Aspect Architecture Document

Pikes Peak hardware platform

Document: 45555-00
Preliminary Revision: P:A.4:A.3:A.4:A.3:A.3:A.1:A.3:A.1:A.4

Revision Date: 3/22/2010

TABLE OF CONTENTS

Table of Contents **1**
List of Figures **6**
List of Tables **7**
Source Document Information **8**
1. Pikes Peak hardware platform - An Overview **12**
1.1. Aspect Introduction 12
1.1.1. Aspect Description 12
1.1.2. Assumptions 14
1.1.3. Related Documents 14
1.1.4. Open Issues 15
1.1.5. AAD References 16
1.2. Aspect High Level Requirements 16
1.2.1. Product Requirements 16
1.2.2. AAD Overview 17
2. Pikes Peak Controller - Element Requirements **18**
2.1. Pikes Peak hardware platform - Element Requirements 18
2.1.1. Introduction 18
2.1.2. Operational Behavior 18
2.1.2.1. Host Interface Characteristics 18
2.1.2.2. Drive Interface Characteristics 18
2.1.2.3. General Controller Characteristics 19
2.1.2.4. Controller Identifiers 21
2.1.2.5. Battery Management 21
2.1.2.6. Smart Battery Parameters 22
2.1.2.7. Seven Segment Display Support 23
2.1.2.8. Protection Information Capability 23
2.1.2.9. Field Replaceable Cache Memory 24
2.1.2.10. Field Replaceable Cache Backup Device 24
2.1.2.11. Field Replaceable HICs 24
2.1.2.12. Push Button Support 25
2.1.2.13. USB Port 25
2.1.2.14. miniUSB Port 25
2.1.2.15. Power Supply 25
2.1.2.16. Wake on LAN Support (WOL) 26
2.2. Controller Type Mismatch - Element Requirements 26
2.2.1. Introduction 26
2.2.2. Operational Behavior 26
2.2.2.1. Controller Hot-Swapped with Incorrect Controller Type 26
2.2.2.2. Controller Hot-Swapped with same Controller Type 26
2.2.2.3. Controller A or B Cold-Swapped with Incorrect Controller Type 26
2.2.2.4. Power Cycle after Hot-Swap with Incorrect Controller Type 27
2.2.2.5. Controller-Drive Enclosure Considerations 27
2.2.2.6. Dual Controller Swap 27

2.2.2.7. Start-of-Day Mismatch Detection..... 27
2.2.2.8. Run-Time Mismatch Detection..... 28
2.2.2.9. Failure to Determine Alternate Board Identifier..... 28
2.2.3. Administrative and Configuration Interfaces..... 28
2.2.3.1. Controller Mismatch Reporting..... 28
2.2.4. Error Handling and Event Notification..... 28
2.2.4.1. Controller Type Mismatch..... 28
2.2.4.2. Device Reported Events..... 28
2.3. Host Interface Cards - Element Requirements..... 29
2.3.1. Introduction..... 29
2.3.2. Operational Behavior..... 29
2.3.2.1. Fibre Channel Host Interface Card (Manitou)..... 29
2.3.2.2. InfiniBand Host Interface Card (Elk Park)..... 29
2.3.2.3. iSCSI-SFP+ Host Interface Card (Glen Cove)..... 29
2.4. SATA Flash Disk Support - Element Requirements..... 30
2.4.1. Introduction..... 30
2.4.2. Operational Behavior..... 30
2.4.2.1. Partitioning..... 30
2.4.2.2. Default boot partition..... 31
2.4.2.3. Mirroring..... 31
2.4.2.4. Disk Configuration Information..... 31
2.4.2.5. Disk Management and Replacement..... 31
2.4.2.6. SATA flash disk configurations..... 31
2.4.3. Administrative and Configuration Interfaces..... 32
2.4.3.1. Management Interface..... 32
2.4.4. Error Handling and Event Notification..... 32
2.4.4.1. Invalid SATA flash disk configurations..... 32
2.5. Auto MDIX - Element Requirements..... 32
2.5.1. Element Functional Behavior Changes..... 32
2.5.2. Introduction..... 32
2.5.3. Operational Behavior..... 33
2.5.3.1. Ethernet Ports..... 33
2.6. ECC and PCI Error Handling - Element Requirements..... 33
2.6.1. Introduction..... 33
2.6.2. Operational Behavior..... 33
2.6.2.1. ECC Error Detection and Recovery..... 33
2.6.2.2. Patrol Scrub..... 33
2.6.2.3. PCI/PCIe Error Detection and Recovery..... 34
2.6.3. Error Handling and Event Notification..... 34
2.6.3.1. ECC Error..... 34
2.6.3.2. PCI/PCIe Error..... 34
2.7. Wake on LAN Support - Element Requirements..... 34
2.7.1. Introduction..... 34
3. Pikes Peak hardware platform - Detailed Architecture..... 35
3.1. Pikes Peak hardware platform - Detailed Architecture..... 35
3.1.1. High Level Design..... 35

