Platform Management Component Intercommunications (PMCI) Architecture

White Paper

Version 1.0.0a
Status: Informational
Publication Date: July, 2007
DSP2015

Version 1.0.0a
Publication Date: July, 2007
DSP2015
Status: Informational

Abstract

Platform Management Components Intercommunications (PMCI) is a sub-group of the Pre-OS working group within DMTF. PMCI defines the standards to address "inside the box" communication and functional interfaces between the components of the platform management subsystem. The PMCI standards and technologies are complementary to DMTF Common Information Model (CIM) profiles and remote access protocols that are defined in the other DMTF working groups such as Desktop and Mobile Work Group (DMWG), Server Management Work Group (SMWG), and WBEM Infrastructure Protocols (WIP) work group. This document is an architectural white paper that describes the high-level PMCI architecture and concepts.

Acknowledgments

The following persons were instrumental in the development of this white paper:
Hemal Shah – Broadcom Corporation, Tom Slaight – Intel Corporation.

Table of Contents

List of Figures

1 Introduction

Platform Management Components Intercommunications (PMCI) is a sub-group of the Pre-OS working group within DMTF. PMCI defines the standards to address "inside the box" communication and functional 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 work-group (Note: This architecture is referred as PMCI architecture or PMCI hereon). 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 and platform management subsystem implementation.

1.1 Target Audience

The intended target audience for this document is the readers interested in understanding management components intercommunications between the components of platform management subsystems of desktop systems, mobile systems, thin clients, bladed PCs, and servers.

1.2 Related Documents

[1] DSP0136, Alert Standard Format Specification.

[2] DSP0236, Management Component Transport Protocol (MCTP) Base Specification.

[3] DSP0237, Management Component Transport Protocol (MCTP) SMBus / I^2C Transport Binding Specification.

[4] DSP0238, Management Component Transport Protocol (MCTP) PCIe VDM Transport Binding Specification.

[5] DSP0239, Management Component Transport Protocol (MCTP) IDs and Codes Specification.

[6] DSP0222, Network Controller Sideband Interface (NC-SI) Specification.

[7] DSP0134, System Management BIOS (SMBIOS) Specification.

1.3 Terminology

Term	Definition
Intelligent Management Device	Intelligent Management Device. A Management Device that is typically implemented using a microcontroller and accessed via a messaging protocol. Management Parameter access provided by an Intelligent Management Device is typically accomplished using an abstracted interface and data model rather than via direct 'register level' accesses.
Legacy Sensor Device	A Management Device that typically utilizes a register-based low-level interface that is not defined by a given standard.

Term	Definition
Management Controller	A microcontroller or processor that aggregates Management Parameters from one or more Management Devices and makes access to those parameters available to local or remote software, or to other Management Controllers, via one or more management data models.
Management Device	Any physical device that provides protocol terminus for accessing one or more Management Parameters. A Management Device responds to management requests, but does not initiate or aggregate management operations except in conjunction with a Management Controller. An example of a simple Management Device would be a temperature sensor chip.
Management Parameter	A particular datum representing a characteristic, capability, status, or control point associated with a Managed Entity. Example Management Parameters include temperature, speed, volts, on/off, link state, uncorrectable error count, device power state, etc.
Network Controller	The component within a system that is responsible for providing connectivity to an external network world.
PMCI	Platform Management Component Intercommunications. Name for a working group under the Distributed Management Task Force's Pre-OS Workgroup that is chartered to define standardized communication protocols, low-level data models, and transport definitions that support communications with and between Management Controllers and Management Devices that form a platform management subsystem within a managed computer system.
Standardized Sensor Device or Sensor Device	A Management Device that utilizes a register-based low-level interface that is defined by a standard.

1.4 Acronyms and Abbreviations

Term	Definition
IMD	Intelligent Management Device
LSD	Legacy Sensor Device
MC	Management Controller
MCTP	Management Component Transport Protocol
MD	Management Device
NC	Network Controller
NC-SI	Network Controller Sideband Interface
PLDM	Platform Level Data Model
PMCI	Platform Management Component Intercommunications
RMII	Reduced Media Independent Interface

Term	Definition
SD	Sensor Device

2 Architecture Overview

A Platform Management Subsystem in today's enterprise computing platforms is comprised 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.

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.) and control functions (for example, platform power-on/off, reset, watchdog timer, etc.). In DMTF manageability architectures, such as DASH, the platform management subsystem services would be accessed via a Manageability Access Point (MAP) function. The platform management subsystem frequently includes one or more intelligent controllers (microcontrollers) that support access to the management monitoring and control functions, or serve as 'intelligent management devices' that provide monitoring and control services for access by other management controllers in the subsystem.

