



OPEN

Compute Project

Composable Memory Fabric Management Software and APIs Architecture

V1.1

Authors:

Hongjian Fan, Joe Skazinski and Mohamad El-Batal, Seagate Technology

Table of Contents

1. License	5
1.1 Open Web Foundation (OWF) CLA	5
1.2 Acknowledgements.....	6
2. Compliance with OCP Tenets	7
2.1 Openness	7
2.2 Efficiency	7
2.3 Impact.....	7
2.4 Scale.....	7
2.5 Sustainability.....	7
3. Change Log	8
4. Scope	9
5. Overview.....	10
5.1 Base Specification	10
5.2 System Fabric Topology	10
6. Management Interface	12
6.1 RESTful API Implementation and HTTP Methods	12
6.2 HTTP Status Codes	13
6.3 Roles and Privileges	14
6.4 Seagate CFM API Resources	15
6.5 Redfish CFM API Resources	16
6.6 Composable Memory Appliance Redfish API Resources	18
6.7 Host CFM Redfish Resources.....	21
7. Use Cases	23
7.1 Adding Endpoints into Memory Fabric Management System	23
7.2 Compose Pooled Region	24
7.3 Compose Interleaved Pooled Memory Region	25
8. Design Example.....	26
9. References:	28
Appendix A - Checklist for IC approval of this Specification (to be completed by contributor(s) of this Spec).....	29
Appendix B - Contribution Process FAQs.....	30
Appendix C - Seagate CFM API resource definition	31

Table of Figures

Figure 1. CXL Composition Fabric Top Level Diagram.....	9
Figure 2. CFM System Topology.....	10
Figure 3. Fabric Memory Management Communication Path.....	11
Figure 4. CFM API Connection	12
Figure 5. Seagate CFM API Model.....	15
Figure 6. Redfish CFM API Model.....	17
Figure 7. MH-MLD Device with Static Binding	18
Figure 8. Memory Appliance Redfish Model.....	20
Figure 9. CXL Host Service Redfish Model	22
Figure 10. CFM software design architecture example.....	27

Table of Tables

Table 1. Change Log Table.....	8
Table 2. HTTP Status Codes for CMA Fabric Management System	13
Table 3. Management Role Groups Definition.....	14
Table 4. Adding Endpoints	23
Table 5. Compose Memory Region.....	24
Table 6. Compose Interleaved Pooled Memory Region	25
Table 7. References.....	28
Table 8. Appendix A checklist	29

1. License

1.1 Open Web Foundation (OWF) CLA

Contributions to this Specification are made under the terms and conditions set forth in Modified OWF-CLA-1.0.2 (As of June 1, 2023) ("Contribution License") by:

Seagate Technology LLC

Usage of this Specification is governed by the terms and conditions set forth in **Modified OWFa1.0.2 Final Specification Agreement (FSA) (As of June 1, 2023) ("Specification License")**.

You can review the applicable Specification License(s) referenced above by the contributors to this Specification on the OCP website at <http://www.opencompute.org/participate/legal-documents/>. For actual executed copies of either agreement, please contact OCP directly.

Notes:

- 1) The above license does not apply to the Appendix or Appendices. The information in the Appendix or Appendices is for reference only and non-normative in nature.

NOTWITHSTANDING THE FOREGOING LICENSES, THIS SPECIFICATION IS PROVIDED BY OCP "AS IS" AND OCP EXPRESSLY DISCLAIMS ANY WARRANTIES (EXPRESS, IMPLIED, OR OTHERWISE), INCLUDING IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR A PARTICULAR PURPOSE, OR TITLE, RELATED TO THE SPECIFICATION. NOTICE IS HEREBY GIVEN, THAT OTHER RIGHTS NOT GRANTED AS SET FORTH ABOVE, INCLUDING WITHOUT LIMITATION, RIGHTS OF THIRD PARTIES WHO DID NOT EXECUTE THE ABOVE LICENSES, MAY BE IMPLICATED BY THE IMPLEMENTATION OF OR COMPLIANCE WITH THIS SPECIFICATION. OCP IS NOT RESPONSIBLE FOR IDENTIFYING RIGHTS FOR WHICH A LICENSE MAY BE REQUIRED IN ORDER TO IMPLEMENT THIS SPECIFICATION. THE ENTIRE RISK AS TO IMPLEMENTING OR OTHERWISE USING THE SPECIFICATION IS ASSUMED BY YOU. IN NO EVENT WILL OCP BE LIABLE TO YOU FOR ANY MONETARY DAMAGES WITH RESPECT TO ANY CLAIMS RELATED TO, OR ARISING OUT OF YOUR USE OF THIS SPECIFICATION, INCLUDING BUT NOT LIMITED TO ANY LIABILITY FOR LOST PROFITS OR ANY CONSEQUENTIAL, INCIDENTAL, INDIRECT, SPECIAL OR PUNITIVE DAMAGES OF ANY CHARACTER FROM ANY CAUSES OF ACTION OF ANY KIND WITH RESPECT TO THIS SPECIFICATION, WHETHER BASED ON BREACH OF CONTRACT, TORT (INCLUDING NEGLIGENCE), OR OTHERWISE, AND EVEN IF OCP HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.2 Acknowledgements

The Contributors of this Specification would like to acknowledge the following companies for their feedback:

- **Penguin Solutions** on the feedback to the early versions of the software.

2. Compliance with OCP Tenets

2.1 Openness

The specification of the Composable Memory Fabric Management software and APIs defines the minimum APIs used for managing composable memory fabric system backed by OCP compliant composable memory appliance. The APIs defined are fully open for building, modify or personalize the fabric management system. The architecture of the related management system and the hardware implementation are not constraint to the APIs. A reference design for the composable memory fabric management system will be provided. Further improvement can be contributed to this spec in later revisions for more advanced features.

2.2 Efficiency

The specification of the Composable Memory Fabric Management software and APIs offers an efficient method for memory appliance's management need and simplifies user's workflow when using memory appliance.

2.3 Impact

The specification of the Composable Memory Fabric Management software and APIs is designed for memory appliance which is a scalable memory device based on CXL technology. Proper fabric management is critical for the utilization rate of the memory pool supplied by the memory appliance.

2.4 Scale

As a management API specification, the APIs are designed to be easily implemented and easily deployed. Software implementation following this specification should be designed with features for convenient maintenance, upgrade, error reporting and management.

2.5 Sustainability

Using a common API for composable memory fabric system between different vendors could help reduce redundant work in defining the APIs and workflow, simplify the design process and improve the efficiency during deployment. Using fabric management system can also improve hardware utilization and reduce stranded memory for memory appliance.

3. Change Log

Date	Version #	Author	Description
9/6/2024	0.9	Hongjian Fan	Initial draft
9/24/2024	1.0	Kanika Arya	Formatting and Structure changes
11/14/2024	1.1	Hongjian Fan	Various minor corrections

Table 1. Change Log Table

4. Scope

Figure 1 shows the top-level view of a CXL composition fabric system, which contains Composable Memory Appliance, CXL host clients, Composable Fabric Management software and optional Orchestration Applications.

