Open. Together.
PMCI Standards for Hardware Management

Hemal Shah
Vice President of Technology and Senior VP, DMTF
Distinguished Engineer/Architect, Compute and Connectivity (CCX), Broadcom Inc.

Patrick Caporale
Vice President of Marketing, DMTF
Principal Engineer, Data Center Group (DCG), Lenovo
PMCI Standards and OCP Community

Workshop Goal:
Inform community on how OCP Projects can adopt existing and emerging PMCI standards

Benefits:
Provide interoperable management interfaces and data models for “inside the box” communication

Current Projects:
OCP NIC 3.0 – Design Specification lists required PMCI standards for key management features
DMTF Standards Applicable to OCP Platforms

The DMTF organization develops open manageability standards spanning diverse emerging and traditional IT infrastructures.

- **OCP Server**
- **OCP NIC**
- **OCP Hardware Management**

Redfish is a REST based external facing interface for remote management of a server platform.

- Redfish Client

**Network Controller Sideband Interface (NC-SI)**
- Management Component Transport Protocol (MCTP)
- Platform Level Data Model (PLDM)

Internal facing interfaces and protocols for platform management subsystem communications.

*OpenBMC (from The Linux Foundation) -- initial MCTP/PLDM implementation proposals submitted Dec 2018*
PMCI Working Group of the DMTF

Platform Management Component Intercommunications (PMCI)

Group formed in 2005, initial specifications released in 2007
Over a decade of implementations within server and desktop products

PMCI suite of standards provide “Inside the box” communication and functional interfaces between components within the platform management subsystem
Creates specifications for MCTP, PLDM, and NC-SI

Applicability to OCP

OCP NIC 3.0 Design Specification ver 0.85b leverages multiple PMCI standards including:
- DSP0236 - MCTP Base Specification
- DSP0222 - Network Controller Sideband Interface (NC-SI) Specification
- DSP0267 - Platform Level Data Model (PLDM) for Firmware Update Specification
- DSP0248 - Platform Level Data Model (PLDM) for Platform Monitoring and Control Specification

PMCI Standards can be complimentary to OCP Redfish Profiles adopted by Hardware Management Project
- DSP0218 -- PLDM for Redfish Device Enablement (WIP)
As well as DSP0248, DSP0267, DSP0236, DSP0222 and others
Platform Management Subsystem

MC ↔ MC Communications
(MCTP, PLDM)

MC ↔ Host Communications
(MCTP, Redfish Host Interface)

MC ↔ MD Communications
(MCTP, PLDM over MCTP,
NC-SI over MCTP,
NVMe-MI over MCTP)

MC ↔ NC Communications
(NC-SI, PLDM over RBT)

Management Controller (MC)

Remote Management Console

External Facing Protocols and Communications
(Redfish, CIM, etc)

Other Standards from DMTF

Platform Software

OS
BIOS
UEFI

Host Interface(s)

Physical Interfaces

PCIe
SMBus
RBT

Managed Device (MD)/
Network Controller (NC)

Serial
KCS
Network

PMCI Architecture

Open. Together.
## PMCI Protocol Stack

<table>
<thead>
<tr>
<th>Layer</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical Layer</td>
<td>NMCI, RMII, PCIe, I2C/SMBus</td>
</tr>
<tr>
<td>Binding</td>
<td>PLDM Base, PLDM Message</td>
</tr>
<tr>
<td>Transport</td>
<td>Management Component Transport Protocol (MCTP)</td>
</tr>
<tr>
<td>Data Model / Message</td>
<td>NC-SI Ctrl, NC-SI Passthru</td>
</tr>
<tr>
<td>Host Interface</td>
<td>MCTP Host Interface</td>
</tr>
</tbody>
</table>

### MCTP Protocols
- PCIe VDM
- SMBus/I2C
- Gen-Z
- Serial
- KCS
- I3C (Future)
- SMBus/I2C (Future)
- Gen-Z (Future)

### PLDM Protocols
- PLDM FRU Data
- PLDM Bios Control
- PLDM Mon & Ctrl
- PLDM SMBIOS
- PLDM BIOS
- PLDM FRU Update
- PLDM RDE

### Key Symbols
- **Blue** = Physical Layer
- **Green** = Data Model / Message
- **Brown** = PLDM Message
- **Orange** = Transport
- **Red** = Host Interface

---

**Open. Together.**
Management Component Transport Protocol (MCTP)

Base transport for “inside-the-box” communication.

