Platform Management Communications Infrastructure (PMCI) Architecture White Paper

Version: 2.3.0

Document Identifier: DSP2015

Date: 2025-06-06

Version History: https://www.dmtf.org/dsp/DSP2015

Supersedes: 2.2.0

Document Class: Informational

Document Status: Published

Document Language: en-US

14 This document's normative language is English. Translation into other languages is permitted.

CONTENTS

15 1 Abstract

16 The Platform Management Communications Infrastructure (PMCI) Working Group defines standards to address "inside the box" communication interfaces between the components of the platform management subsystem.

17 The group develops the Network Controller Sideband Interface (NC-SI), Management Component Transport Protocol (MCTP), and Platform Level Data Model (PLDM) specifications that provide a comprehensive, common architecture for improved communication between management subsystem components. These specifications enable the monitoring and control of systems independent of the OS state, when the OS is running or an OS is not available (for example, when a system is booting, before the OS has loaded, or when the OS is inoperable).

18 The PMCI Working Group creates intra-platform manageability standards and technologies, which complement DMTF inter-platform standards such as the Security Protocol and Data Model (SPDM) from the SPDM Working Group, Redfish API from the Redfish Forum, Common Information Model (CIM) profiles, as well as remote access protocols that are defined in the other DMTF groups.

2 Foreword

The *Platform Management Communications Infrastructure (PMCI) Architecture White Paper* (DSP2015) was prepared by the Platform Management Communications Infrastructure Working Group.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. For information about DMTF, see https://www.dmtf.org.

2.1 Acknowledgments

DMTF acknowledges the following individuals for their contributions to this document:

Editor:

- Patrick Caporale — Lenovo

Contributors:

- Richelle Ahlvers — Broadcom Inc.
- Samer El-Haj-Mahmoud — ARM Inc.
- Jeff Hilland — Hewlett Packard Enterprise
- Yuval Itkin — NVIDIA Corporation
- Ira Kalman — Intel Corporation
- Eliel Louzoun — Intel Corporation
- Rob Mapes — Marvell International Ltd
- Balaji Natrajan — Microchip Technology Inc.
- Edward Newman — Hewlett Packard Enterprise
- Scott Phuong — Cisco Systems, Inc.
- Jeffrey Plank — Microchip Technology Inc.
- Bill Scherer — Hewlett Packard Enterprise
- Patrick Schoeller — Hewlett Packard Enterprise, Intel Corporation
- Brett Scrivner — Lenovo
- Hemal Shah — Broadcom Inc.
- Bob Stevens — Dell Technologies
- Supreeth Venkatesh — ARM Inc.

3 Introduction

The Platform Management Communications Infrastructure (PMCI) Working Group defines standards to address “inside the box” communication interfaces between the components of the platform management subsystem.

This document lays forth the basic architectural concepts that are driving the specifications being defined by the PMCI Working Group (Note: This architecture is referred to as PMCI architecture or PMCI herein). The focus of PMCI architecture is to enable intercommunications between different management components of a platform management subsystem in a standard manner across any implementation of a management component, independent of the operating system state.

3.1 Document conventions

The following typographical conventions are used in this document:

- Document titles are marked in *italics*.

31 4 Scope

32 This white paper provides an overview of the PMCI working group and its goals, the PMCI architecture, and a high-level summary of the primary specifications which it creates.

33 The intended target audience for this document is the readers interested in understanding management components intercommunications between the components of platform management subsystems. A platform management subsystem may be contained within servers, desktop systems, mobile systems, thin clients, bladed systems, and other types of devices.

34 This white paper is not a replacement for the individual PMCI specifications, but will provide an overview on how the specifications relate to each other within the PMCI stack model.

35

5 References

36 The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

37 DMTF DSP0218, *Platform Level Data Model (PLDM) for Redfish Device Enablement* 1.2 https://www.dmtf.org/standards/published_documents/DSP0218_1.2.pdf

38 DMTF DSP0220, *Field Replaceable Unit (FRU) Data Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0220_1.0.pdf

39 DMTF DSP0222, *Network Controller Sideband Interface (NC-SI) Specification* 1.2 https://www.dmtf.org/standards/published_documents/DSP0222_1.2.pdf

40 DMTF DSP0233, *Management Component Transport Protocol (MCTP) I3C Transport Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0233_1.0.pdf

41 DMTF DSP0234, *CXL Fabric Manager API over MCTP Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0234_1.0.pdf

42 DMTF DSP0235, *NVMe (NVM Express) Management Messages over MCTP Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0235_1.0.pdf

43 DMTF DSP0236, *Management Component Transport Protocol (MCTP) Base Specification* 1.3 https://www.dmtf.org/standards/published_documents/DSP0236_1.3.pdf

44 DMTF DSP0237, *Management Component Transport Protocol (MCTP) SMBus/I2C Transport Binding Specification* 1.2 https://www.dmtf.org/standards/published_documents/DSP0237_1.2.pdf

45 DMTF DSP0238, *Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification* 1.3 https://www.dmtf.org/standards/published_documents/DSP0238_1.3.pdf