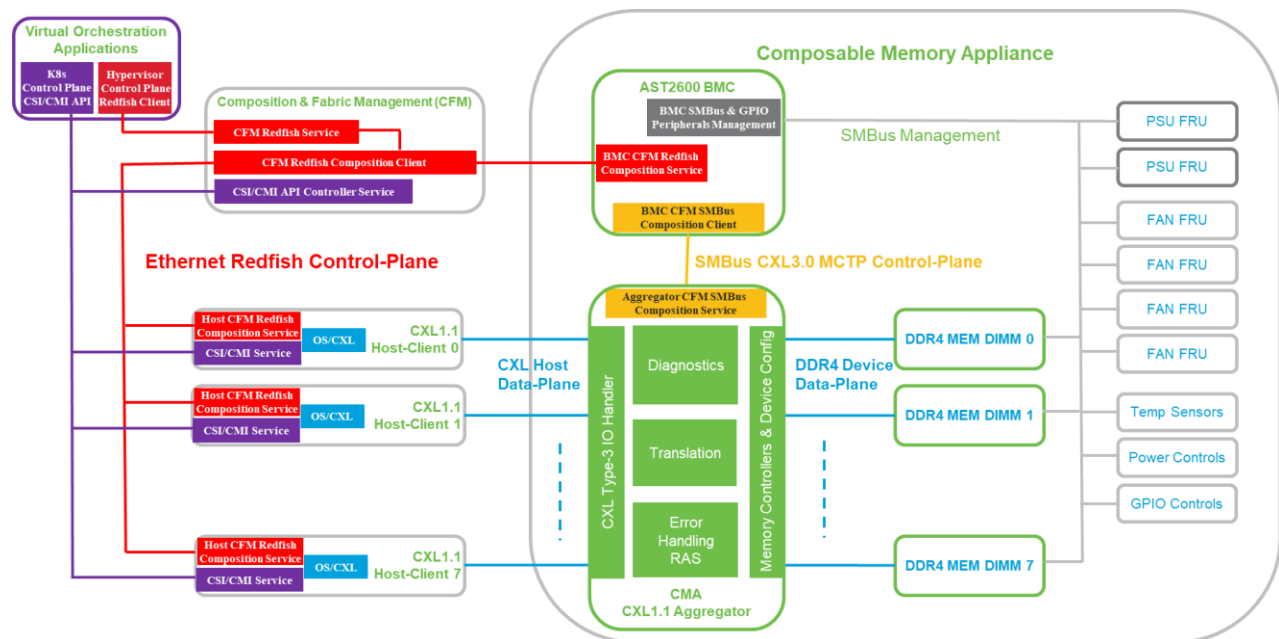


Figure 1. CXL Composition Fabric Top Level Diagram

This document defines the open composable fabric management (CFM) software to provide a common interface with APIs for the composable memory appliance (CMA) composition and management. This includes:

- Fabric system description
- API definition
 - Seagate CFM APIs for general management
 - Redfish CFM APIs for Orchestration Application automation
 - BMC CFM Redfish APIs for CMA's BMC
 - Host CFM Redfish APIs for host client
- Use cases
- Example software design

This specification provides a common interface with minimum APIs to utilize a memory appliance system. Further improvement could be contributed to the specification by participated parties.

5. Overview

Memory appliance is a composable memory system based on CXL fabric solution. This API specification is to provide the clarifications and implementation guide for the management interface for the memory appliance. In the rest of this chapter, the basic system architecture of a memory appliance will be described to show the fabric topology and explain the terminologies defined for the memory appliance to build up the API.

5.1 Base Specification

This specification offers management API based on OCP - Composable Memory Appliance (CMA) Base Specification.

5.2 System Fabric Topology

A memory appliance system contains one or multiple memory appliances, multiple CXL capable host servers, a service server. Each endpoint inside the memory appliance system connects to the same local area network. A memory appliance contains multiple memory blades, and each blade can individually connect to multiple CXL host servers.

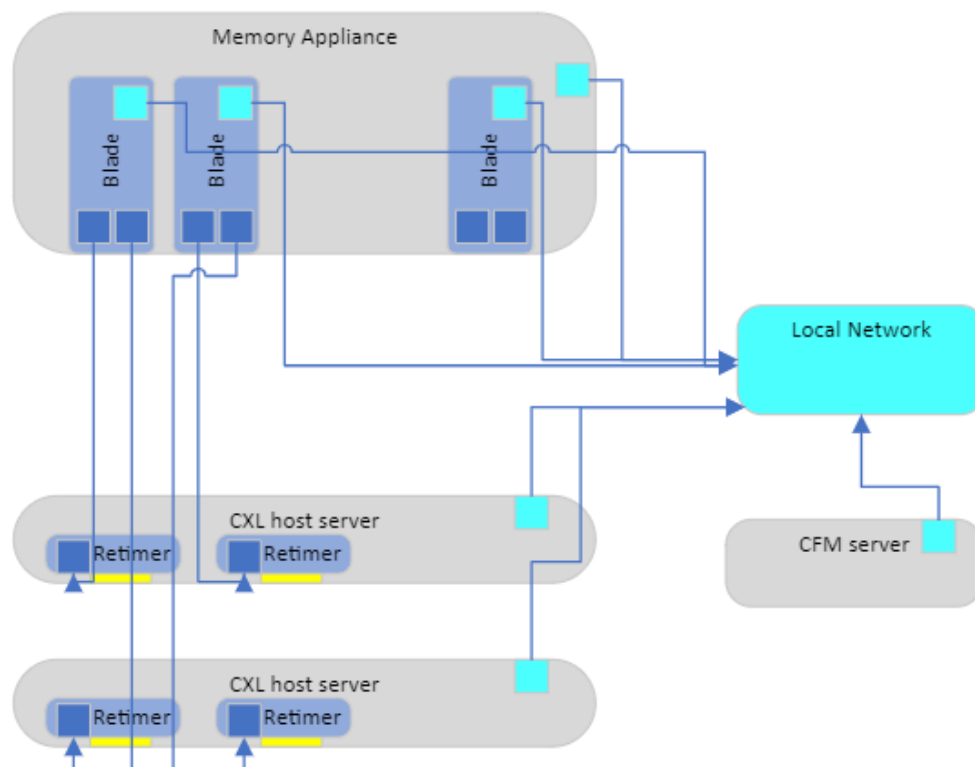


Figure 2. CFM System Topology

Figure 2. CFM System Topology illustrates the connectivity of an example memory appliance system. The fabric management software runs on a server to provides the service. Device manufacture might implement the management interface as part of the memory appliance or provide individual management interface on each memory blade. The CXL host server needs to provide CXL device related information back to the CFM service with Redfish interface based on CXL device schema. This could be done either from the server's BMC or a host application installed on the server as system service.

