



1

2

3

**Platform Management Component Intercommunications (PMCI)  
Architecture**

4

5

6

**White Paper**

7

8

**Version 1.0.0a**

9

**Status: Informational**

10

**Publication Date: July, 2007**

11

**DSP2015**

12 Copyright © 2007 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

13 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems manage-  
14 ment and interoperability. Members and non-members may reproduce DMTF specifications and documents for uses  
15 consistent with this purpose, provided that correct attribution is given. As DMTF specifications may be revised  
16 from time to time, the particular version and release date should always be noted.

17 Implementation of certain elements of this standard or proposed standard may be subject to third party patent rights,  
18 including provisional patent rights (herein "patent rights"). DMTF makes no representations to users of the standard  
19 as to the existence of such rights, and is not responsible to recognize, disclose, or identify any or all such third party  
20 patent right, owners or claimants, nor for any incomplete or inaccurate identification or disclosure of such rights,  
21 owners or claimants. DMTF shall have no liability to any party, in any manner or circumstance, under any legal the-  
22 ory whatsoever, for failure to recognize, disclose, or identify any such third party patent rights, or for such party's  
23 reliance on the standard or incorporation thereof in its product, protocols or testing procedures. DMTF shall have no  
24 liability to any party implementing such standard, whether such implementation is foreseeable or not, nor to any  
25 patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is  
26 withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the  
27 standard from any and all claims of infringement by a patent owner for such implementations.

28 For information about patents held by third-parties which have notified the DMTF that, in their opinion, such patent  
29 may relate to or impact implementations of DMTF standards, visit  
30 <http://www.dmtf.org/about/policies/disclosures.php>.

32  
33  
34  
35  
36

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

37 **Abstract**

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

46 **Acknowledgments**

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



## Table of Contents

50	Abstract.....	3
51	Acknowledgments.....	3
52	1 Introduction .....	6
53	1.1 Target Audience.....	6
54	1.2 Related Documents.....	6
55	1.3 Terminology.....	6
56	1.4 Acronyms and Abbreviations .....	7
57	2 Architecture Overview.....	9
58	2.1 Principal Goals.....	9
59	2.2 Platform Management Subsystem Components.....	10
60	3 PMCI Architecture.....	12
61	3.1 Architecture Model.....	12
62	3.2 PMCI Stack .....	12
63	4 PMCI Standards Overview.....	14
64	4.1 Management Component Transport Protocol (MCTP).....	14
65	4.2 Platform Level Data Model (PLDM) .....	14
66	4.3 Pass-through Communications .....	15
67	5 Conclusion.....	16

68

## 69 List of Figures

70	Figure 1: Platform Management Subsystem Components.....	10
71	Figure 2: PMCI Architecture Model.....	12
72	Figure 3: A PMCI Stack.....	13

73 **1 Introduction**

74 Platform Management Components Intercommunications (PMCI) is a sub-group of the Pre-OS  
75 working group within DMTF. PMCI defines the standards to address “inside the box” communi-  
76 cation and functional interfaces between the components of the platform management subsystem.

77 This document lays forth the basic architectural concepts that are driving the specifications being  
78 defined by the PMCI work-group (Note: This architecture is referred as PMCI architecture or  
79 PMCI hereon). The focus of PMCI architecture is to enable intercommunications between differ-  
80 ent management components of a platform management subsystem in a standard manner across  
81 any implementation of a management component, independent of the operating system state and  
82 platform management subsystem implementation.

83 **1.1 Target Audience**

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

87 **1.2 Related Documents**

- 88 [1] DSP0136, Alert Standard Format Specification.
- 89 [2] DSP0236, Management Component Transport Protocol (MCTP) Base Specification.
- 90 [3] DSP0237, Management Component Transport Protocol (MCTP) SMBus / I<sup>2</sup>C Transport  
91 Binding Specification.
- 92 [4] DSP0238, Management Component Transport Protocol (MCTP) PCIe VDM Transport  
93 Binding Specification.
- 94 [5] DSP0239, Management Component Transport Protocol (MCTP) IDs and Codes Specifica-  
95 tion.
- 96 [6] DSP0222, Network Controller Sideband Interface (NC-SI) Specification.
- 97 [7] DSP0134, System Management BIOS (SMBIOS) Specification.