46 DMTF DSP0239, *Management Component Transport Protocol (MCTP) IDs and Codes* 1.11 https://www.dmtf.org/standards/published_documents/DSP0239_1.11.pdf

47 DMTF DSP0240, *Platform Level Data Model (PLDM) Base Specification* 1.2 https://www.dmtf.org/standards/published_documents/DSP0240_1.2.pdf

48 DMTF DSP0241, *Platform Level Data Model (PLDM) over MCTP Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0241_1.0.pdf

49 DMTF DSP0242, *Platform Level Data Model (PLDM) for File Transfer Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0242_1.0.pdf

50 DMTF DSP0245, *Platform Level Data Model (PLDM) IDs and Codes Specification* 1.4 https://www.dmtf.org/standards/published_documents/DSP0245_1.4.pdf

51 DMTF DSP0246, *Platform Level Data Model (PLDM) for SMBIOS Data Transfer Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0246_1.0.pdf

52 DMTF DSP0247, *Platform Level Data Model (PLDM) for BIOS Control and Configuration Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0247_1.0.pdf

53 DMTF DSP0248, *Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification* 1.3 https://www.dmtf.org/standards/published_documents/DSP0248_1.3.pdf

54 DMTF DSP0249, *Platform Level Data Model (PLDM) State Set Specification* 1.3 https://www.dmtf.org/standards/published_documents/DSP0249_1.3.pdf

55 DMTF DSP0253, *Management Component Transport Protocol (MCTP) Serial Transport Binding* 1.0 https://www.dmtf.org/standards/published_documents/DSP0253_1.0.pdf

56 DMTF DSP0254, *Management Component Transport Protocol (MCTP) KCS Transport Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0254_1.0.pdf

57 DMTF DSP0256, *Management Component Transport Protocol (MCTP) Host Interface Specification* 2.0 https://www.dmtf.org/standards/published_documents/DSP0256_2.0.pdf

58 DMTF DSP0257, *Platform Level Data Model (PLDM) for FRU Data Specification* 2.0 https://www.dmtf.org/standards/published_documents/DSP0257_2.0.pdf

59 DMTF DSP0261, *NC-SI over MCTP Binding Specification* 1.3 https://www.dmtf.org/standards/published_documents/DSP0261_1.3.pdf

60 DMTF DSP0267, *Platform Level Data Model (PLDM) for Firmware Update Specification* 1.3 https://www.dmtf.org/standards/published_documents/DSP0267_1.3.pdf

61 DMTF DSP0275, *Security Protocol and Data Model (SPDM) over MCTP Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0275_1.0.pdf

62 DMTF DSP0276, *Secured Messages using SPDM over MCTP Binding Specification* 1.2 https://www.dmtf.org/standards/published_documents/DSP0276_1.2.pdf

63 DMTF DSP0280, *PMCI Test Tools Interface and Design Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0280_1.0.pdf

64 DMTF DSP0281, *CXL Type 3 Device Component Command Interface over MCTP Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0281_1.0.pdf

65 DMTF DSP0282, *Memory-Mapped Buffer Interface (MMBI) Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0282_1.0.pdf

66 DMTF DSP0283, *Management Component Transport Protocol (MCTP) Universal Serial Bus (USB) Transport Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0283_1.0.pdf

67 DMTF DSP0284, *Management Component Transport Protocol (MCTP) Memory-Mapped Buffer Interface (MMBI) Transport Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0284_1.0.pdf

68 DMTF DSP0291, *PCIe Management Interface (PCIe-MI) over MCTP Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0291_1.0.pdf

69 DMTF DSP0292, *Management Component Transport Protocol (MCTP) Platform Communications Channel (PCC) Transport Binding Specification* 1.0 https://www.dmtf.org/standards/published_documents/DSP0292_1.0.pdf

70 DMTF DSP2016, *Management Component Transport Protocol (MCTP) Overview White Paper* 1.0 https://www.dmtf.org/standards/published_documents/DSP2016_1.0.pdf

71 DMTF DSP2037, *MCTP Packets and NC-SI over MCTP Overview* 1.0 https://www.dmtf.org/standards/published_documents/DSP2037_1.0.pdf

72 DMTF DSP2054, *PLDM NIC Modeling* 1.0 https://www.dmtf.org/standards/published_documents/DSP2054_1.0.pdf

73 DMTF DSP2061, *PLDM Accelerator Modeling* 1.0 https://www.dmtf.org/standards/published_documents/DSP2061_1.0.pdf

74 DMTF DSP2067, *PLDM CXL Memory Modeling* 1.0 https://www.dmtf.org/standards/published_documents/DSP2067_1.0.pdf

75 DMTF DSP2070, *PLDM Host SoC Modeling* 1.0 https://www.dmtf.org/standards/published_documents/DSP2070_1.0.pdf

6 Terms and definitions

For the purposes of this document, the following terms and definitions apply:

Managed Device

A device that is typically implemented using a microcontroller and accessed through a messaging protocol and is used for accessing one or more management parameters. Management parameter access provided by a managed device is typically accomplished using an abstracted interface and data model rather than through direct "register level" accesses. A managed device responds to management requests, but does not initiate or aggregate management operations except in conjunction with a management controller (that is, it is a satellite device that is subsidiary to one or more management controllers).