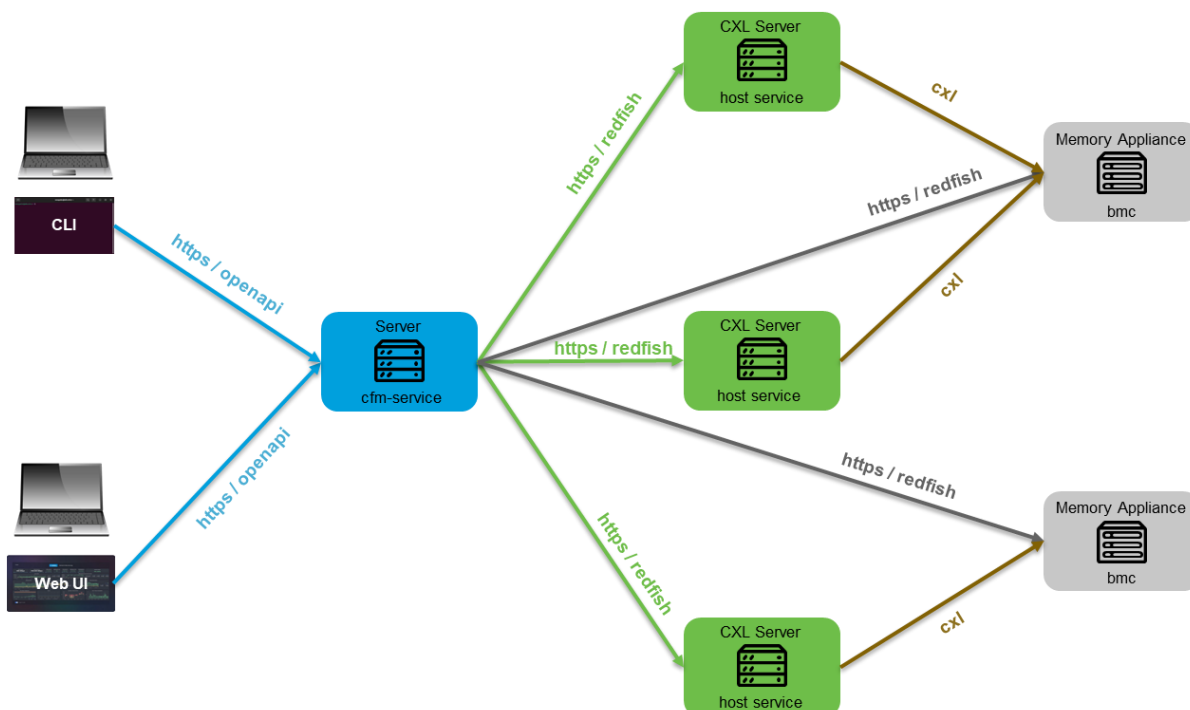


Figure 3. Fabric Memory Management Communication Path

Figure 3 shows the communication path inside a memory appliance system. CXL host servers connects to memory appliance could exchange information with the CXL mailbox CCI defined in CXL2.0 or later specifications. The CFM service communicates to the memory appliances and the cxl host service through https network protocol and using redfish interface. The memory appliance would present the switch resources for management, which will be explained in detail in the later section. The cxl host service should present the CXL device model based on DMTF's CXL redfish schemas. The CFM service provides OpenAPI to end users who is managing the composable memory system via https protocol. End user might further utilize CLI tool or integrated webUI to simplify the management.

6. Management Interface

The control plane of the memory appliance system is built on the network communication. Regardless of the communication between end users and the fabric management service, or the service to query and control the hardware, it should all be based on HTTPs protocol with RESTful APIs. There are four interfaces used in the management system. Two of the interfaces are user facing, which is the interface between end user and the management service. The *Seagate CFM APIs* are OpenAPI complaint RESTful APIs, designed based on the hardware topology and aimed to provide simple management interface for general operations. We have also designed CLI and WebUI tools to leverage the Seagate CFM APIs for improved user experience. The *Redfish CFM APIs* are Redfish APIs based on the Redfish design and how CXL devices should be mapped into redfish structures. The other two APIs are the interfaces between the CFM software to the endpoints to be managed and are both Redfish APIs. The *BMC CFM Redfish APIs* support the communication between the management service and the composable memory appliance, and the *Host CFM Redfish APIs* support the communication between the management service and the host servers. Figure 4 shows how those APIs are used in the management system between the components.

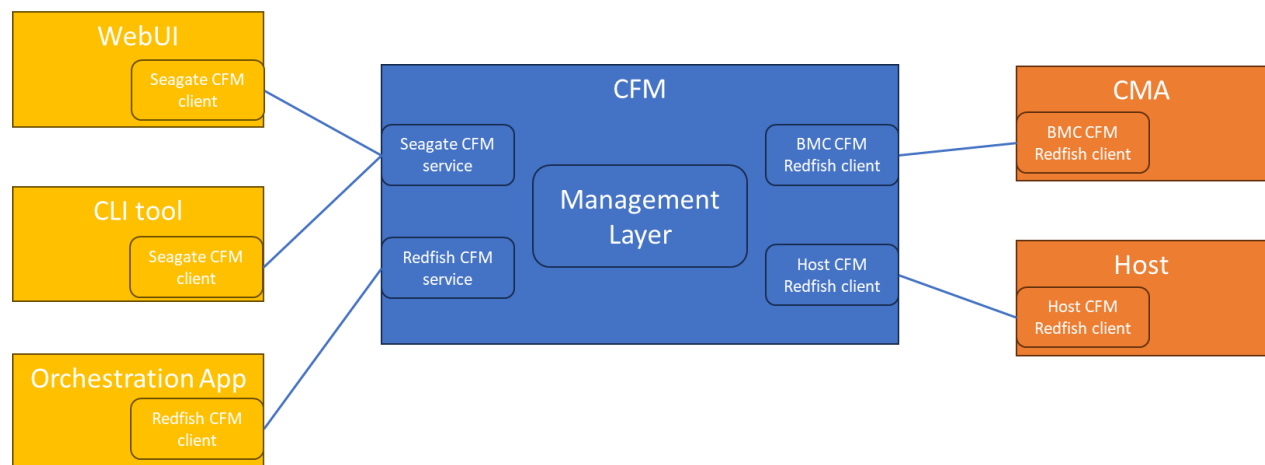


Figure 4. CFM API Connection

6.1 RESTful API Implementation and HTTP Methods

A RESTful API (Representational State Transfer) is an interface that allows two computer systems to securely exchange information over the network. Resources provided by the memory appliance system are presented with REST architecture. HTTP methods GET, PUT, PATCH, DELETE are used in the management system.

6.2 HTTP Status Codes

The following HTTP status codes are used in the CMA fabric management system:

HTTP Status Code	Description
200 OK	Request completed successfully
201 Created	Resource was created and a representation of that resource is returned in the response body
206 Partial Content	Request was partially succeeded. This happens when the request involves with multiple steps and not all the steps finished successful
400 Bad Request	Request could not be processed because it contains invalid information, such as an invalid input field, or is missing a required value
401 Unauthorized	Authentication credentials are required to perform the request
404 Not Found	Request specified a URI of a resource that does not exist
405 Method Not Allowed	The HTTP verb (DELETE, GET, POST, PATCH, HEAD, or PUT) is not supported for the URI requested
409 Conflict	The memory appliance does not have the resources to fulfill the request
500 Internal Server Error	The service encountered an unexpected error that prevented it from fulfilling the request
501 Not Implemented	The service does not currently support this request

Table 2. HTTP Status Codes for CMA Fabric Management System

6.3 Roles and Privileges

Roles are defined for distinguishing different privileges between user groups:

Role	Privilege
Administrator	Add/Modify/Delete Users and their roles, plus the privileges from Operator role
Operator	Add/Remove memory appliance and cxl host endpoints. Compose Memory resources within the CFM system, plus the privileges from read-only role
Read-Only	Retrieve CFM system information

Table 3. Management Role Groups Definition

6.4 Seagate CFM API Resources

This section defines the user facing Seagate CFM API resources on the fabric management service. The Seagate CFM API is designed based on hardware topology and aimed at user convenience.

Appliances, Blades, Hosts, Ports, Memory, and Memory-Devices Resources: Unlike Redfish APIs we have to map the hardware into existing collections like Systems or Chassis, the Seagate CFM API names the API by the hardware.

Figure 5 shows the abstracted structures of the Seagate CFM API model.