3.1.1.1. Details ... 35
3.1.1.1.1. Local Controller Baseboard ID ... 35
3.1.1.1.2. Alternate Controller Baseboard ID ... 35
3.1.1.1.2.1. Error getting alternate controller's constructed baseboard ID ... 37
3.1.1.1.3. Host Board ID ... 37
3.1.1.1.4. SBB Validation ... 37
3.1.1.1.4.1. SBB Validation Success ... 38
3.1.1.1.4.2. SBB Validation Failure ... 40
3.1.1.1.5. Controller Type Mismatch ... 42
3.1.1.1.5.1. Scenario 1 (Alternate controller is up and internal NIC is working normally) ... 42
3.1.1.1.5.2. Scenario 2 (Alternate controller is physically in-place but running late) ... 42
3.1.1.1.5.3. Scenario 3 (Alternate controller is physically in-place but not responding in a timely manner) ... 43
3.1.1.1.5.4. Scenario 4 (Alternate controller is not in-place) ... 44
3.1.1.1.5.5. Scenario 5 (Alternate controller is not in-place but inserted later) ... 44
3.1.1.1.5.6. Special Lockdown ... 45
3.1.1.1.6. Power Supply SBB validation and PMBus support ... 46
3.1.1.1.7. HIC Support ... 46
3.1.1.1.7.1. FC and InfiniBand ... 46
3.1.1.1.7.2. Glen Cove (iSCSI and Ethernet HIC) ... 47
3.1.1.1.7.2.1. Scenarios ... 49
3.1.1.1.7.3. Controller Lockdown due to mismatch or unsupported HIC ... 52
3.1.1.1.7.4. Cache backup device diagnostics ... 53
3.1.1.1.8. Verify DIMM configuration ... 53
3.1.1.1.9. Disable USB Port ... 54
3.1.1.1.10. Push Button Support ... 55
3.1.1.1.11. SATA Flash handling ... 55
3.1.1.1.11.1. Flash Layout record ... 55
3.1.1.1.11.2. Flash partitioning ... 55
3.1.1.1.11.3. Default Boot Partition ... 55
3.1.1.1.11.4. Invalid SATA flash disk configurations ... 55
3.1.1.1.12. CPU Temperature Monitoring ... 58
3.1.1.1.12.1. Setting CPU Threshold Temperature ... 58
3.1.1.1.12.2. CPU over temperature condition (Threshold # 1) detected ... 60
3.1.1.1.12.3. Abatement of CPU over temperature condition (Threshold # 1) ... 62
3.1.1.1.12.4. CPU over temperature condition (Threshold # 2) detected ... 62
3.1.1.1.13. ECC error handling ... 62
3.1.1.1.14. PCI/PCIe error handling ... 62
3.1.1.1.15. Wake On LAN (WOL) ... 62
3.1.1.1.16. NTB ... 62
3.1.2. Core Assets ... 62
3.1.2.1. Core Application Services ... 63
3.1.2.1.1. [IOVM] brdm - Board Manager ... 63
3.1.2.1.2. [Domain0] brdm - Board Manager ... 63
3.1.2.1.3. [IOVM] ssm - Sub System Monitor ... 63
3.1.2.1.4. [IOVM] hcvh - Hardware Configuration Validation Handler ... 64