Suitable for use with multiple media: SMBus, PCIe, etc.

Suitable for all computer platform types

Supports logical addressing based on Endpoint IDs

Provides simple message fragmentation/reassembly

Built-in capability discovery and supports path transmission unit discovery

Carries multiple message types: MCTP Control, PLDM, NC-SI, NVMe
MCTP Packet Fields

Medium-specific Header

MCTP transport header

MCTP packet payload

Medium-specific trailer

Physical Medium-Specific Header
Includes physical source and destination addresses.

<table>
<thead>
<tr>
<th>RSVD</th>
<th>Header version</th>
<th>Destination endpoint ID</th>
<th>Source endpoint ID</th>
<th>SOM</th>
<th>EOM</th>
<th>Pkt seq</th>
<th>T</th>
<th>O</th>
<th>Msg tag</th>
</tr>
</thead>
</table>

Message header

Message type

Message type-specific Header fields

Message body

Message integrity check

Physical Medium-Specific Trailer (for example, data integrity fields)
MCTP Transport Bindings

Bindings for four physical mediums defined

• SMBus
• PCIe VDM
• KCS
• Serial

Bindings for I3C and Gen-Z are work-in-progress
## MCTP over SMBus Packet Format

<table>
<thead>
<tr>
<th>Byte 1 &gt;</th>
<th>Byte 5 &gt;</th>
<th>Byte 9 &gt;</th>
<th>Byte N &gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Slave Address</td>
<td>MCTP Reserved</td>
<td>I C</td>
<td>PEC</td>
</tr>
<tr>
<td>Command Code = MCTP = 0Fh</td>
<td>Hdr Version</td>
<td>Msg Type</td>
<td></td>
</tr>
<tr>
<td>Byte Count</td>
<td>Destination Endpoint ID</td>
<td>Message Header</td>
<td>Message Data</td>
</tr>
<tr>
<td>Source Slave Address</td>
<td>Source Endpoint ID</td>
<td>Message Integrity Check</td>
<td></td>
</tr>
</tbody>
</table>

### MCTP Message Header

- **IC**: Message Integrity Check
- **PEC**: PEC
- **Message Type byte**: (only required in first packet header of message.)
- **Message Type** byte (Varies based on Message Type)

### Common Fields for all MCTP Messages

- Common fields for all MCTP messages

---

**OCP Summit**

**Open. Together.**
Out-Of-Band Management and NC-SI

A common interoperable sideband interface and protocol to transfer management traffic between a management controller (MC) & network controller (NC)

Supports Multiple Types of Management Traffic

Pass-Thru Management Traffic
- Enable MC-Network communication via NC
- NC-SI Command/Response Packets
  - Command (Response) sent by MC (NC) to NC (MC)
  - Request/Response Semantics
  - Functions: Control, Configuration, Status, Statistics,…
- NC-SI Notification Packets
  - Generated and sent by NC to MC
  - Functions: OS/Link Status Change; NC Soft Reset
NC-SI Transport Bindings

NC-SI Over RMII Based Transport (RBT)
- Defines NC-SI Binding over Reduced Media Independent Interface™ (RMII)
- Physical-level interface is based on RMII
- Media-level interface is based on Ethernet
- Defines hardware arbitration scheme to share single RMII based NC-SI bus

NC-SI over Management Component Transport Protocol (MCTP)
- Enables NC-SI communications over an MCTP network
- Enables communication between an MC/NCs over MCTP-capable interconnects like PCIe/SMBus
- Supports the ability to migrate the NC-SI and pass-through traffic seamlessly from PCIe to SMBus
- 3 levels of addresses: Physical, MCTP – Endpoint ID (EID), NC-SI – Package ID per device, Channel ID per port
- Hardware based arbitration is not required as NC-SI over MCTP assumes a switched media
Platform Level Data Model (PLDM)

An effective interface & data model for efficient access to:
  Low-level platform inventory, BIOS, and config data
  Platform monitoring/control, alerting, event log, etc.