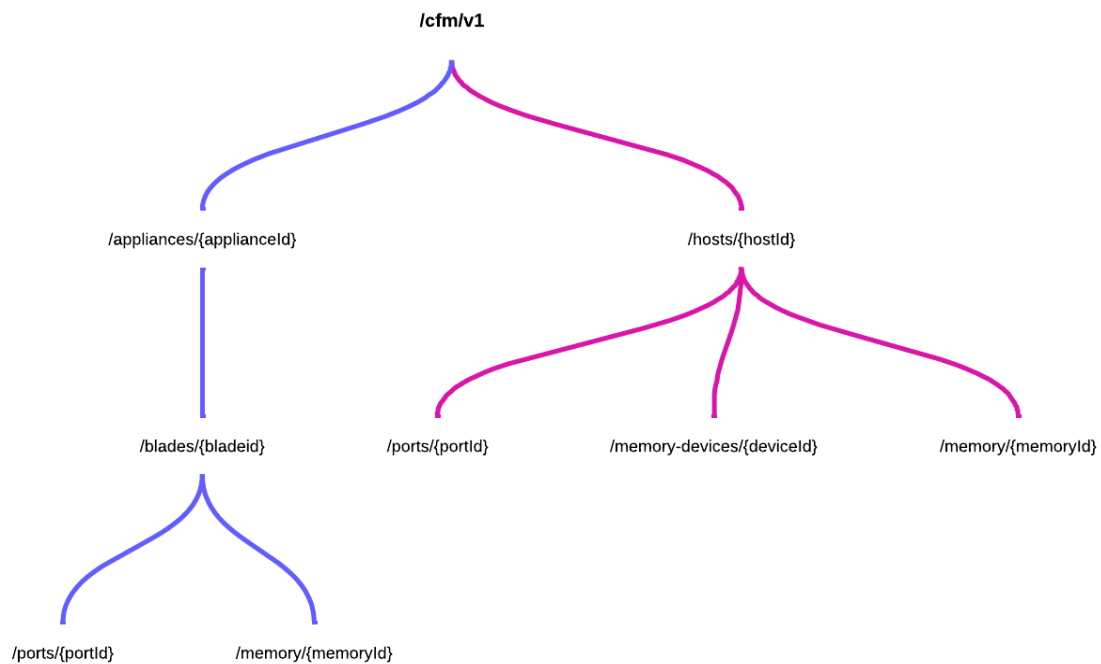


Figure 5. Seagate CFM API Model

6.5 Redfish CFM API Resources

This section defines the Redfish management resources on the composable fabric management software user facing interface.

Systems, Fabrics, and Chassis Resources: The CFM abstracts the endpoints under management into different collections. The memory appliance is mapped to the member in the Systems collection for its hardware information and mapped to the Fabric collection for the composability. The CXL host server is mapped to the member in the Chassis collection.

PCIe® Devices and CXL logical Devices Resources: are resources under Chassis only in this model. It represents the physical CXL devices shown on the PCIe/CXL bus. The CXL logical device also links the memory chunk resources between the memory node on the host server and the allocated memory from the blade.

Memory Domains and Memory Chunks Resources: Under the system resource, the memory domains are the blades inside the enclosure and each blade could be managed independently. The memory Chunks are allocated memory to each of the physical port. On the Chassis side, we hard code the memory domain to “dimm” and “CXL”, and the memory chunks underneath are the Numa nodes from the host.

Memory Resources: Represents the memory DIMM channels on the blade.

Switches, Endpoints, Connections and Ports Resources: The Redfish fabric modelling on the CFM software is similar to the one on the CMA itself, where the Switches represent each blade, and the endpoints are the upstream port which connects the cxl host to the blade and the connections are the composed element to connect the memory abstraction between the host and the blade.

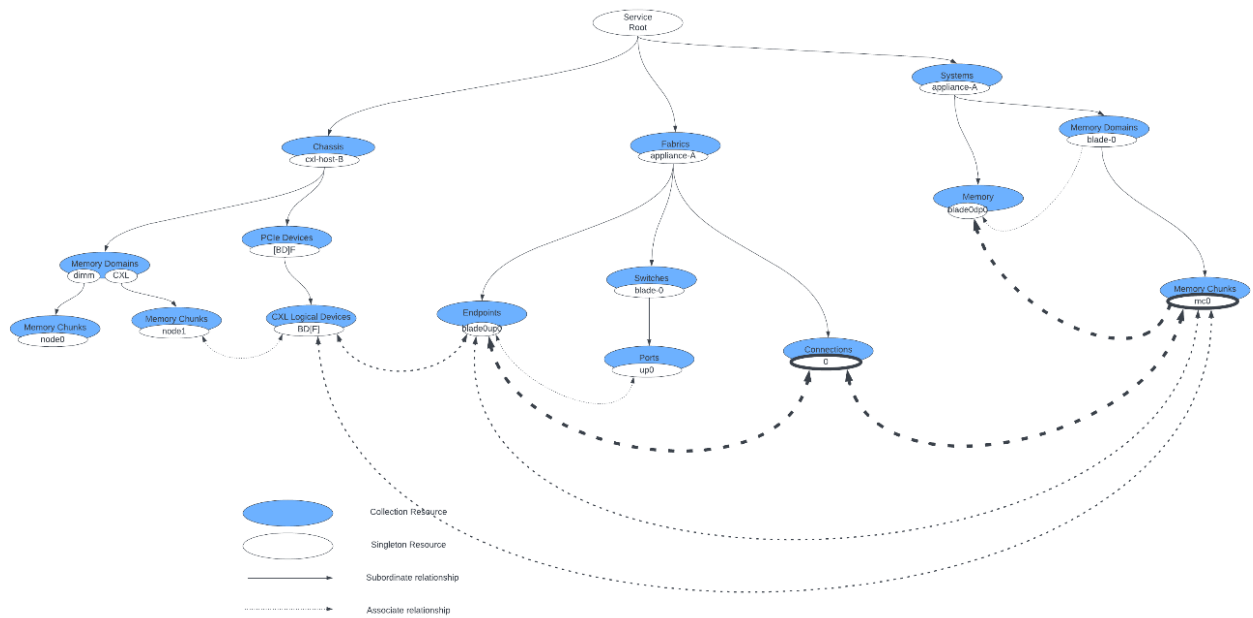


Figure 6. Redfish CFM API Model

Figure 6 is the device model graph for the CFM redfish model. Refer to Redfish schema guide for the detail of each individual schema.

6.6 Composable Memory Appliance Redfish API Resources

This section defines the Redfish management resources on the memory appliance. The memory appliance is treated as MH-MLD device with static binding so we can fit it into the Redfish model.

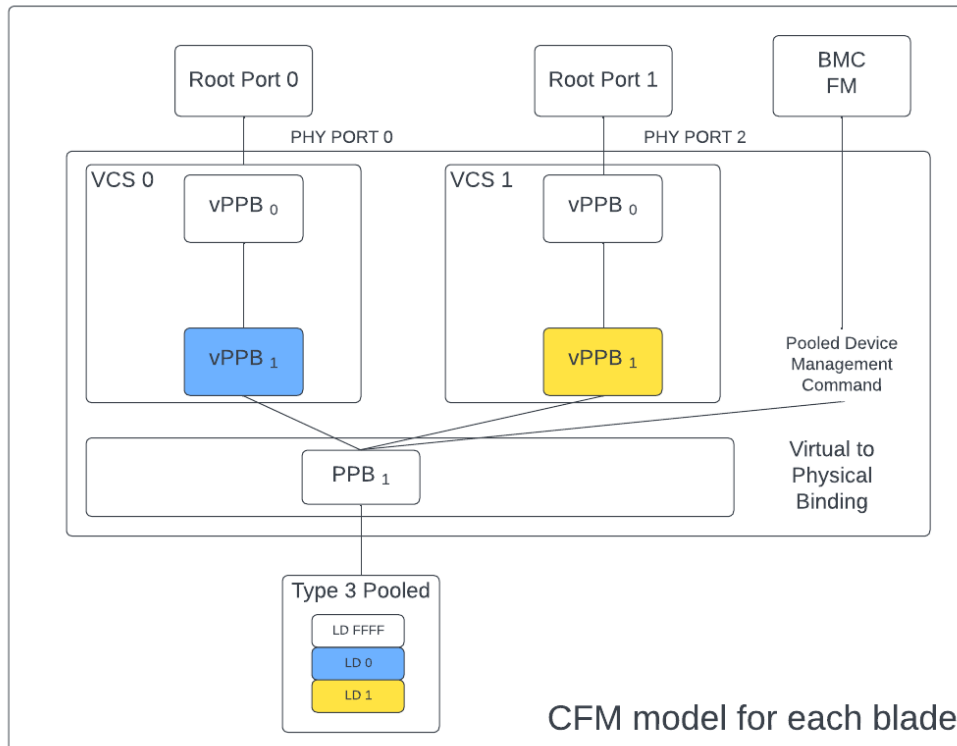


Figure 7. MH-MLD Device with Static Binding

Systems, Fabrics, and Chassis Resources: The memory appliance will host multiple memory blades. The memory appliance could have a Redfish instance for the centralized BMC, or individual Redfish for each memory blade. If centralized BMC is used, then multiple members would be created in the collections for Systems, Fabrics, and Chassis, with each member representing a physical blade. If individual BMC is used, then each BMC will host a Redfish instant at its own IP address and only one instance would show up in the collections for Systems, Fabrics, and Chassis.