Currently, there are a set of standards that cover some aspects of the intercommunications between components of the platform management subsystem. For example, ASF 2.0 specification defines alert-related and boot options-related SMBus messages as well as firmware interfaces for ASF configuration and capabilities reporting. System Management BIOS (SMBIOS) specification defines how the motherboard and system hardware information is represented in a standard format by extending the BIOS interface. Network Controller Sideband Interface (NC-SI) defines an interoperable sideband communication interface standard to enable the exchange of management data between the Management Controller (MC) and Network Controller (NC).

All of these efforts have addressed only certain aspects of the platform intercommunications. PMCI captures knowledge from ASF, NC-SI, and SMBIOS specifications efforts and covers all the aspects of intercommunications among platform management subsystem components. PMCI also leverages SMBus, IPMI, PCI-e and other related industry technologies.

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.

Extra emphasis has been placed in the development of PMCI standards to enable lightweight implementations which are architecturally consistent. This has been done to enable a full spectrum of implementations without sacrificing the richness of the PMCI standards. This includes software-only solutions and small footprint firmware solutions. Emphasis has been placed on ensuring that these implementations will be interoperable, independent of implementation, component architecture, platform solutions, vendor or operating environment.

2.1 Principal Goals

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 Section 2.2 to illustrate different types of components and intercommunications within a platform.

Another goal of PMCI is to enable the same semantics, protocols, and interfaces to work across a full range of platforms – traditional desktop systems, mobile, laptop, and server computers, bladed PCs as well as "thin clients".