98

99 **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.

<b>Term</b>	<b>Definition</b>
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.

100

## 1.4 Acronyms and Abbreviations

<b>Term</b>	<b>Definition</b>
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



## 101 **2 Architecture Overview**

102 A Platform Management Subsystem in today's enterprise computing platforms is comprised of a  
103 set of components which communicate to perform management functions within the platform. In  
104 many cases, these communications and interfaces are specialized and adapted to each individual  
105 platform, installation and component in the environment.

106 A platform management subsystem provides hardware management services such as platform  
107 environmental monitoring functions (for example, temperature probing, voltage monitoring, fan  
108 speeds, hardware error status, etc.) and control functions (for example, platform power-on/off,  
109 reset, watchdog timer, etc.). In DMTF manageability architectures, such as DASH, the platform  
110 management subsystem services would be accessed via a Manageability Access Point (MAP)  
111 function. The platform management subsystem frequently includes one or more intelligent con-  
112 trollers (microcontrollers) that support access to the management monitoring and control func-  
113 tions, or serve as 'intelligent management devices' that provide monitoring and control services  
114 for access by other management controllers in the subsystem.

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

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

127 PMCI supports a suite of specifications which include architectural semantics, industry standard  
128 protocols, and platform level data models to standardize the management related intercommuni-  
129 cations between the components of platform management subsystem independent of component  
130 implementation, platform state, and platform management subsystem implementation.