Memory Resources: Each memory blade would contain multiple memory channels. Each channel has its own capacity and bandwidth. Channels within the same blade could be interleaved for better performance. Memory appliance should present memory channel as Memory resource under Memory collection.

Resource Block Resources: A memory channel on the blade will be divided into an array of physically continue resource blocks with the same capacity. The capacity of the resource block is

the minimum granularity for memory composition and the composed memory chunk are aligned to the resource block size. Resource block Collection will present under the Composition Service.

Memory Chunk Resources: A memory chunk must be composed by a group of memory resource blocks from the same blade. If multiple resource blocks are used, they could be from a single Memory resource (memory channel) or multiple, but they must be physically continued within the memory channel and be equal in size between different memory channels. Quality of Service (QoS) could be directly configured by interleaving between memory channels.

Fabric Resources: Each upstream port supported by the memory blade, or the physical cable connect point will be abstracted as a Port object in Redfish and each CXL logical device within the upstream port present as an endpoint object. A Single Logical CXL device will have one Endpoint linked to one Port. A Multi Logical CXL device should have multiple Endpoints linked to one Port. Each endpoint represents a logical device, and should be programmed with the logical device ID. A connection object needs to be created to link a memory chunk to an endpoint. Pooled memory should only connect one memory chunk to one endpoint. Shared memory could connect one memory chunk to multiple endpoints.

Figure 8 is the device model graph for the memory appliance. Refer to Redfish schema guide for the detail of each individual schema.

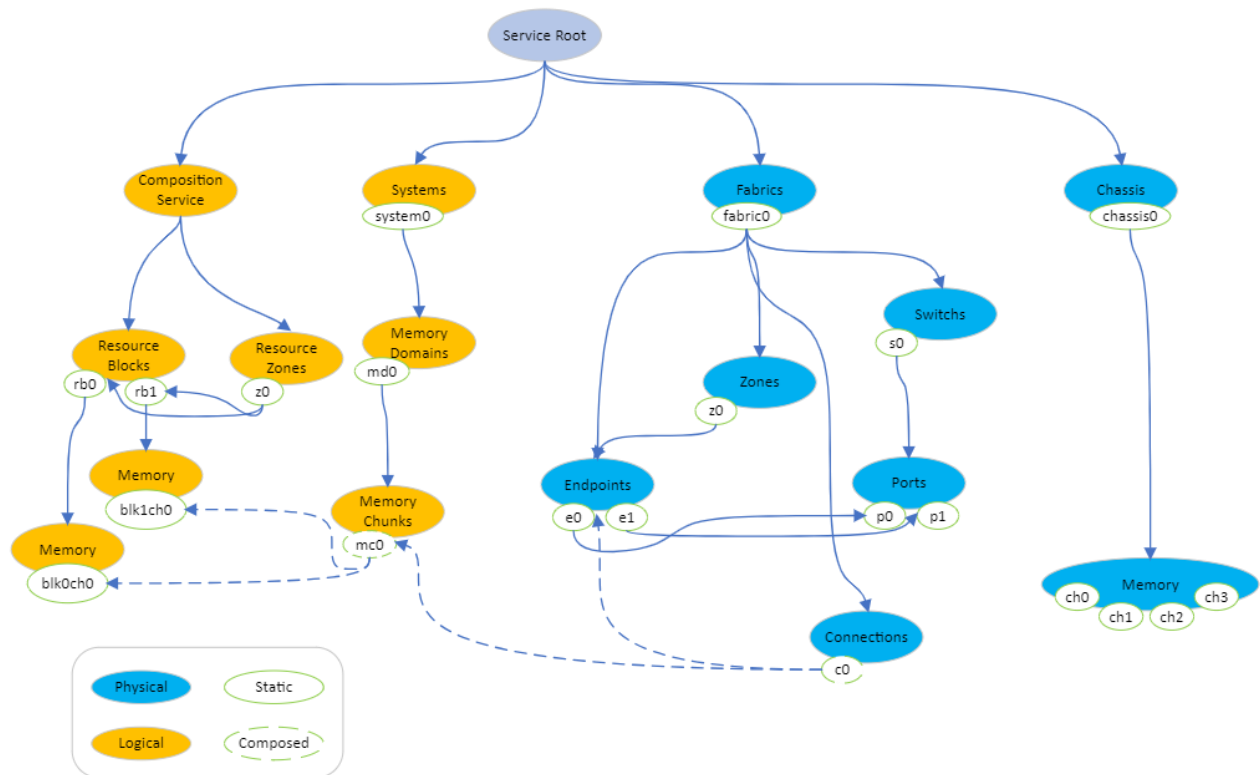


Figure 8. Memory Appliance Redfish Model

6.7 Host CFM Redfish Resources

This section defines the Redfish management resources on the cxl host service. Local CXL Type 3 device model would be applied for CXL memory backed by memory appliance.

Chassis Resources: Chassis contain the physical device information on the server. This includes the CXL device information shared with PCIe, CXL logical device information, memory chunk as a host managed physical memory address range.

Systems Resources: Systems contain the resources abstracted for system and user's view. This could be a NUMA node or a CXL memory enumerated character device.

Memory Chunk Resources: A CXL device memory would be assigned with host managed address in system's physical memory map. However, software should not operate in physical memory space. Either a CPU-less NUMA node, or a CXL IO device would be the idea interface for software if any direct memory access is desired. A memory chunk could be the physical address of a CXL device under chassis' view, or a NUMA node or IO device under system's view.

PCIe Devices and CXL logical Devices: A CXL device is built on top of the PCIe standard and will also show up in the host as a PCIe device. The part a CXL device shares with PCIe standard will be present as the PCIe device resource and unique to CXL will be presents the CXL logical device. A CXL physical device might carry multiple CXL logical devises.

Below is the device model graph for the CXL host service. Refer to Redfish schema guide for the detail of each individual schema.