2.2 Platform Management Subsystem Components

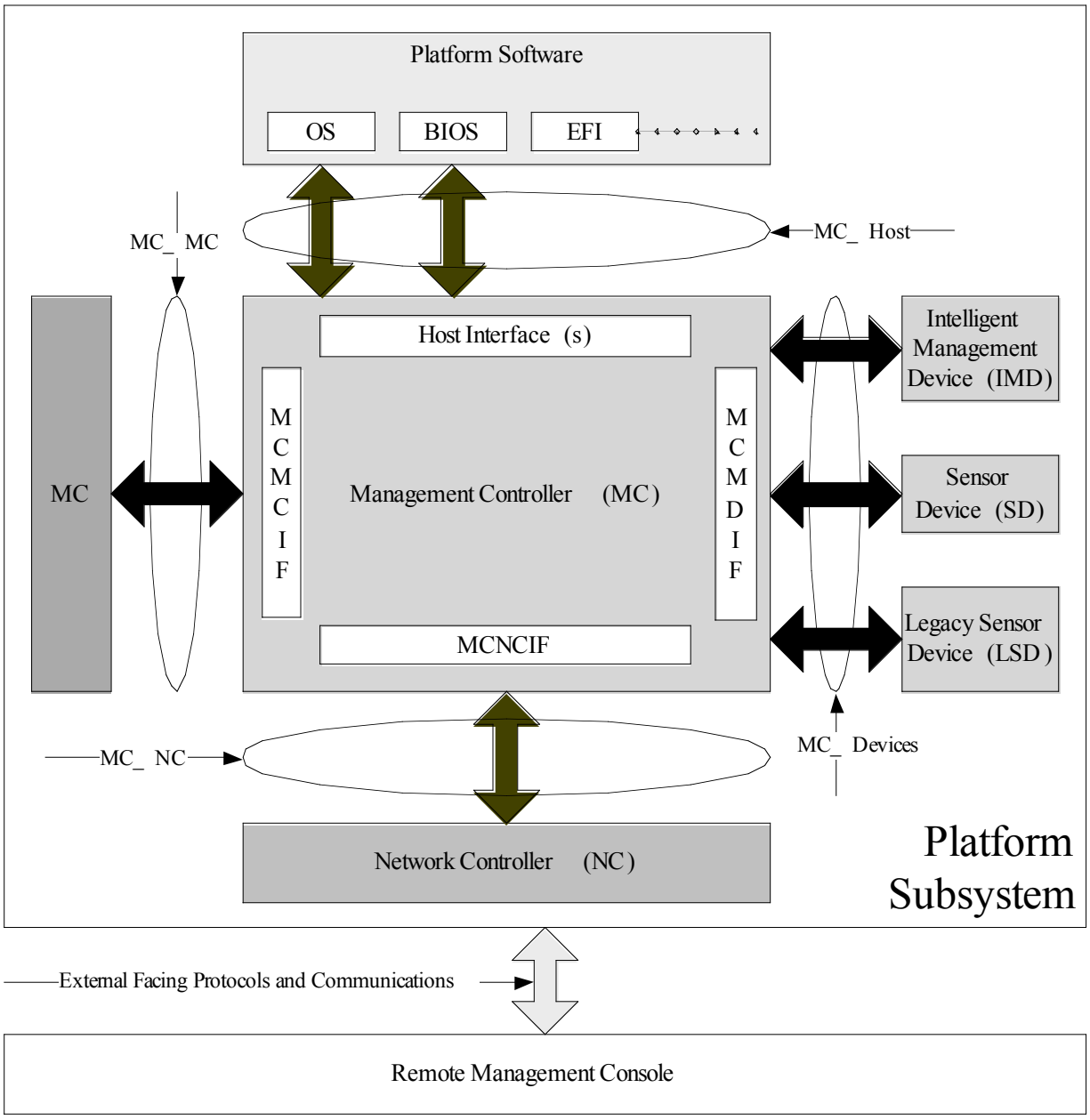

Figure 1: Platform Management Subsystem Components

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 Management Devices and makes access to those parameters available to local or remote software, or to other Management Controllers, via

one or more management data models. The microcontroller or processor that serves as a Management Controller can also incorporate the functions of a Management Device.

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 are BIOS, OS, and EFI firmware.
3. Management Device (MD): A Management Device responds to management requests, but does not initiate or aggregate management operations except in conjunction with a Management Controller. An example of a simple Management Device would be a temperature sensor chip. There are three main types of management devices: standard sensor device (SD) that exposes a standard low-level interface, legacy sensor device (LSD) that uses a register level low-level interface that is not standardized, and intelligent management device (IMD) that provides Management Parameter access typically using an abstracted interface and data model rather than via direct 'register level' accesses.
4. Network Controller (NC): is a component within a system that is responsible for providing connectivity to an external network world. For example, a Gigabit Ethernet network controller.

PMCI covers all four types of intercommunications between the above components. Specifically, PMCI covers the intercommunications between:

1. Management Controller and Host (platform software)
2. Management Controller and Management Devices
3. Management Controller and Network Controller
4. Management Controller and Management Controller

3 PMCI Architecture

3.1 Architecture Model

The PMCI architecture model is shown in Figure 2. The architecture model covers the work areas of PMCI. The following are the four main areas of the PMCI standardization efforts.

1. Interfaces: This covers the types of interconnects and interfaces defined for the platform management. These include MC to MC interface (MCMCIF), MC to MD interface (MCMDIF), MC to NC interface (MCNCIF), and host interfaces.
2. Management Component Transport Protocol (MCTP): is used to move the management data between the components. This provides a common protocol across different interconnects and interfaces.
3. Platform Level Data Model (PLDM): defines how platform level management functions such as inventory, monitoring, control, eventing, and data transfer are abstracted and accessed.
4. PLDM to CIM Mapping: defines how platform level data model maps onto the data model defined by CIM profiles.

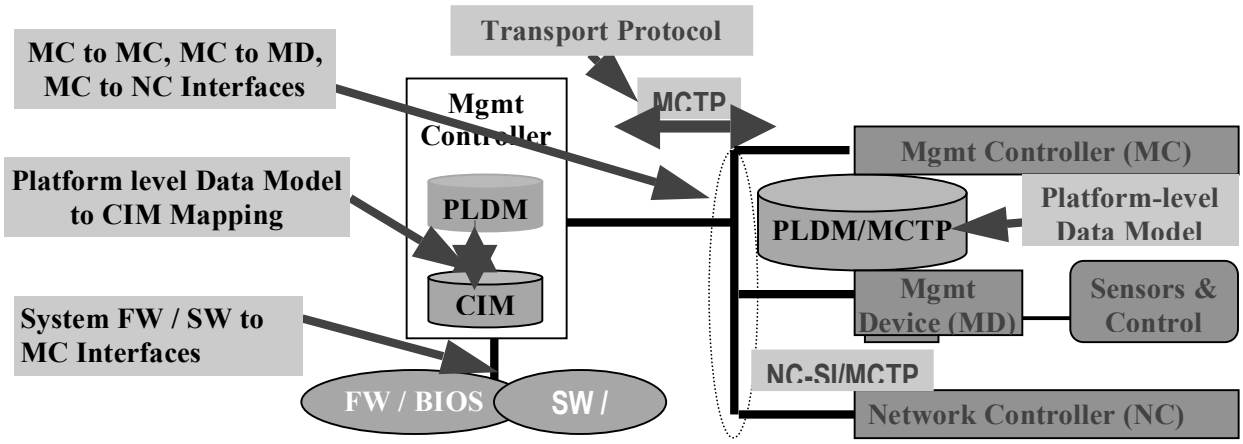

Figure 2: PMCI Architecture Model

3.2 PMCI Stack

Platform Management Components Intercommunications (PMCI) work-group is defining a set of protocols that can be used for communications between the platform components. The following figure shows a simple view of a PMCI stack.