Management Controller

An intelligent entity composed of hardware/firmware/software that resides within a platform and is responsible for some or all of the management functions associated with the platform; also known as BMC and Service Processor.

Management Parameter

A particular datum representing a characteristic, capability, status, or control point associated with a managed entity. Example management parameters include temperature, speed, voltage, on/off, link state, uncorrectable error count, device power state, and so on.

Network Controller

A managed device within a system that is responsible for providing connectivity to an external network world.

Network Controller Sideband Interface

The interface of the Network Controller that provides network pass-through and/or a control path to a Management Controller; also shown as Sideband Interface or NC-SI as appropriate in the context.

Platform Management Communications Infrastructure

The Platform Management Communications Infrastructure (PMCI) Working Group defines standards to address “inside the box” communication interfaces between the components of the platform management subsystem.

7 Symbols and abbreviated terms

The following abbreviations are used in this document.

AMC

Auxiliary Management Controller

API

Application Programming Interface

BIOS

Basic Input/Output System

BMC

Baseboard Management Controller

CCI

Component Command Interface

CIM

Common Information Model

CXL®

Compute Express Link

FRU

Field Replaceable Unit

I^2C

Inter-Integrated Circuit

I^3C

Improved Inter-Integrated Circuit

IID

Instance ID

KCS

Keyboard Controller Style

MC

Management Controller

MCTP

Management Component Transport Protocol

MD

Managed Device

MMBI

Memory-Mapped Buffer Interface

NC

Network Controller

NC-SI

Network Controller Sideband Interface

NVMe-MI™

NVM Express® Management Interface

OS

Operating System

PC

Personal Computer

PCIe®

Peripheral Component Interconnect Express

PLDM

Platform Level Data Model

PMCI

Platform Management Communications Infrastructure

RBT

RMII-Based Transport

RDE

Redfish Device Enablement

RMII

Reduced Media Independent Interface

SMBus™

System Management Bus

SoC

System on a Chip

SPDM

Security Protocol and Data Model

UEFI

Unified Extensible Firmware Interface

USB

Universal Serial Bus

VDM

Vendor Defined Message

158 8 Platform Management Subsystem Architecture Overview

159 A Platform Management Subsystem in a modern enterprise computing platforms is composed of a set of components which communicate to perform management functions within the platform. In many cases, these communications and interfaces are specialized and adapted to each individual platform, installation, and component in the environment.

160 A platform management subsystem provides hardware management services such as:

- Platform environmental monitoring functions (for example, temperature probing, voltage monitoring, fan speeds, hardware error status, etc.)
- Control functions (for example, platform power-on/off, reset, watchdog timer, etc.)
- Device firmware update and device functional management

161 The platform management subsystem frequently includes one or more intelligent controllers (microcontrollers) that support access to the management monitoring and control functions, which provide monitoring and control services for access by other management controllers in the subsystem. The platform management subsystem can be represented externally via the management controller through outward bound standards provided by other working groups or forums within DMTF. One example is the Redfish API that can be implemented as a service provider contained within the management controller which will enable a full end to end management approach. The use of the Redfish API standard for external connectivity, and a combination of MCTP, PLDM, NC-SI, and MMBI standards for internal communication provides for complete DMTF standards management of a Platform Management Subsystem.

162 PMCI supports a suite of specifications (MCTP, PLDM, NC-SI, MMBI, and FRU) which include architectural semantics, industry standard protocols, and platform level data models to standardize the management related intercommunications between the components of platform management subsystem independent of component implementation, platform state, and platform management subsystem implementation.