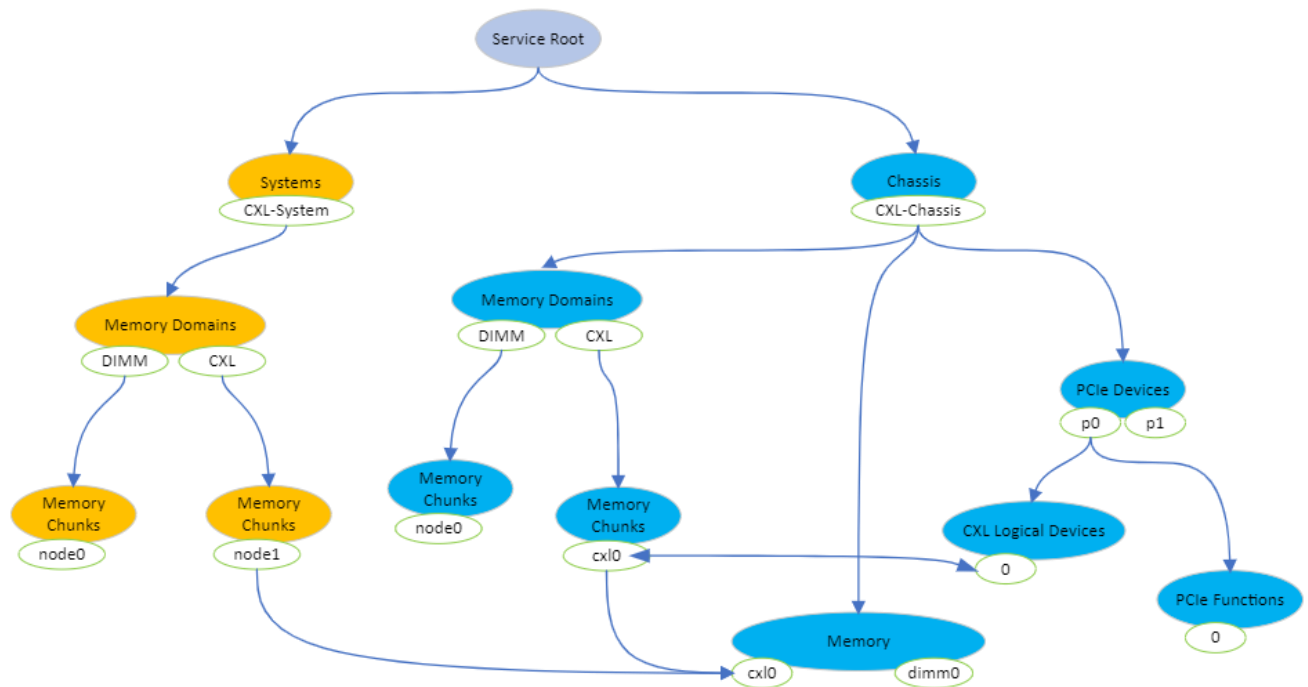


Figure 9. CXL Host Service Redfish Model

7. Use Cases

This section shows a few examples of using the CFM system. The Seagate CFM APIs are used in those examples.

7.1 Adding Endpoints into Memory Fabric Management System

The following steps show the API used to add appliances, blades, and host servers into the CFM system:

Step	Method	API	Notes
Add Appliance	POST	/cfm/v1/appliances	Iterate for each memory appliance to be add. Proper credential needs to be sent with the request body if centralized BMC is used for managing all plugged blades
Add Blade	POST	/cfm/v1/appliances/{applianceId}/blades	Iterate for each memory blade if separate BMC manages each memory blade. Proper credential needs to be sent with the request body
Add Host	POST	/cfm/v1/hosts	Iterate for each server with the cxi host service installed. Proper credential needs to be sent with the request body
View Appliance	GET	/cfm/v1/appliances /cfm/v1/appliances/{applianceId}	View appliance list and individual appliance information
View Blades	GET	/cfm/v1/appliances/{applianceId}/blades /cfm/v1/appliances/{applianceId}/blades/{bladeId}	View blade list within an appliance and individual blade information
View Hosts	GET	/cfm/v1/hosts /cfm/v1/hosts/{hostId}	View cxi host list and individual cxi host information

Table 4. Adding Endpoints

7.2 Compose Pooled Region

The following steps show the API used to compose a memory region for a single host server:

Step	Method	API	Notes
View Blade resource blocks	GET	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/resources/	View existing resources on the blade and make sure resource is available for composition
View Blade ports	GET	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/ports/	Find the correct port linked to the desired cxl host server
Compose memory chunk	POST	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/memory	Post desired size and QoS to create a new memory chunk
Assign memory chunk	PATCH	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/memory/{memoryId}	Assign the newly created memory chunk to the port which is linked to the desired cxl host server
Reboot			Need admin to coordinate the reboot timing to avoid any disrupt
View Hosts	GET	/cfm/v1/hosts /cfm/v1/hosts/{hostId}	Verify that the composed memory shows up on the host

Table 5. Compose Memory Region

7.3 Compose Interleaved Pooled Memory Region

The following steps show the API used to compose a set of memory regions to be interleaved for a single host server:

Step	Method	API	Notes
View Hosts	GET	/cfm/v1/hosts /cfm/v1/hosts/{hostId}	Find the target host server and available ports
View Blade resource blocks	GET	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/resources/	Trace the port to the blade and view existing resources on the blade and make sure resource is available for composition
Compose memory chunk	POST	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/memory	Post desired size and QoS to create a new memory chunk
...			Repeat until all memory region to be interleaved on the host is composed. Interleaved memory region has to be the same in size and QoS
Assign memory chunk	PATCH	/cfm/v1/appliances/{applianceId}/blades/{bladeId}/memory/{memoryId}	Assign the memory chunk to the port which is linked to the first desired cxi host server
...			Repeat the process until all interleave memory regions have been assigned to the desired hosts
Reboot			Need admin to coordinate the reboot timing to avoid any disrupt
View Hosts	GET	/cfm/v1/hosts /cfm/v1/hosts/{hostId}	Verify that the composed memory shows up on the host

Table 6. Compose Interleaved Pooled Memory Region

8. Design Example

This section contains design example for the Composer and Fabric Manager (CFM) software. This software allows users to manage memory appliances and compose memory for CXL host servers. This software service will have the capability to manage multiple memory appliances and multiple CXL hosts. The CFM will consist of a core service with a REST API front-end to support multiple user interfaces and a back-end interface that supports Redfish and plugin backend implementation for other potential future protocols.

REST API layer: Network interface to service the API. End users could use any net tool to send the REST API to this service.

Fabric Manager layer: Handles the business logic for the fabric management.

Backend Interface: Abstracted backend interface. Backend plug-ins are required to communicate to memory appliance or the cxl host service.

Figure 10 shows the example architecture to support multiuser interfaces and multi backend device types.