3.1.2.1.5. [Domain0] hcvh - Hardware Configuration Validation Handler..... 64
3.1.2.2. Diagnostic Services..... 64
3.1.2.2.1. [IOVM] bcdm - Base Controller Diagnostics Manager..... 64
3.1.2.3. Firmware Architecture..... 64
3.1.2.3.1. VariationMgmt – Variation Management Tools..... 64
3.1.2.3.1.1. Gears Variable..... 64
3.1.2.3.1.1.1. hostboard::Supported Boards..... 64
3.1.2.3.1.2. Module and Mixin:..... 65
3.1.2.3.1.2.1. Hostboard..... 65
3.1.2.3.1.3. Recipe Process..... 67
3.1.2.4. Foundations 1..... 67
3.1.2.4.1. [IOVM] SYMbol API..... 67
3.1.2.4.2. [IOVM] Meldb – Major Event Log Database..... 67
3.1.2.4.2.1. Invalid SATA flash disk configurations..... 67
3.1.2.4.2.2. CPU Thermal Control Circuit activated..... 68
3.1.2.5. Foundations 2..... 68
3.1.2.5.1. [Domain0] vmmgr - Virtual Machine Manager..... 68
3.1.2.5.2. [Domain0] olm - OSA Lockdown Manager..... 69
3.1.2.5.3. [Domain0] fpmgr - Flash Partition Manager..... 69
3.1.2.5.4. [IOVM] cmgr – Controller Manager..... 69
3.1.2.5.5. [IOVM lem - Lockdown Error Manager..... 69
3.1.2.6. IO Interfaces 1..... 69
3.1.2.6.1. [IOVM] ioni - I/O Network Interface..... 69
3.1.2.7. IO Interfaces 3..... 70
3.1.2.7.1. [IOVM] isni - iSCSI Network Interface..... 70
3.1.2.7.2. [IOVM] b_isn - Breckenridge iSCSI network Manager..... 70
3.1.2.7.3. [IOVM] mtlsebe..... 70
3.1.2.7.4. [IOVM] be2nic..... 70
3.1.2.8. IO Interfaces 4..... 70
3.1.2.8.1. [IOVM] ib_hw – IB Hardware Specific Driver..... 71
3.1.2.8.2. [IOVM] ib_core – InfiniBand Core Driver..... 71
3.1.2.8.3. [IOVM] ibHcaFw – InfiniBand HCA Firmware Image..... 71
3.1.2.8.4. [IOVM] LCL – Linux Compatibility Layer..... 71
3.1.2.8.5. [IOVM] STP_SRP – SCSI RDMA Protocol..... 71
3.1.2.9. Platforms..... 71
3.1.2.9.1. [IOVM] BCM – Board Configuration Module..... 71
3.1.2.9.2. [IOVM] common – Common PCI Definitions..... 71
3.1.2.9.3. [IOVM] Diag – Diagnostic Platform Module..... 71
3.1.2.9.4. [IOVM] mpm - Midplane Manager..... 72
3.1.2.9.5. [IOVM] pwsplm - Power Supply Manager..... 72
3.1.2.9.6. BIOS..... 72
3.1.2.10. Volume IO Services..... 72
3.1.2.10.1. [IOVM] ccm - Cache Configuration Manager..... 72
3.1.2.10.2. [IOVM] rpa - RAID Parity Assist..... 72
3.1.2.10.3. [IOVM] cache - Cache Management..... 72
3.1.2.10.4. [IOVM] pbm - Persistent Backup Manager..... 72