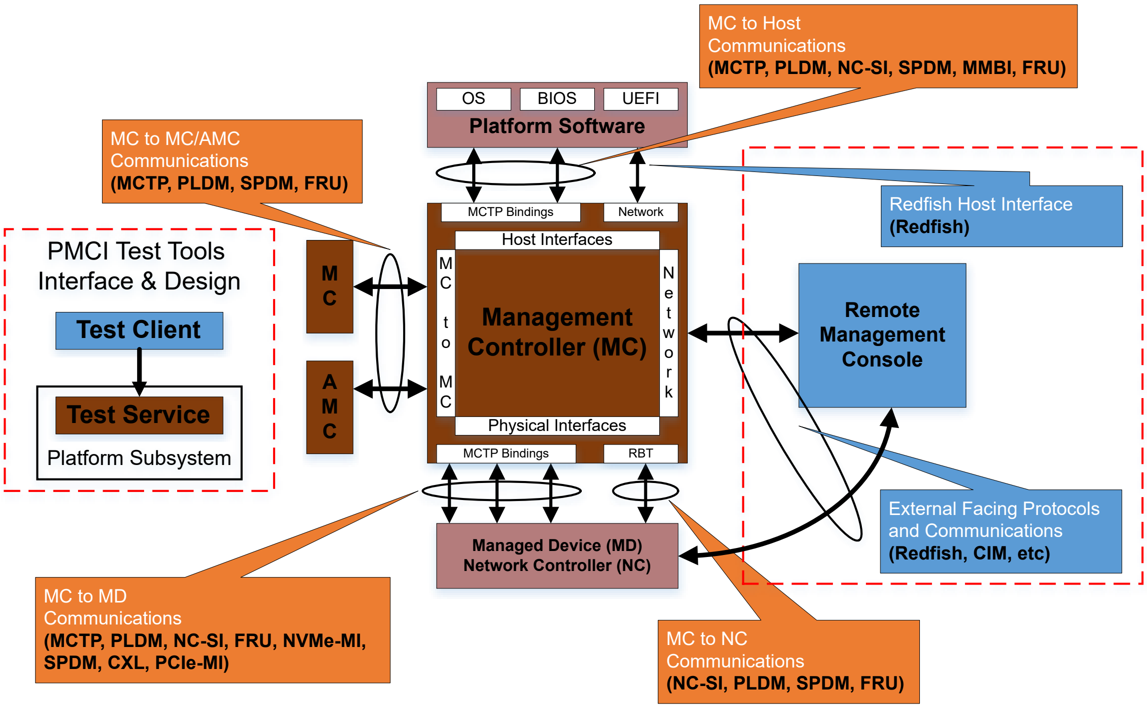
163 8.1 Principal Goals

164 One goal of PMCI is to enable intercommunications between different types of platform components using a set of standards protocols, interfaces, and platform level data models. An example of the platform management subsystem is provided in Figure 1 to illustrate different types of components and intercommunications within a platform.

165 Another goal of PMCI is to enable the same semantics, protocols, and interfaces to work across a full range of platforms — traditional servers, desktop systems, mobile, laptop, bladed PCs as well as “thin clients”.

166 8.2 Platform Management Subsystem Components

167



168 **Figure 1 — Platform Management Subsystem**

169 Figure 1 shows the different components within a platform management subsystem. The components can be divided into the following four categories:

1. Management Controller (MC): A microcontroller or processor that aggregates Management Parameters from one or more Managed Devices and Network Controllers and makes access to those parameters to local or remote software, or to other Management Controllers, via one or more management data models.
2. Platform Software: The software running on the host CPUs that communicates with a management controller for performing a set of management functions. The examples of the platform software may include BIOS, OS, UEFI firmware, etc.
3. Managed Device (MD) or Network Controller (NC): A Managed Device responds to management requests from the Management Controller, and can also initiate asynchronous messages, such as events, if enabled to by a Management Controller. A Network Controller is a managed device that additionally supports the NC-SI standard. A Network Controller may also provide connectivity to an external network.
4. Remote Management Console: is a function that enables communications with the management

controller through one or more DMTF standards (for example the Redfish API or CIM). The remote console may initiate management queries or actions by sending requests to the MC which can use PMCI standards to communicate to Managed Devices or Network Controllers. The remote management console can also be located within the Platform Software and use MCTP Host Interface to communicate with the MC. Other DMTF standards such as Redfish Host Interface could also be used in the connectivity between the host and the MC.

170 PMCI covers all four types of intercommunications between the above components.

1. Management Controller and Host (platform software)
2. Management Controller and Managed Devices
3. Management Controller and Network Controller
4. Management Controller and another Management Controller or similar device

171 Other DMTF standards such as the Redfish API or CIM provide the external facing intercommunications between a management controller and a remote console or client.

9 PMCI Overview

9.1 Standards

The PMCI Working Group produces standards for six primary intercommunication interfaces/data models.

1) A family of specifications for a transport protocol known as Management Component Transport Protocol (MCTP). This protocol can be used to send messages between components of the platform management subsystem. Additional binding specifications are available for MCTP that permit the transport to operate over different physical mediums, which can support MCTP messages.

2) A family of specifications known as Platform Level Data Model (PLDM). These specifications define how individual management functions such as inventory, monitoring & control, eventing, firmware update, and Redfish device enablement (RDE) are abstracted and accessed by an MC.

3) The Network Controller Sideband Interface (NC-SI) specification defines how an MC can communicate to an NC for management functions such as inventory, external Ethernet pass-through to the MC, events, configuration, and statistics collection.

4) A Memory-Mapped Buffer Interface (MMBI) specification defines a shared-memory concept to permit packet exchanges between host software and an MC.