Defines low level data representations and commands

Provides transport independent Request/Response Model

Supports a subtype to distinguish types of PLDM Msgs
  Allows messages to be grouped based on the functions
  Allows the discovery of the functionality supported

PLDM specs: Base, IDs & Codes, SMBIOS data transfer, BIOS control and configuration, Platform Monitoring and Control, FRU, Firmware Update, and Redfish Device Enablement (RDE)
PLDM Messages and PLDM over MCTP Binding

### PLDM Message Fields

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Field Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rq</td>
<td>1 bit</td>
<td>Request bit.</td>
</tr>
<tr>
<td>D</td>
<td>1 bit</td>
<td>Datagram bit.</td>
</tr>
<tr>
<td>rsvd</td>
<td>1 bit</td>
<td>Reserved</td>
</tr>
<tr>
<td>Instance ID</td>
<td>6 bits</td>
<td>Used to identify the instances of a PLDM request.</td>
</tr>
<tr>
<td>PLDM Type Code</td>
<td>6 bits</td>
<td>The PLDM Type field identifies the type of PLDM that is being used.</td>
</tr>
<tr>
<td>PLDM Command Code</td>
<td>8 bits</td>
<td>Identifies the type of operation the message (per PLDM type) is requesting.</td>
</tr>
<tr>
<td>PLDM Completion Code</td>
<td>8 bits</td>
<td>The PLDM Completion Code field provides the status of the operation.</td>
</tr>
<tr>
<td>PLDM Message Payload</td>
<td>Variable</td>
<td>Zero or more bytes of PLDM message payload that is specific to a particular payload type, PLDM type, command code, and/or completion code.</td>
</tr>
<tr>
<td>PLDM Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>PLDM Base</td>
<td>PLDM Messages used to support control and discovery operations for PLDM.</td>
<td></td>
</tr>
<tr>
<td>PLDM for SMBIOS</td>
<td>PLDM Messages used to support SMBIOS data Transfer. Supports both pull/push models of SMBIOS data transfer.</td>
<td></td>
</tr>
</tbody>
</table>
| PLDM for Platform Monitoring and Control | Provides a model for monitoring **Numeric Sensors** (e.g. temperature) and **State Sensors** (e.g. link state)  
Provides a model for control via PLDM **Effecters** (e.g. fan control)  
Provides a model for **platform events** for asynchronous reporting of state changes from sensors and defines an **event log Model** |
| PLDM for BIOS Control and Configuration | Provides internal model to exchange the BIOS config/control data  
Defines the PLDM data structures/messages for communicating BIOS settings, BIOS attributes, Boot configurations, and Boot order settings |
| PLDM for FRU Data                     | PLDM Messages used to support FRU data transfer                                                                                              |
| PLDM for Firmware Update              | Defines messages and data structures for updating firmware or other code objects maintained within the firmware devices of a platform       |
| PLDM for RDE                          | Defines messages and data structures to communicate Redfish operations and events using binary coded JSON and PLDM.                         |
| OEM Specific                          | Reserved for OEM-specific PLDM Commands                                                                                                       |
PMCI Security

New Task Force created within PMCI Working Group

Goals:

Create specification(s) to provide security for PMCI standards and protocols.

Align component authentication and integrity objects across the industry

Specification should be implementable on existing hardware designs.
Do not require changes to existing hardware/silicon.

Can be referenced by other industry standards organizations.
e.g: Security Project of Open Compute Project (OCP)
Call to Action

Join
PMCI Standards are being incorporated in the Design Specification for OCP NIC 3.0
Join the Server Mezzanine Card Subgroup - http://www.opencompute.org/wiki/Server/Mezz to learn more and participate

Learn more about the DMTF & the PMCI suite of standards
Visit the PMCI website - https://www.dmtf.org/standards/pmci
Consider bringing work into the DMTF

Alliance Partners (e.g. OCP)
DMTF Originated Work
  From the DMTF: Work In Progress Release capability, Informational Specifications
  Input to the DMTF: Alliance Liaison, Joint Members, DMTF Technology Adoption, DMTF Feedback Portal
Alliance Partner Originated Work
  Similar mechanisms would speed things along if you wish DMTF input