3.1.2.11. Hypervisor..... 73
3.1.2.11.1. [Domain0] ivmhb - Inter VM Heartbeat Manager..... 73
3.1.2.11.2. XenStore Key/Values..... 73

LIST OF FIGURES

Figure 1: Pikes Peak SBB 2.0 Controller..... 12
Figure 2: Retrieve Alternate controller's Baseboard ID..... 36
Figure 3: SBB Validation success sequence relative to SOD..... 38
Figure 4: SBB Validation failure sequence relative to SOD..... 40
Figure 5: Controller Type Mismatch - Scenario 1..... 42
Figure 6: Controller Type Mismatch - Scenario 2..... 43
Figure 7: Controller Type Mismatch - Scenario 3..... 43
Figure 8: Controller Type Mismatch - Scenario 5..... 44
Figure 9: PCI function assignments for Glen Cove..... 47
Figure 10: DIMM Verification..... 53
Figure 11: SATA flash disk configuration validation..... 56
Figure 12: Setting CPU temperature threshold values..... 58
Figure 13: CPU Over temperature condition detected..... 60

LIST OF TABLES

Table 1: Pikes Peak Host Interface (On base board) Characteristics..... 18
Table 2: Pikes Peak Drive Interface Characteristics..... 19
Table 3: Pikes Peak General Controller Characteristics..... 19
Table 4: Pikes Peak Controller Identifiers and Model Names..... 21
Table 5: Pikes Peak Controller Smart Battery Configuration..... 22
Table 6: Pikes Peak Controller Smart Battery Attributes..... 22
Table 7: Pikes Peak Controller Smart Battery Temperature Logging Ranges (deg C)..... 22
Table 8: Pikes Peak Component Failure Identification..... 23
Table 9: Pikes Peak Protection Information Capability..... 24
Table 10: Cache Backup Device Partition..... 30
Table 11: Ctlr Lockdown scenario due to HICs..... 53
Table 14: XenStore Key/Values..... 73

Source Document Information

Section 1. Pikes Peak hardware platform - An Overview
Section 2.1. Pikes Peak hardware platform - Element Requirements
Section 2.2. Controller Type Mismatch - Element Requirements
Section 2.3. Host Interface Cards - Element Requirements
Section 2.4. SATA Flash Disk Support - Element Requirements
Section 2.5. Auto MDIX - Element Requirements
Section 2.6. ECC and PCI Error Handling - Element Requirements
Section 2.7. Wake on LAN Support - Element Requirements
Section 3.1. Pikes Peak hardware platform - Detailed Architecture

Section 1. Pikes Peak hardware platform - An Overview

Type: overview
Document: 45555-00
Revision: A.4
Revision Date: 3/22/2010
Author(s): Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
A.4	Incorporated comments from summit and issues discovered so far. The tool has issues with track changes enabled and hence changes are without track changes enabled.
A.3	Incorporated review comments
A.2	Incorporated Review comments
A.1	Initial revision.

Section 2.1. Pikes Peak hardware platform - Element Requirements

Type: requirement
Document: 00000-00
Revision: A.3
Revision Date: 3/22/2010
Author(s): Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
A.4	Incorporated review comments

Revision	Description of Changes
A.3	Incorporated review comments
A.2	Incorporated Review Comments
A.1	Initial revision.

Section 2.2. Controller Type Mismatch - Element Requirements

Type: requirement
Document: 45552-00
Revision: A.4
Revision Date: 3/22/2010
Author(s): Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
A.4	Incorporated the comments from discussions so far
A.3	Incorporated review comments
A.2	Incorporated review comments
A.1	Initial revision.

Section 2.3. Host Interface Cards - Element Requirements

Type: requirement
Document: 44339-00
Revision: A.3
Revision Date: 3/21/2010
Author(s): Ashish Batwara, Jim Lynn

REVISION HISTORY

Revision	Description of Changes
A.3	Glen cove supports 10 GB only
A.2	Incorporated Review comments
A.1	Initial revision.