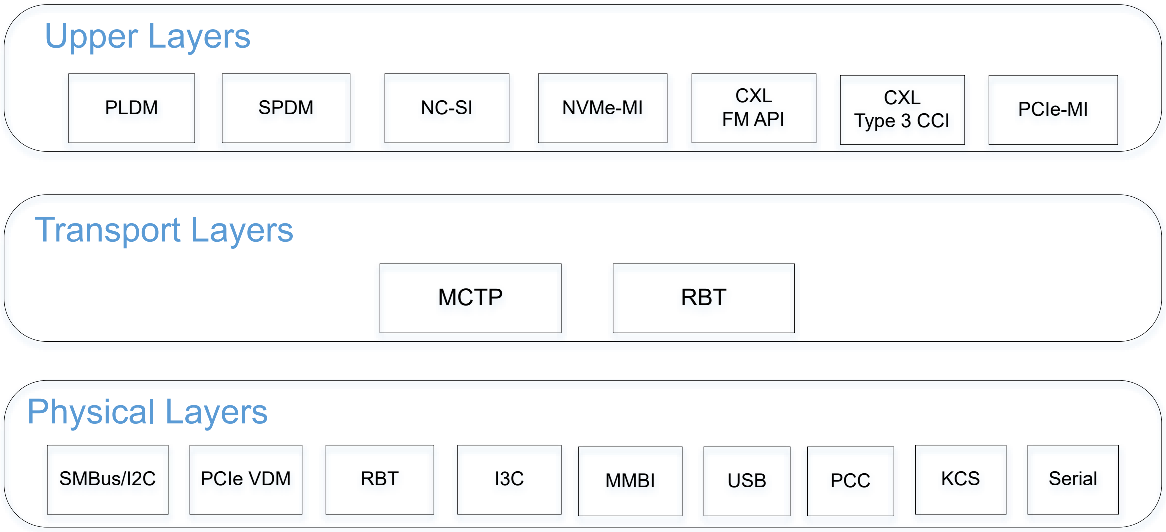
5) A Field Replaceable Unit (FRU) Data specification defines a data content structure for vital product data that describes a platform asset or component.

6) A PMCI Test Tools Interface and Design specification defines messages, data objects, and sequences for testing PMCI protocol implementations in devices over a variety of transport media.

9.2 PMCI Stack

Platform Management Communications Infrastructure (PMCI) Working Group is defining a set of standards that can be used for communications between the platform components. A simplified view of the PMCI stack is shown below as it organizes the standards into three primary groupings (upper layers, transport layer, and physical layer). This figure does not show the relationship or binding between each layer.

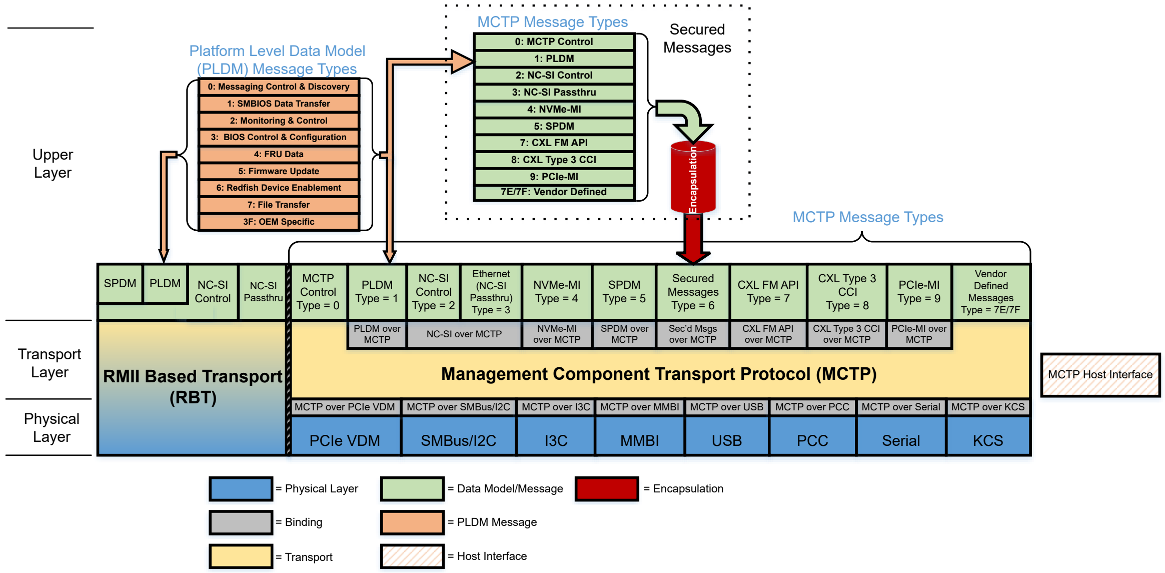
183



184 **Figure 2 — Simplified view of the PMCI Stack**

185 The following figure shows the full view of a PMCI stack, which includes the binding details.

186



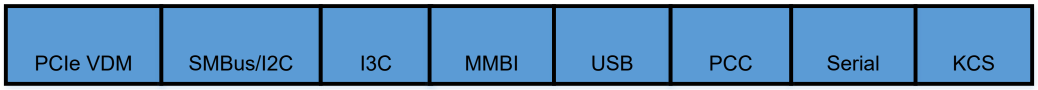
187 **Figure 3 — Full PMCI Stack**

188 In order to understand the full PMCI stack, each layer of the stack will be described in further detail in the next sections.

189 9.2.1 Physical Medium Layer

190 All of the PMCI standards and protocols are architected to be implemented on a physical medium. The diagram below represents the lowest portion of the PMCI stack and shows all the physical mediums that are currently supported. PMCI continues to expand the list of supported physical mediums, and additional binding specifications may be available in the future.

