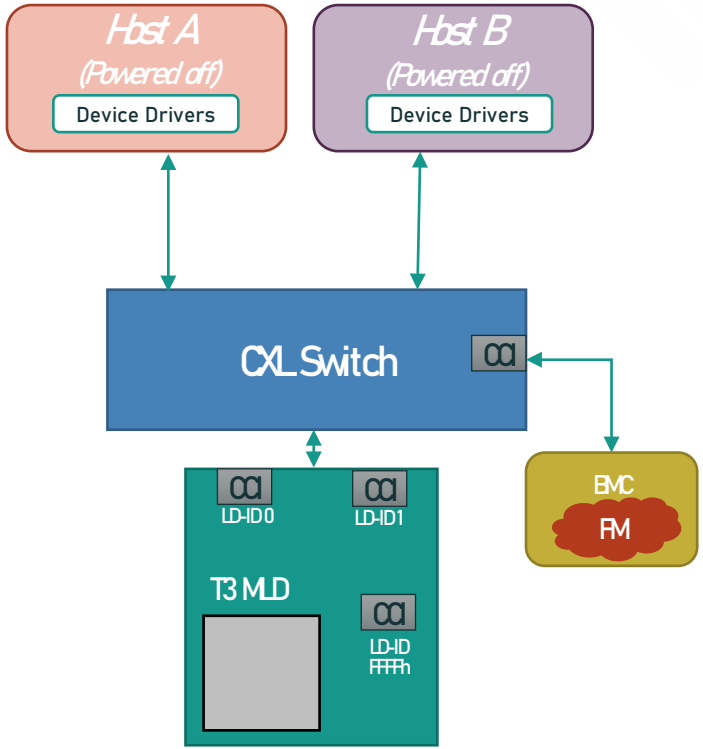


Command Tunneling

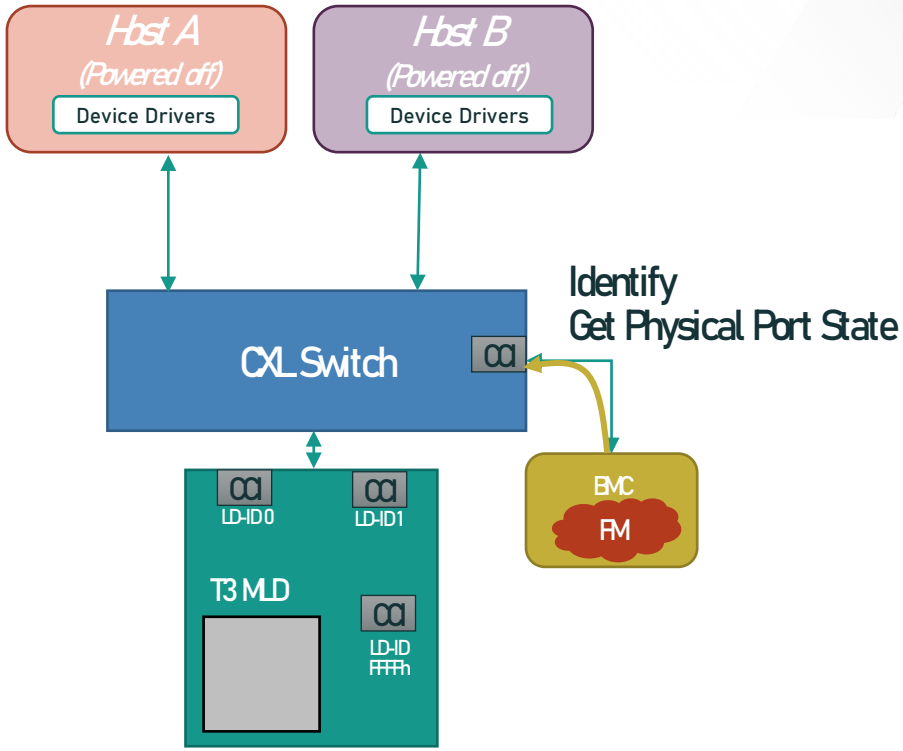
Provisioning individual LDs via tunneling through a switch:



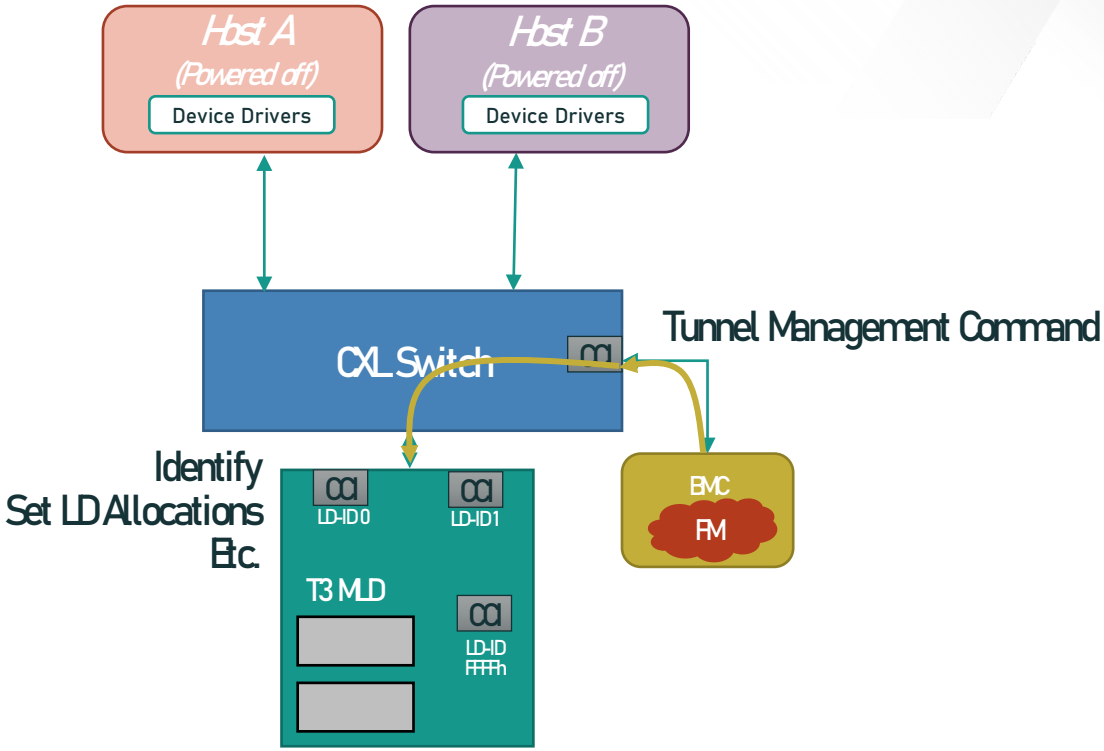
MLD Management



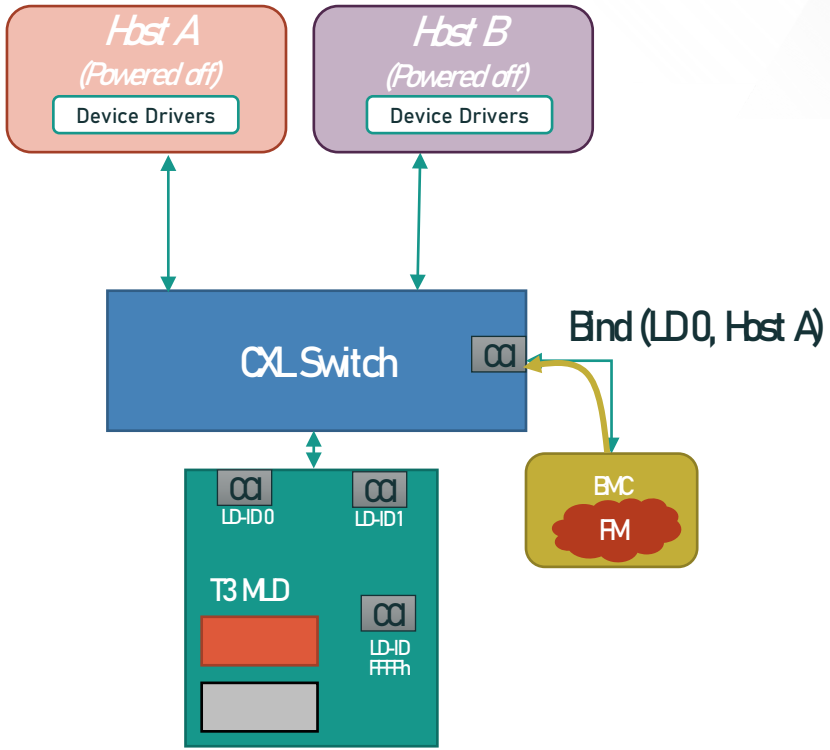
MLD Management



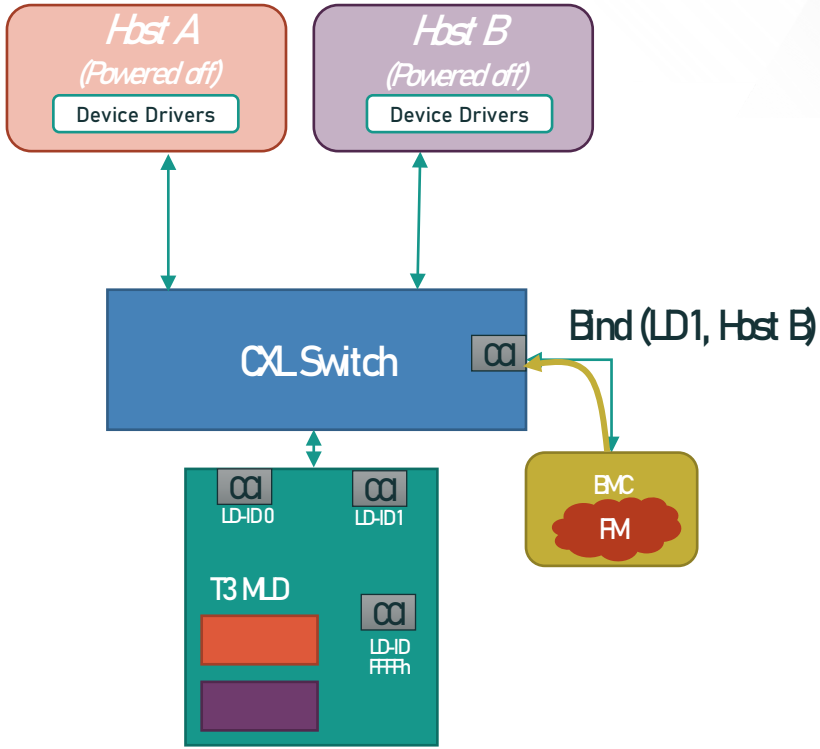
MLD Management



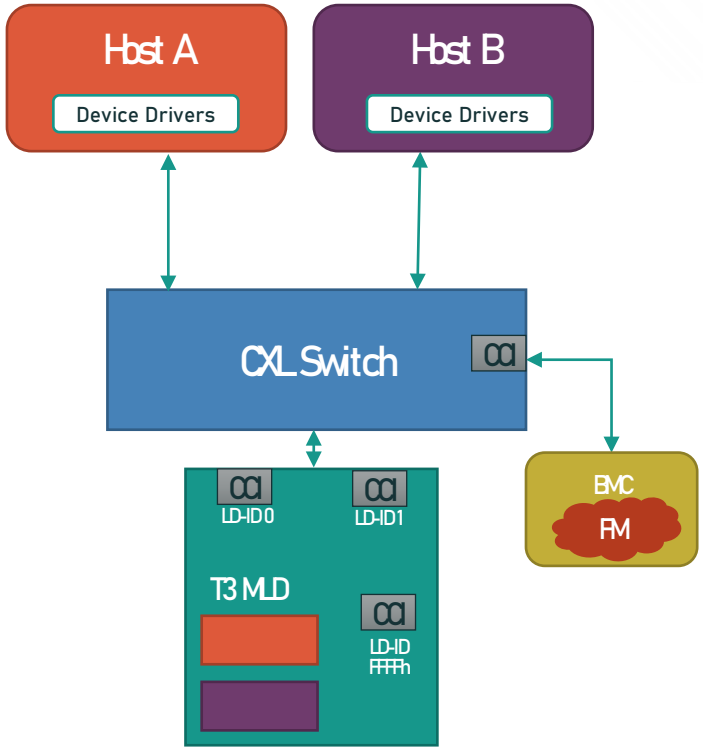
MLD Management