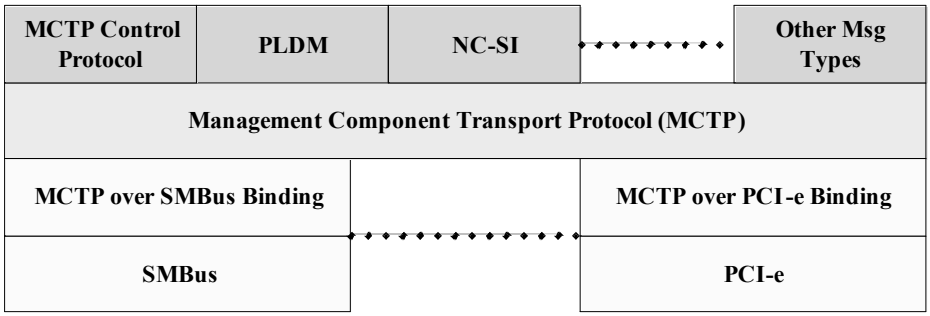

Figure 3: A PMCI Stack

The central component of the PMCI stack is Management Component Transport Protocol (MCTP). MCTP is defined for 'inside the box' communication of platform management traffic. MCTP can carry multiple message types: MCTP control, Platform level data model, Network pass-through, etc. MCTP is suitable for use with multiple media types. The layer below MCTP is the binding layer that is used to bind MCTP over a specific physical medium. The lowest layer shows different physical mediums.

The layers above MCTP define different communication and data models mapped over MCTP. MCTP Control Protocol is used to setup/initialize MCTP control communications within an MCTP network.

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. PLDM is designed to be an effective source for mapping under CIM.

NC-SI/MCTP defines a pass-through model of communications between a management controller and a network controller. The PMCI components are discussed in the next section.

4 PMCI Standards Overview

The PMCI standards are composed of technologies defined in multiple standard specifications, including the Management Component Transport Protocol (MCTP) related specifications, Network Controller Sideband Interface (NC-SI) over MCTP specification, and Platform Level Data Model (PLDM) over MCTP specification.

4.1 Management Component Transport Protocol (MCTP)

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.

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/ I^2C. This approach enables future transport bindings to be defined to support additional buses such as USB, RMII, and others, without affecting the base MCTP specification.

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 simple message fragmentation/reassembly mechanism that allows large data transfers using MCTP packetization.

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.

4.2 Platform Level Data Model (PLDM)

The Platform Level Data Model (PLDM) is being designed to be an effective data and control source for mapping under CIM. PLDM is targeted to provide an efficient access to low-level platform inventory, monitoring, control, eventing, and data/parameters transfer functions such as temperature, fan, voltage, event logging, and boot control.

PLDM is defining data representations and commands that abstract the platform management hardware. The PLDM specification work includes:

1. Messages and data model for SMBIOS data transfer within the platform.
2. Messages and data structures for Field Replaceable Unit (FRU), asset information, and firmware inventory data transfer.
3. Messages and data structures for monitoring processors, caches, memory, sensors, fans, power state monitoring, time stamp clock monitoring, etc.
4. Control messages/data structures for sensors, fans, power state mgmt, boot control, real time stamp, and watchdog timer.

5. Low level data models and messages to represent and transfer opaque data, BIOS data, and event data.
6. Messages to transfer text console redirection and media redirection related messages.

4.3 Pass-through Communications

For pass-through communications, the Pre-OS Sideband sub-group within the Pre-OS WG has defined a sideband interface and protocol to transfer management traffic between a management controller and network controller. The Network Controller Sideband Interface (NC-SI) specifies a Sideband Interface that uses RMII as a physical 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. One of the usage models envisioned for MCTP is a Sideband Interface between one or more Management Controllers and one or more Network Controllers. PMCI is planning to specify an alternative NC-SI mapping to MCTP.

5 Conclusion

PMCI covers the standards to address “inside the box” communication and functional interfaces between the components of the platform management subsystem. The PMCI standards and technologies are complementary to DMTF CIM profiles and remote access protocols that are defined in the other DMTF working groups such as Desktop and Mobile Work Group (DMWG), Server Management Work Group (SMWG), and WBEM Infrastructure Protocols (WIP) work group.

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.