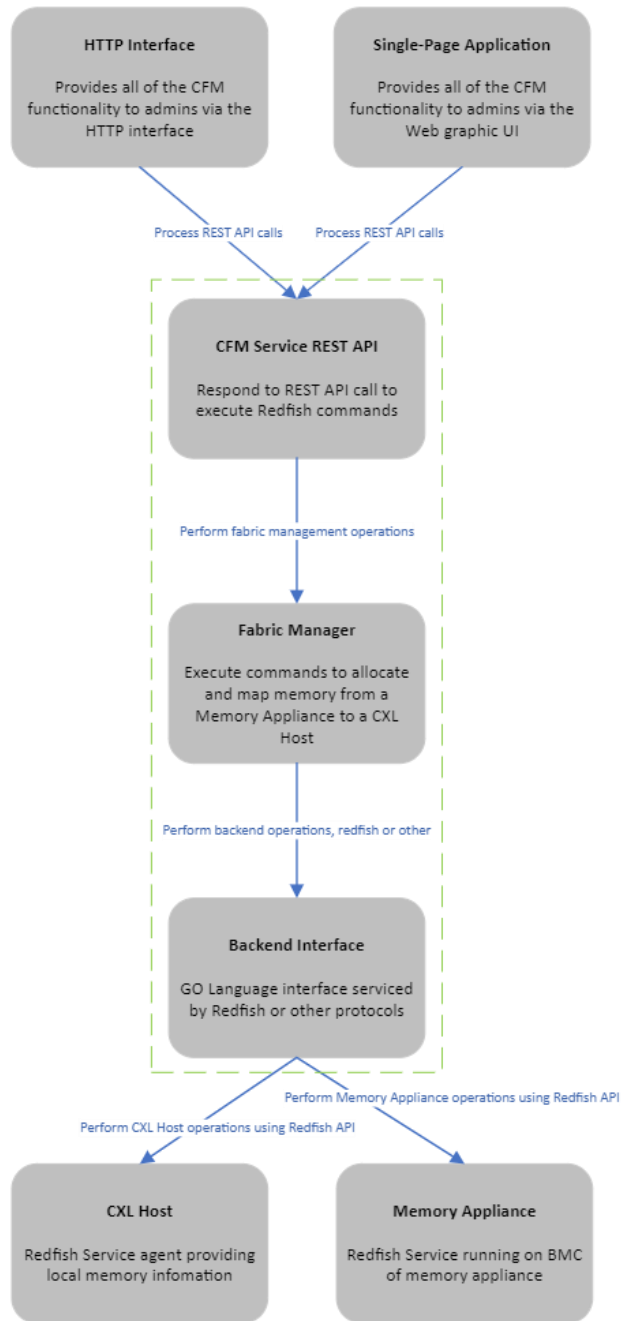


Figure 10. CFM software design architecture example

9. References:

Document:
C4 Model - c4model.com/
CXL Consortium - www.computeexpresslink.org/
OpenAPI Initiative - www.openapis.org/
Redfish Documents - www.dmtf.org/standards/redfish
Redfish CXL Device Management models - https://www.dmtf.org/documents/redfish-spmf/redfish-cxl-device-management-models-bundle-08wip
PCIe 5.0 base spec - https://pcisig.com/specifications/pciexpress
CXL 1.1 spec - https://www.computeexpresslink.org/download-the-specification
OCP - Composable Memory Appliance (CMA) Base Specification

Table 7. References

Appendix A - Checklist for IC approval of this Specification (to be completed by contributor(s) of this Spec)

Complete all the checklist items in the table with links to the section where it is described in this spec or an external document:

Item	Status	Link to detailed explanation
Has this contribution been presented to an OCP Project group during a project call or engineering workshop?		If “No”, please state the reason
Approval by Project Leads		If “No”, please state the reason
Is this contribution entered into the OCP Contribution Portal?		If “No”, please state the reason
Was it approved in the OCP Contribution Portal?		If “No”, please state the reason

Table 8. Appendix A checklist

Appendix B - Contribution Process FAQs

As a contributor to a hardware specification, here are some questions that often come up.

Q1. What type of specification am I contributing to OCP?

- a. **The Design Specification** for a de-facto standard (ex: interface type)
- b. The Design Specification for a new product <product type>
- c. Modification of an existing <type> specification (state which existing spec is being modified) resulting in a revised specification.
- d. **Design specification** (based on an existing Design Specification) with more refined design details.
- e. If none of the above, please contact OCP Staff for better direction.

Q2. How do I know if what I am contributing will be accepted by OCP?

- a. Before contributing any specifications, please contact either OCP Staff (Rob Coyle, Michael Schill) or the Project Lead for the Project that best represents your contribution. They will guide you as to what's the best form for your contribution. Project List [here](#).

Q3. What is the contribution process for my hardware spec?

- a. Follow the flow for your spec type [here](#).

Q4. What if my spec is not developed yet and I want to collaborate with other companies?

- a. Please contact either OCP Staff (Rob Coyle or Michael Schill) or the Project Lead for the Project that best represents your contribution. They will help you find other collaborators and help you with the contribution process for a multi-party contribution.

Q5. I have a question about the Contribution License Agreement (CLA).

- a. Please contact OCP Staff and we can help you with questions.

Appendix C - Seagate CFM API resource definition

This section defines all APIs and their resources in the Seagate CFM API set. For each API, there is a brief description, the URL of the API, the HTTP method supported, and the properties and parameters to the related HTTP method where applicable.

Service Root

Description:

The root resource of the CFM service.

URL:

[/cfm/v1](#)

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	OK

Appliances Collection

Description:

A collection of the managed memory appliances.

URL:

/cfm/v1/appliances

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve the collection of all managed memory appliances. Collection may be empty
POST	Operator	Add a new memory appliance and optional credentials to be managed by CFM service

Supported GET method property:

Property	Type	Description
memberCount	Integer	Number of total members in this collection
members	Array	Array of uri to each of the member in this collection

Supported POST method parameter:

Parameter	Type	Required	Description
username	string	True	The username credentials used to communicate with the service to be added
password	string	True	The password credentials used to communicate with another service
ipAddress	string	True	The IP Address in dot notation of the service with optional port number appended by colon

Supported POST method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
blades	object	Array of memory blades managed by this memory appliance
totalMemoryAvailableGiB	integer	Total memory managed by this memory appliance
totalMemoryAllocatedGiB	integer	Total memory available for composition by this memory appliance

Appliance

Description:

A single memory appliance.

URL:

/cfm/v1/appliances/{applianceId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single memory appliance. {applianceId} has to match the id of the appliance
DELETE	Operator	Remove a memory appliance by id from CFM service management

Supported GET method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
blades	object	Array of memory blades managed by this memory appliance
totalMemoryAvailableGiB	integer	Total memory managed by this memory appliance

totalMemoryAllocatedGiB	integer	Total memory available for composition by this memory appliance
--------------------------------	---------	---

Supported DELETE method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
blades	object	Array of memory blades managed by this memory appliance
totalMemoryAvailableGiB	integer	Total memory managed by this memory appliance
totalMemoryAllocatedGiB	integer	Total memory available for composition by this memory appliance

BladesCollection

Description:

A collection of the memory blades within a memory appliance.

URL:

/cfm/v1/appliances/{applianceId}/blades

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of all managed blades under a memory appliance. Collection may be empty
POST	Operator	Add a new blade under a memory appliance and credentials to be managed by CFM service

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of uri to each of the member in this collection

Supported POST method parameter:

Parameter	Type	Required	Description
username	string	True	The username credentials used to communicate with the service to be added
password	string	True	The password credentials used to communicate with another service

ipAddress	string	True	The IP Address in dot notation of the service with optional port number appended by colon
------------------	--------	------	---

Supported POST method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
IP Address	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
ports	object	CXL upstream ports managed by this memory blade
resources	object	Memory resource blocks managed by this memory blade
memory	object	Existing memory regions composed by this memory blade
totalMemoryAvailableGiB	integer	Total memory managed by this memory blade
totalMemoryAllocatedGiB	integer	Total memory available for composition by this memory blade

Blade

Description:

A single memory blade.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single memory blade. {bladeId} has to match the id of the blade
DELETE	Operator	Remove a memory blade by id from CFM service management

Supported GET method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
ports	object	CXL upstream ports managed by this memory blade
resources	object	Memory resource blocks managed by this memory blade

memory	object	Existing memory regions composed by this memory blade
totalMemoryAvailableGiB	integer	Total memory managed by this memory blade
totalMemoryAllocatedGiB	integer	Total memory available for composition by this memory blade

Supported DELETE method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
ports	object	CXL upstream ports managed by this memory blade
resources	object	Memory resource blocks managed by this memory blade
memory	object	Existing memory regions composed by this memory blade
totalMemoryAvailableGiB	integer	Total memory managed by this memory blade
totalMemoryAllocatedGiB	integer	Total memory available for composition by this memory blade

resourcesCollection

Description:

A collection of the memory blocks as resources within a memory blade. Memory block is the minimum element to compose a memory region, and memory blocks are physically continuous ordered by the block id.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}/resources

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of memory block resources

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of uri to each of the member in this collection

Resource

Description:

A single memory resource block.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}/resources/{resourceId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single memory resource block

Supported GET method property:

Property	Type	Description
id	string	The id of this resource
compositionStatus	object	The composition status of this memory resource block
capacityGiB	integer	The capacity of this memory resource block in gibiBytes

portsCollection

Description:

A collection of the memory blade ports. Each port matches a physical connection port on the memory blade. It is also equivalent to a CXL upstream port managed by the blade's internal CXL fabric switch.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}/ports

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of memory blade ports

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of uri to each of the member in this collection

Port

Description:

A single memory blade port.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}/ports/{portId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single port

Supported GET method property:

Property	Type	Description
id	string	The id of this resource
gcxlid	string	The global CXL identifier
linkedPort	string	The uri of the port on the other side of the cable
currentSpeedGbps	integer	The current speed of this port
statusHealth	string	The health of the resource
statusState	string	The state of the resource
currentLinkWidth	integer	The number of lanes, phys, or other physical transport links that this port contains
linkStatus	string	Status of the link, such as LinkUp or LinkDown
linkState	string	State of the link, such as Enabled or Disabled

memoryCollection

Description:

A collection of the memory regions.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}/memory

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of memory regions. The collection may be empty
POST	Operator	Allocate a new memory region from a Memory Blade and optionally assign it to a port

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of URL to each of the member in this collection

Supported POST method parameter:

Parameter	Type	Required	Description
Port	string	False	The CXL port name on the Memory Blade
memorySizeGiB	integer	True	A gibibyte equals 2**30 or 1,073,741,824 bytes
QoS	integer	True	

Supported POST method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region
memoryBladeId	string	Memory blade id associated with this memory region
memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters

Memory

Description:

A single memory region.

URL:

/cfm/v1/appliances/{applianceId}/blades/{bladeId}/memory/{memoryId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single memory region
PATCH	Operator	Update a memory region's linked port
DELETE	Operator	Free a memory region by id. This also Unassign the memory region from connected blade ports

Supported GET method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region

memoryBladeId	string	Memory blade id associated with this memory region
memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters

Supported PATCH method parameter:

Parameter	Type	Required	Description
Port	string	True	The CXL port name on the Memory Blade
Operation	string	True	-assign- an existing memory to a port or -unassign- an existing link between memory and port

Supported PATCH method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region

memoryBladeId	string	Memory blade id associated with this memory region
memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters

Supported DELETE method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region
memoryBladeId	string	Memory blade id associated with this memory region
memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters

hostsCollection

Description:

A collection of the cxi host servers under management.

URI:

/cfm/v1/hosts

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieves a collection of the cxi host servers managed by CFM service
POST	Operator	Add a new CXL host and credentials to be managed by the CFM service

Supported GET method property:

Property	Type	Description
memberCount	Integer	Number of total members in this collection
members	Array	Array of uri to each of the member in this collection

Supported POST method parameter:

Parameter	Type	Required	Description
username	string	True	The username credentials used to communicate with the service to be added
password	string	True	The password credentials used to communicate with another service
ipAddress	string	True	The IP Address in dot notation of the service with optional port number appended by colon

Supported POST method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
ports	object	CXL root ports on this CXL host server
memory-devices	object	CXL logical devices on this CXL host server
memory	object	Total memory regions on this CXL host server
localMemoryGiB	integer	Total local memory (from DIMM slots) in GiB on this CXL host server
remoteMemoryGiB	integer	Total remote memory (from CXL memory appliance) in GiB on this CXL host server

Host

Description:

A single CXL host server.

URL:

/cfm/v1/hosts/{hostId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Get the information for a single CXL Host
DELETE	Operator	Remove a CXL Host by id from CFM service management

Supported GET method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
ports	object	CXL root ports on this CXL host server
memory-devices	object	CXL logical devices on this CXL host server
memory	object	Total memory regions on this CXL host server
localMemoryGiB	integer	Total local memory (from DIMM slots) in GiB on this CXL host server
remoteMemoryGiB	integer	Total remote memory (from CXL memory appliance) in GiB on this CXL host server

Supported DELETE method property:

Property	Type	Description
id	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters
ipAddress	string	The IP Address in dot notation of the service with optional port number appended by colon
status	string	A response string
ports	object	CXL root ports on this CXL host server
memory-devices	object	CXL logical devices on this CXL host server
memory	object	Total memory regions on this CXL host server
localMemoryGiB	integer	Total local memory (from DIMM slots) in GiB on this CXL host server
remoteMemoryGiB	integer	Total remote memory (from CXL memory appliance) in GiB on this CXL host server

portsCollection

Description:

A collection of the cxl host ports. Host ports are virtual ports backed by CXL memory devices.

URL:

/cfm/v1/hosts/{hostId}/ports

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of CXL host ports

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of uri to each of the member in this collection

Port

Description:

A single CXL host port.

URL:

/cfm/v1/hosts/{hostId}/ports/{portId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve the information for a single CXL host port

Supported GET method property:

Property	Type	Description
id	string	The id of this resource
gcxlid	string	The global CXL identifier
linkedPort	string	The uri of the port on the other side of the cable
currentSpeedGbps	integer	The current speed of this port
statusHealth	string	The health of the resource
statusState	string	The state of the resource
currentLinkWidth	integer	The number of lanes, phys, or other physical transport links that this port contains
linkStatus	string	Status of the link, such as LinkUp or LinkDown
linkState	string	State of the link, such as Enabled or Disabled

memory-devicesCollection

Description

A collection of the memory devices (CXL logical devices) on the CXL host server.

URL:

/cfm/v1/hosts/{hostId}/memory-devices

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of CXL host memory devices (CXL logical devices)

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of uri to each of the member in this collection

MemoryDevice

Description:

A single memory device (CXL logical device).

URL:

/cfm/v1/hosts/{hostId}/memory-devices/{memoryDeviceId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single memory device (CXL logical device)

Supported GET method property:

Property	Type	Description
id	string	The id of this resource
deviceType	string	The type of the device
memorySizeGiB	integer	The memory size of the device in GibiBytes
linkStatus	object	The speed and width about the link
statusState	string	The state of the resource

MemoryCollection

Description:

A collection of the memory regions on the CXL host servers.

URL:

/cfm/v1/hosts/{hostId}/memory

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve a collection of memory regions on CXL host
POST	Operator	Request a memory region to be composed from a corresponding memory blade and assigned to this CXL host's virtual port

Supported GET method property:

Property	Type	Description
memberCount	integer	Number of total members in this collection
members	array	Array of url to each of the member in this collection

Supported POST method parameter:

Parameter	Type	Required	Description
Port	string	False	The CXL port name on the Memory Blade
memorySizeGiB	integer	True	A gibibyte equals 2^{30} or 1,073,741,824 bytes
QoS	integer	True	

Supported POST method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region
memoryBladeId	string	Memory blade id associated with this memory region
memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters

Memory

Description:

A single memory region on the CXL host servers.

URL:

/cfm/v1/hosts/{hostId}/memory/{memoryId}

Supported HTTP Method:

HTTP Method	Required Role	Description
GET	ReadOnly	Retrieve information for a single memory region
DELETE	Operator	Free a memory region by id from its corresponding memory blade

Supported GET method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region
memoryBladeId	string	Memory blade id associated with this memory region

memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters

Supported DELETE method property:

Property	Type	Description
id	string	The identifier of this memory region
status	string	A response string
type	string	The memory type of this memory region
sizeGiB	integer	Memory capacity in GiB
bandwidth	integer	Either measured or self-claimed bandwidth of the memory region
latency	integer	Either measured or self-claimed latency of the memory region
memoryApplianceId	string	Memory appliance id associated with this memory region
memoryBladeId	string	Memory blade id associated with this memory region
memoryBladePort	string	Blade CXL upstream port assigned to this memory region
mappedHostId	string	CXL host id associated with this memory region
mappedHostPort	string	The id uniquely identifies the resource within a resource collection. Must not contain RFC1738 unsafe characters