191



192 **Figure 4 — Physical Medium Layer**

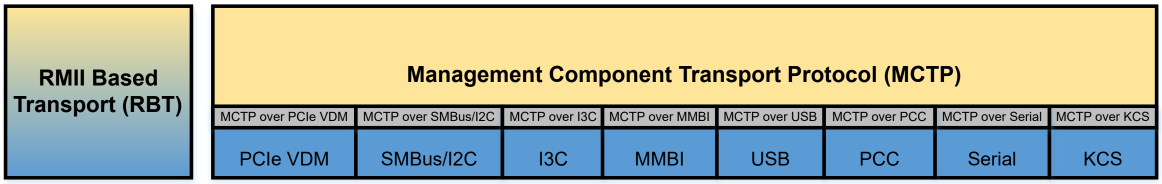
193 The RMII-Based Transport (RBT) physical medium is the foundation for the NC-SI specification and is derived from the RMII specification. The electrical and timing requirements for an RBT interface is fully described within the NC-SI specification, and as its name implies also includes the transport details for sending and receiving messages. The RBT interface therefore is special within the PMCI stack as it is both a physical layer medium, and a transport layer combined.

194 The remainder of physical mediums shown in the figure above, represent available interconnects that the MCTP specification can be used with.

195 9.2.2 Transport Layer

196 There are two transports available from the PMCI Working Group, RBT and MCTP. Each of these transports define a message passing protocol though there are differences between these two PMCI transports.

197



198 **Figure 5 — Transport Layer**

199 The RBT transport is a simple protocol used to track pass-through traffic and the reliable reception of command packets. Pass-through traffic consists of packets that are transferred between the external network interface and the Management Controller using the RBT interface. The transport protocol is based upon a command/response paradigm and involves the use of unique Instance IDs (IIDs) in the packet headers to allow responses received to be matched to previously transmitted commands. The Management Controller is the generator of command packets sent to the Sideband Interface of one or more Network Controllers in the system, and it receives response packets from them. Most but not all request messages sent over the RBT transport have a corresponding response message.

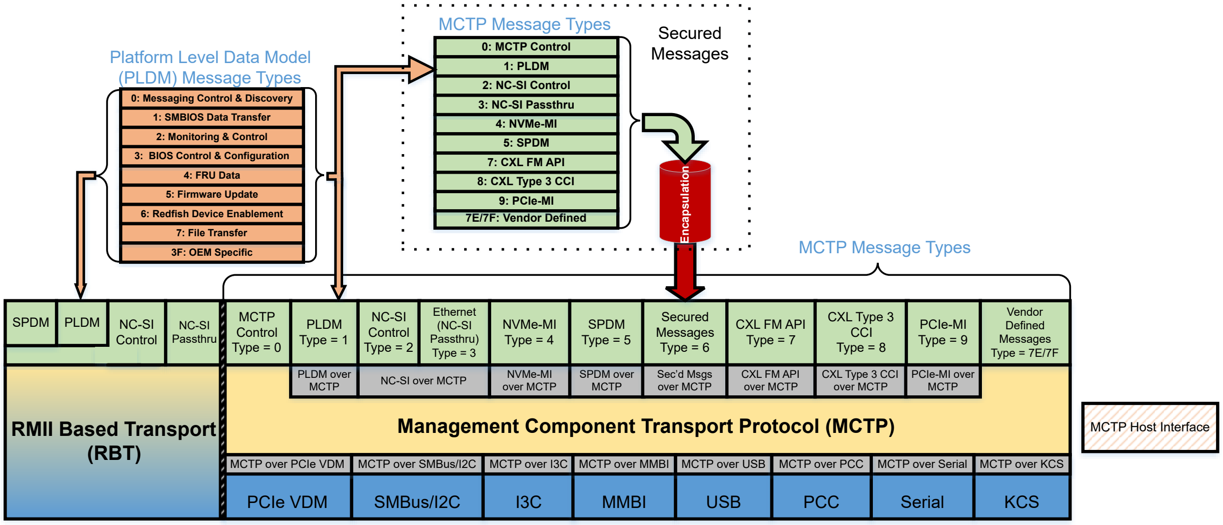
An asynchronous event notification is one example of a packet sent by the Network Controller without a corresponding response message.

200 The MCTP transport can support both acknowledged (typically request/response) and unacknowledged messages (asynchronous). MCTP specifications include a grouping of documents known as binding specifications, which define the necessary header and timing requirements for the transport to be used on the applicable physical mediums. Separate specifications are available for bindings to different physical media, such as MCTP over PCIe VDM Binding and MCTP over SMBus/I2C Binding. MCTP can also uniquely attach to interfaces used to communicate to/from a host system and its software (OS, UEFI, BIOS, etc.).

201 9.2.3 Upper (Data Model) Layer

202 Sitting on top of the two PMCI transports are multiple choices for message definition and data models. MCTP provides a base control set of messages - and through additional binding specifications; PLDM, NC-SI, NVMe-MI, SPDM, CXL FM API, CXL Type 3 CCI, and PCIe-MI® messages.