Section 2.4. SATA Flash Disk Support - Element Requirements

Type: requirement
Document: 45550-00
Revision: A.3

Revision Date: 3/22/2010
Author(s): Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
A.4	Incorporated review comments. There are many open items and items under Architecture note.
A.3	Incorporated review comments
A.2	Incorporated Review comments
A.1	Initial revision.

Section 2.5. Auto MDIX - Element Requirements

Type: requirement
Document: 44889-00
Revision: A.1
Revision Date: 3/21/2010
Author(s): Aaron Dailey, Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
Revision	Description of Changes
A.1	Initial revision.

Section 2.6. ECC and PCI Error Handling - Element Requirements

Type: requirement
Document: 44539-00
Revision: A.3
Revision Date: 3/21/2010
Author(s): Ashish Batwara, Aaron Dailey

REVISION HISTORY

Revision	Description of Changes
A.3	Incorporated review comments
A.2	Incorporated Review comments
A.1	Initial revision.

Section 2.7. Wake on LAN Support - Element Requirements

Type: requirement
Document: 00000-00
Revision: A.1
Revision Date: 3/21/2010
Author(s): Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
Revision	Description of Changes
A.1	Initial revision.

Section 3.1. Pikes Peak hardware platform - Detailed Architecture

Type: architecture
Document: 45555-00
Revision: A.4
Revision Date: 3/22/2010
Author(s): Ashish Batwara

REVISION HISTORY

Revision	Description of Changes
A.4	Incorporated comments from summit and issues discovered so far. There are major changes in the rev, hence a cleaned version.
A.3	Incorporated review comments
A.2	Incorporated review comments
A.1	Initial revision.

1. Pikes Peak hardware platform - An Overview

1.1.Aspect Introduction

1.1.1.Aspect Description

The Pikes Peak RAID Controller is designed for use in a SBB 2.0 Compliant Enclosure. The controller platform provides IA multi-core processor support and is designed for heterogeneous host connection to SAS 2.0 disk drives. On the host side it provides a quad 8 Gb/s Fibre Channel connection using an onboard Tachyon QE8 device. Other host connections are provided through a replaceable, modular host interface board.