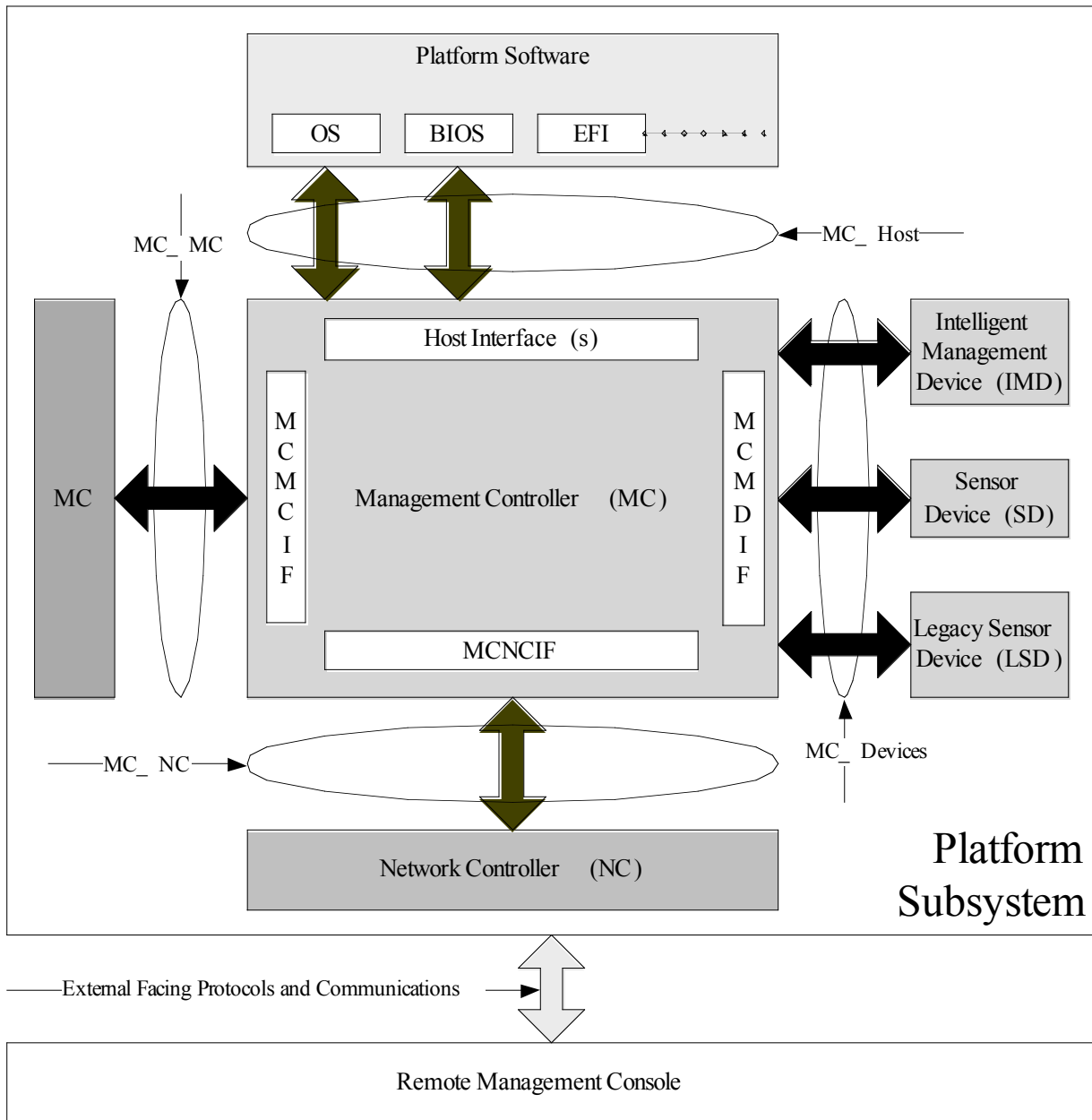
131 Extra emphasis has been placed in the development of PMCI standards to enable lightweight im-  
132 plementations which are architecturally consistent. This has been done to enable a full spectrum  
133 of implementations without sacrificing the richness of the PMCI standards. This includes soft-  
134 ware-only solutions and small footprint firmware solutions. Emphasis has been placed on ensur-  
135 ing that these implementations will be interoperable, independent of implementation, component  
136 architecture, platform solutions, vendor or operating environment.

### 137 **2.1 Principal Goals**

138 One goal of PMCI is to enable intercommunications between different types of platform compo-  
139 nents using a set of standards protocols, interfaces, and platform level data models. An example  
140 of the platform management subsystem is provided in Section 2.2 to illustrate different types of  
141 components and intercommunications within a platform.

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

145 **2.2 Platform Management Subsystem Components**  
 146



147 **Figure 1: Platform Management Subsystem Components**

148 Figure 1 shows the different components within a platform management subsystem. The compo-  
 149 nents can be divided into the following four categories:

- 151 1. Management Controller (MC): A microcontroller or processor that aggregates Manage-  
 152 ment Parameters from one or more Management Devices and makes access to those pa-  
 153 rameters available to local or remote software, or to other Management Controllers, via

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

156 2. Platform Software: The software running on the host CPUs that communicates with a  
157 management controller for performing a set of management functions. The examples of  
158 the platform software are BIOS, OS, and EFI firmware.

159 3. Management Device (MD): A Management Device responds to management requests,  
160 but does not initiate or aggregate management operations except in conjunction with a  
161 Management Controller. An example of a simple Management Device would be a tem-  
162 perature sensor chip. There are three main types of management devices: standard sensor  
163 device (SD) that exposes a standard low-level interface, legacy sensor device (LSD) that  
164 uses a register level low-level interface that is not standardized, and intelligent manage-  
165 ment device (IMD) that provides Management Parameter access typically using an ab-  
166 stracted interface and data model rather than via direct 'register level' accesses.