203



204 **Figure 6 — Data Model Layer**

205 The layers above MCTP define different communication and data models mapped over MCTP. The MCTP Control Protocol is used to set up and initialize managed devices within an MCTP network.

206 Platform Level Data Model (PLDM) provides efficient access to low-level platform monitoring, control, and data transfer functions such as temperature, fan, voltage, inventory data, event data transfer, and boot control. PLDM over MCTP defines data representations and commands that abstract the platform management hardware. More recent PLDM specifications have defined methods to perform a firmware update and support Redfish enablement on managed devices.

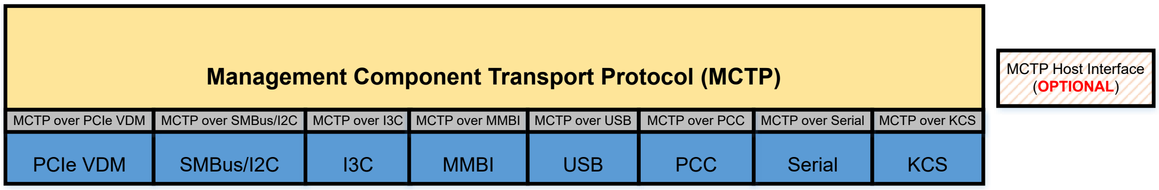
207 NC-SI defines a pass-through model of Ethernet communications between a management controller and a network controller.

208 SPDM defines a set of commands for authentication, firmware measurements, and certificate management.

209 9.2.4 Host Interface

210 MCTP provides a method for the host to communicate to the management controller through a physical layer host accessible interface. Initially, DSP0256 version 1.0 only supported Serial and KCS binding specifications; however with the updated 2.0 version of the Host Interface Specification, the entire set of physical interfaces could now be used to permit host to management controller communications. This includes the original PCIe VDM, I2C/SMBus bindings as well as the newer USB, MMBI, and PCC Transport bindings. This provides a wide choice of options to a given management controller where it can optionally support one or more of the binding methods for host-based traffic.

211



212 **Figure 7 — MCTP Host Interface**

213 10 PMCI Standards Overview

214 The PMCI standards are composed of technologies defined in a suite of standard specifications. These standards include the Management Component Transport Protocol (MCTP) related specifications, the Platform Level Data Model (PLDM) related specifications, the Network Controller Sideband Interface (NC-SI) specification, the Memory-Mapped Buffer Interface (MMBI) specification, the Field Replaceable Unit (FRU) Data specification, and the PMCI Test Tools Interface and Design specification.

215 10.1 Management Component Transport Protocol (MCTP)

216 The Management Component Transport Protocol (MCTP) is a protocol for intercommunications among intelligent devices within a platform management subsystem. This protocol is independent of the underlying physical bus properties, as well as the "data-link" layer messaging used on the bus.

217 The physical and data-link layer methods for MCTP communication across a given medium are defined by companion "transport binding" specifications, such as MCTP over PCIe® Vendor Defined Messaging and MCTP over SMBus/I2C. This approach enables future transport bindings to be defined to support additional buses without affecting the base MCTP specification.

218 The MCTP communication model includes a message format, transport description, message exchange patterns, and operational Endpoint characteristics. MCTP uses logical addressing based on Endpoint IDs that enables static/dynamic endpoint ID assignments as well as bridging/routing support. MCTP defines a simple message fragmentation/reassembly mechanism that allows large data transfers using MCTP packetization.

219 MCTP Control Protocol is used to setup/initialize MCTP control communications within an MCTP network. MCTP Control Protocol supports request/response, broadcast, and one-way communications.

220 The following specifications are available for MCTP:

- MCTP Base Specification — DSP0236
- MCTP PCIe VDM Transport Binding Specification — DSP0238
- MCTP SMBus/I2C Transport Binding Specification — DSP0237
- MCTP Serial Transport Binding Specification — DSP0253
- MCTP KCS Transport Binding Specification — DSP0254
- MCTP I3C Transport Binding Specification — DSP0233
- MCTP USB Transport Binding Specification — DSP0283
- MCTP PCC Transport Binding Specification — DSP0292
- MCTP Host Interface Specification — DSP0256
- MCTP ID & Codes — DSP0239
- NVMe Management Messages over MCTP Binding Specification — DSP0235
- SPDM over MCTP Binding Specification — DSP0275
- Secured Messages using SPDM over MCTP Binding Specification — DSP0276

- CXL Fabric Manager API over MCTP Binding Specification — DSP0234
- CXL Type 3 Device CCI over MCTP Binding Specification — DSP0281
- PCIe-MI over MCTP Binding Specification — DSP0291
- MCTP Overview White Paper — DSP2016

221 10.2 Platform Level Data Model (PLDM)

222 PLDM defines a method to provide efficient access to low-level inventory, monitoring, control, eventing, and data/ parameters transfer functions such as temperature, fan, voltage, event logging, and boot control. Recent PLDM extensions enable device firmware updates as well as device management consistent with the DMTF Redfish standard.