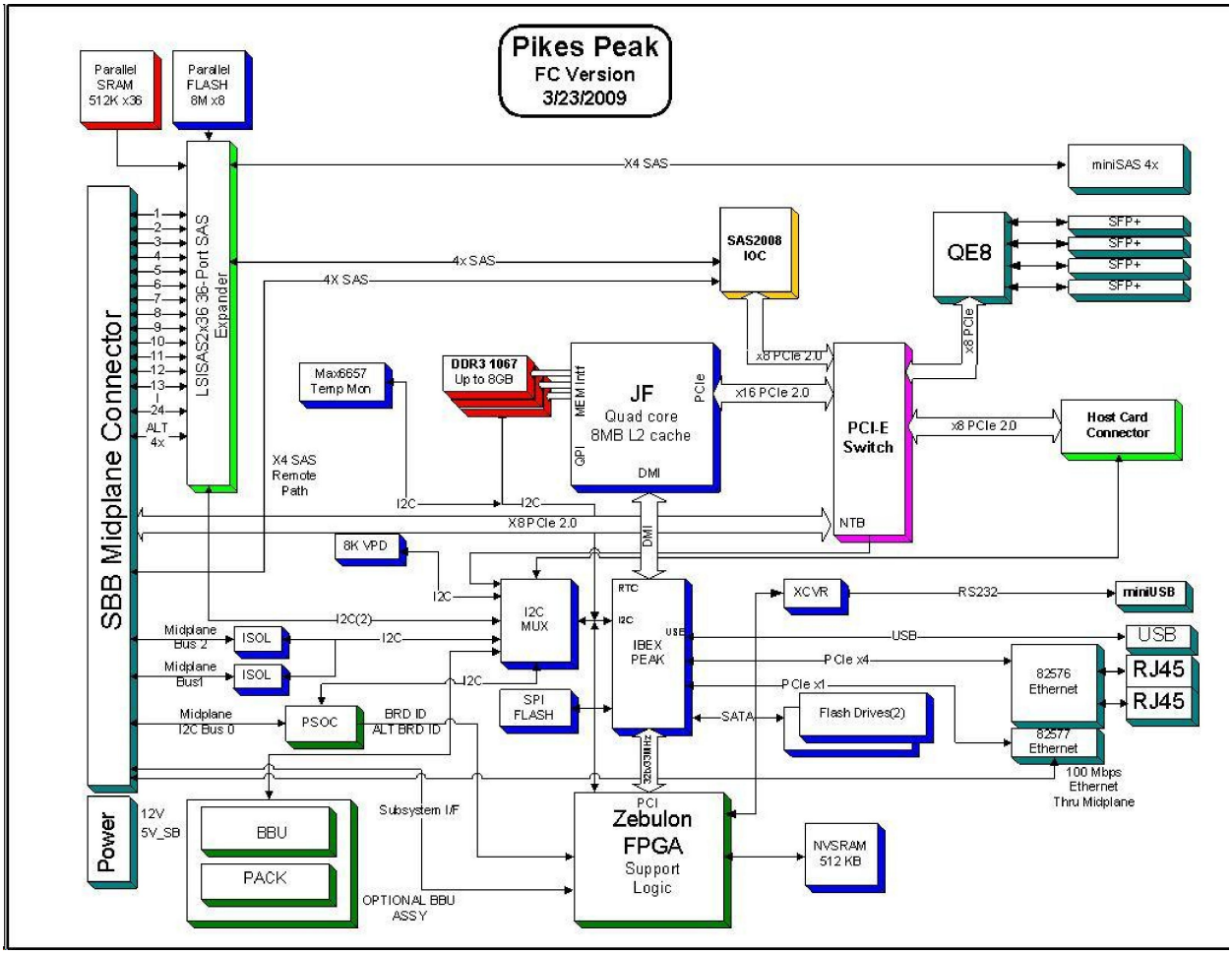
The Pikes Peak RAID Controller core architecture consists of a multi-core Intel Jasper Forest (JF) Processor. The Jasper Forest Processor provides connection up to three Double Data Rate 3 Synchronous DRAM (DDR3 VLP SDRAM) memory DIMMs operating at up to 1067 MT/s. The Jasper Forest Processor also provides an x16 PCI Express Gen2 bus connection to a PCI Express Switch. Also connected to the PCI Express switch is a host board connector, a Tachyon QE8, the SBB midplane, and an LSISAS2008.

The Tachyon QE8 provides on-board Fibre Channel host connectivity. The SBB Midplane connection provides a high speed data path to an alternate controller for data mirroring. The LSISAS2008 chip provides a connection to the backend SAS expander chip and to alternate controller. An LSISAS2x36 Expander chip is utilized for the backend disk connections. The Pikes Peak controller is inserted into trays that contain internal drives; the number of drives is dependent upon the tray type and can support up to 60 disk drives. The Camden, Ebbets and SAS Wembley enclosures support Pikes Peak controller.

The Jasper Forest also connects to an Intel Ibex Peak platform controller hub (PCH). The main devices connected to the Ibex Peak are an Intel 82576, an Intel 82577, SATA Flash drives, and the Zebulon FPGA. The Intel 82576 provides two 1 Gb Ethernet management ports. The Intel 82577 provides a 100 Mb Ethernet connection to an alternate controller through the midplane. The SATA Flash drive(s) provide storage for a persistent cache offload when a power loss occurs. In addition to this, SATA flash drive(s) also provide storage for code, logs, configuration etc.