167 4. Network Controller (NC): is a component within a system that is responsible for provid-  
168 ing connectivity to an external network world. For example, a Gigabit Ethernet network  
169 controller.

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

172 1. Management Controller and Host (platform software)

173 2. Management Controller and Management Devices

174 3. Management Controller and Network Controller

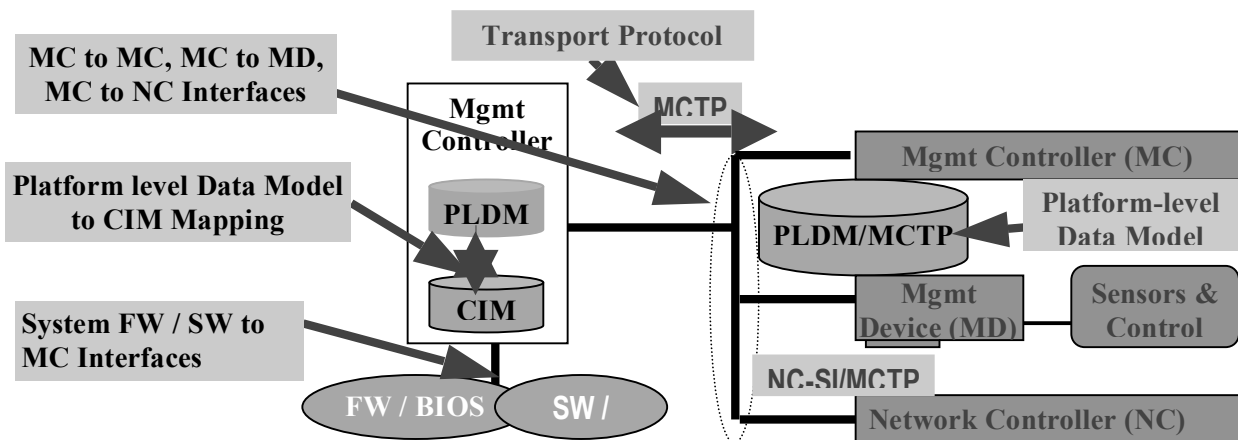
175 4. Management Controller and Management Controller

176 **3 PMCI Architecture**

177 **3.1 Architecture Model**  
178

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

- 181 1. Interfaces: This covers the types of interconnects and interfaces defined for the platform  
182 management. These include MC to MC interface (MCMCIF), MC to MD interface  
183 (MCMDIF), MC to NC interface (MCNCIF), and host interfaces.
- 184 2. Management Component Transport Protocol (MCTP): is used to move the management  
185 data between the components. This provides a common protocol across different inter-  
186 connects and interfaces.
- 187 3. Platform Level Data Model (PLDM): defines how platform level management functions  
188 such as inventory, monitoring, control, eventing, and data transfer are abstracted and ac-  
189 cessed.
- 190 4. PLDM to CIM Mapping: defines how platform level data model maps onto the data  
191 model defined by CIM profiles.  
192  
193



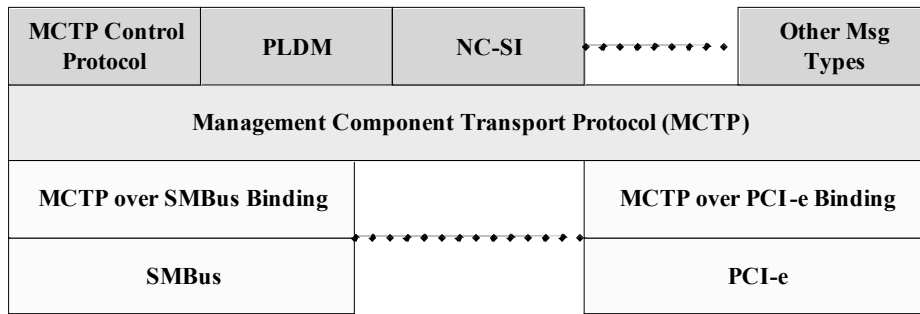
194  
195  
196  
197  
198

**Figure 2: PMCI Architecture Model**

199 **3.2 PMCI Stack**  
200

201 Platform Management Components Intercommunications (PMCI) work-group is defining a set of  
202 protocols that can be used for communications between the platform components. The following  
203 figure shows a simple view of a PMCI stack.  
204

205



206

207

208

**Figure 3: A PMCI Stack**

209

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.