223 10.2.1 PLDM Messaging types and applications

224 PLDM has defined data representations and commands that abstract the platform management hardware. Extensions of the core PLDM specification work includes:

- Messages and data model for SMBIOS data transfer within the platform.
- Messages and data structures for Field Replaceable Unit (FRU), asset information, and firmware inventory data transfer.
- Messages and data structures for monitoring processors, caches, memory, sensors, fans, power state monitoring, time stamp clock monitoring, etc.
- Control messages/data structures for sensors, fans, power state management, boot control, real time stamp, and watchdog timer.
- Low level data models and messages to represent and transfer opaque data, BIOS data, and event data.
- Messages to transfer text console redirection and media redirection related messages.
- Data models and messages to facilitate device firmware management.
- Messages and data models that enable management controllers to effectively interact with targeted devices using an encapsulated Redfish based JSON format.
- Enablement of sending PLDM messages over the RBT transport which allows for managed devices with only the sideband RBT interface to communicate to an MC using PLDM.

225 The following specifications are available for PLDM:

- PLDM Base Specification — DSP0240
- PLDM over MCTP Binding Specification — DSP0241
- PLDM ID and Codes Specification — DSP0245
- PLDM State Set Specification — DSP0249
- PLDM for FRU Data Specification — DSP0257
- PLDM for SMBIOS Data Transfer Specification — DSP0246
- PLDM for BIOS Control and Configuration Specification — DSP0247
- PLDM for Platform Monitoring and Control Specification — DSP0248

- PLDM for Firmware Update Specification — DSP0267
- PLDM for Redfish Device Enablement (RDE) — DSP0218
- PLDM for File Transfer Specification — DSP0242
- PLDM NIC Modeling — DSP2054
- PLDM Accelerator Modeling — DSP2061
- PLDM CXL Memory Modeling — DSP2067
- PLDM Host SoC Modeling — DSP2070

226 10.3 Network Controller Sideband Interface (NC-SI)

227 The Network Controller Sideband Interface (NC-SI) specifies a sideband interface that uses RBT for both the transport and physical medium. In addition, NC-SI can be supported with MCTP as the transport. NC-SI defines the formats for communicating network traffic, control commands, responses, and asynchronous event notifications between a management controller and a network controller. NC-SI can support multiple Network Controllers through the use of hardware or command-based arbitration.

228 The following specifications are available for NC-SI:

- NC-SI Specification — DSP0222
- NC-SI over MCTP Binding Specification — DSP0261

229 10.4 Memory-Mapped Buffer Interface (MMBI)

230 The Memory-Mapped Buffer Interface specifies a shared-memory concept that allows packet exchanges between host software and an MC. The described memory mapping makes it possible for boot code (such as UEFI firmware) and OS-level software (such as the OS kernel or drivers) to establish efficient communication with an MC at bandwidth and latency limited by the underlying memory mapping mechanisms. The following specifications are available for MMBI:

- Memory-Mapped Buffer Interface (MMBI) Specification — DSP0282
- Management Component Transport Protocol (MCTP) Memory-Mapped Buffer Interface (MMBI) Transport Binding Specification — DSP0284

231 10.5 Field Replaceable Unit Data (FRU)

232 The Field Replaceable Unit (FRU) Data Specification defines a data content structure for vital product data that describes a platform asset or component. The FRU specification defines the data content structure for access in raw format. Access of FRU data via PLDM is defined in DSP0257. The following specifications are available for FRU:

- Field Replaceable Unit (FRU) Data Specification — DSP0220
- PLDM for FRU Data Specification — DSP0257

233 10.6 PMCI Test Tools Interface and Design

234 The PMCI Test Tools Interface and Design Specification defines messages, data objects, and sequences for testing PMCI protocol implementations in devices over a variety of transport media. The description of message exchanges includes secure registration of test clients, administrative messages for configuring test sessions, and specific messages for testing individual messages in the various PMCI protocols. The following specification is available for the tools interface:

- PMCI Test Tools Interface and Design Specification — DSP0280

235 11 Conclusion

236 PMCI supports a suite of specifications, which include architectural semantics, industry standard protocols, and platform level data models to standardize the management related intercommunications between the components of platform management subsystem independent of component implementation, platform state, and platform management subsystem implementation.

237 When used in conjunction with other DMTF standards for external facing communications, a complete end-to-end platform management subsystem can be developed for all management operations.

238 12 ANNEX A (informative) Change Log

Version	Date	Description
1.0.0	2007-07-23	Initial Release
2.0.0	2019-08-06	Updates to describe the latest architecture model available from PMCI.
2.1.0	2022-02-15	Updates to include the latest specifications & stack diagrams for PMCI architecture
2.2.0	2024-04-09	Updates to include the 2.0 MCTP Host Interface, PLDM File Transfer, and new MCTP bindings (MMBI, USB, PCIe-MI, and PCC)
2.3.0	2025-05-30	Updates to include the FRU spec, the PMCI Test Tools Interface and Design, and PLDM Host SoC Modeling.

13 Bibliography

DMTF DSP4014, *DMTF Process for Working Bodies*, https://www.dmtf.org/sites/default/files/standards/documents/DSP4014.pdf