MLD Management



MLD Management



Summary

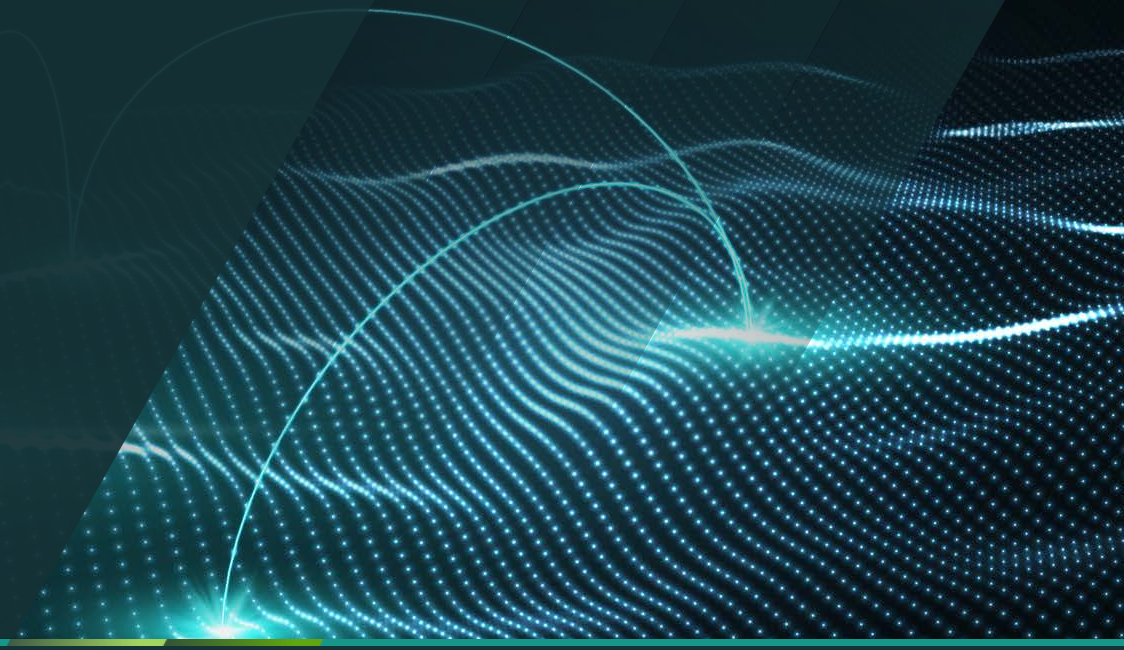
Summary

Key concepts:

- Flexibly-defined architecture to serve variety of applications
- Management available over many interfaces
- Fabric Manager (FM) – any logic initiating management commands
- Component Command Interface (CCI) – management command target in components
- Management Command Sets
- MLD Management

Q&A

Please share your questions in the Question Box



Thank You