215

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.

218

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.

223

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.

224

## 225 **4 PMCI Standards Overview**

226  
227 The PMCI standards are composed of technologies defined in multiple standard specifications,  
228 including the Management Component Transport Protocol (MCTP) related specifications, Net-  
229 work Controller Sideband Interface (NC-SI) over MCTP specification, and Platform Level Data  
230 Model (PLDM) over MCTP specification.

### 231 **4.1 Management Component Transport Protocol (MCTP)**

232  
233 The Management Component Transport Protocol (MCTP) is a protocol for intercommunications  
234 among intelligent devices within a platform management subsystem. This protocol is independ-  
235 ent of the underlying physical bus properties, as well as the "data-link" layer messaging used on  
236 the bus.

237  
238 The physical and data-link layer methods for MCTP communication across a given medium are  
239 defined by companion "transport binding" specifications, such as MCTP over PCIe® Vendor  
240 Defined Messaging and MCTP over SMBus/ I<sup>2</sup>C. This approach enables future transport bind-  
241 ings to be defined to support additional buses such as USB, RMI, and others, without affecting  
242 the base MCTP specification.

243  
244 The MCTP communication model includes a message format, transport description, message ex-  
245 change patterns, and operational Endpoint characteristics. MCTP uses logical addressing based  
246 on Endpoint IDs that enables static/dynamic endpoint ID assignments as well as bridging/routing  
247 support. MCTP defines simple message fragmentation/reassembly mechanism that allows large  
248 data transfers using MCTP packetization.

249  
250 MCTP Control Protocol is used to setup/initialize MCTP control communications within an  
251 MCTP network. MCTP Control Protocol supports request/response, broadcast, and one-way  
252 communications.

### 253 **4.2 Platform Level Data Model (PLDM)**

254  
255 The Platform Level Data Model (PLDM) is being designed to be an effective data and control  
256 source for mapping under CIM. PLDM is targeted to provide an efficient access to low-level  
257 platform inventory, monitoring, control, eventing, and data/parameters transfer functions such as  
258 temperature, fan, voltage, event logging, and boot control.

259  
260 PLDM is defining data representations and commands that abstract the platform management  
261 hardware. The PLDM specification work includes:

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

- 269 5. Low level data models and messages to represent and transfer opaque data, BIOS data,  
270 and event data.
- 271 6. Messages to transfer text console redirection and media redirection related messages.  
272

### 273 **4.3 Pass-through Communications**

274

275 For pass-through communications, the Pre-OS Sideband sub-group within the Pre-OS WG has  
276 defined a sideband interface and protocol to transfer management traffic between a management  
277 controller and network controller. The Network Controller Sideband Interface (NC-SI) specifies  
278 a Sideband Interface that uses RMII as a physical transport. NC-SI defines the formats for com-  
279 municating network traffic, control commands, responses, and asynchronous event notifications  
280 between a management controller and a network controller. One of the usage models envisioned  
281 for MCTP is a Sideband Interface between one or more Management Controllers and one or  
282 more Network Controllers. PMCI is planning to specify an alternative NC-SI mapping to MCTP.

283 **5 Conclusion**

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

289 PMCI supports a suite of specifications which include architectural semantics, industry standard  
290 protocols, and platform level data models to standardize the management related intercommuni-  
291 cations between the components of platform management subsystem independent of component  
292 implementation, platform state, and platform management subsystem implementation.

293