The Zebulon FPGA provides support logic for the Pikes Peak. Flexibility in host interface configurations is provided using replaceable host interface boards. The Pikes Peak Controller does not operate in simplex mode.

Figure 1: Pikes Peak SBB 2.0 Controller



The Pikes Peak Controller is offered in two versions to cover different price points and performance requirements. The two versions are referred to as PP400 and PP200. The PP400 is the high end controller and the PP200 is the low end controller.

The Pikes Peak controller has the following major components:

- Processor: The PP400 has SKU-4 of the Jasper Forest processor, which is a quad-core 2.13GHz Nehalem based processor with SMP. The PP200 has SKU-10 of the Jasper Forest processor, which is a dual-core 1.73GHz Nehalem based processor with SMP, supports Hyper-threading with VT capability.
- Host Ports on baseboard: Four on-board host side FC ports provided by Tachyon QE8.
- Host Interface Card (HIC): One HIC per controller. HIC can be of the following types:
  1. Dual-port 40Gbps InfiniBand Host Interface (Elk Park)
  2. Dual-port 10 GbE (iSCSI-SFP+ optical) iSCSI Host Interface (Glen Cove)
  3. Quad-port 2/4/8 Gbps FC Host Interface (Manitou)
- DIMM: Up to 24 GB of 64-bit DDR3 SDRAM. The PP400 supports three DIMMs of either 4 GB or 8 GB. The PP200 supports two DIMMs of either 2 GB or 4 GB. The DIMMs on a controller must have to be of same capacity.
- Local Disk Drives: Pikes Peak supports up to 60 local SAS disk drives.
- Drive Channels: The Pikes Peak controller has 1 drive side channel which can be connected to

expansion trays.
- miniUSB: The Pikes Peak controller has 1 miniUSB connector which can be used during manufacturing or extreme-field debugging situations. It carries RS-232 connections.
- USB 2.0: The USB port on controller is used for manufacturing purpose. In a production environment, the USB port is unsupported and a device plugged into USB port is not recognized.
- The Pikes Peak controller has 1 USB connector which can be used during manufacturing or extreme-field debugging situations. This port can also be used to plug-in bootable USB drive. The USB device is not reported in object graph.
- SPI Flash: The SPI flash is of capacity 8 MB and connected to Ibex Peak SPI interface. This flash is having BIOS code for initial boot-up.
- NVSRAM: The Pikes Peak controller has NVSRAM of capacity 512 KB. It is behind Zebulon FPGA.
- SATA flash drives: The PP200 and PP400 have two SATA flash drives of capacity 8 GB for a total of 16 GB.
- FPGA: It is called Zebulon. It has three PCI functions. The function F0 and F2 map general purpose registers and NVSRAM. The function F1 maps UART. The function F2 is having duplicate mapping of general purpose registers required to route specific interrupts to appropriate virtual machine, i.e. Domain0 on Pikes Peak.
- Ethernet Ports: The Pikes Peak controller has two 1G external Ethernet ports for management purpose. The Ethernet port supports Auto MDIX, a mechanism by which NIC automatically detects the cable type (crossover v/s conventional) and handle. The Pikes Peak controller also has one 100 mbps internal Ethernet port to connect to alternate controller via midplane.
- The Pikes Peak controller has various LEDs managed by controller firmware:
  1. Cache Active LED
  2. Service Action Allowed LED (blue)
  3. Service Action Required LED (amber)
  4. BBU charging LED (green)
  5. BBU service action LED (amber)
- Two seven-segment displays.
- Smart BBU. The gas gauge type is bq20z70.
- RAID 5 XOR, RAID6 (P+Q) hardware acceleration.
- A pushbutton switch to reset the IP address or setup initial configuration. At boot time, pressing this button changes the boot order preference.
- PCIE to PCIE NTB to support memory mirroring between controllers.

The Pikes Peak controller will be installed in SBB 2.0 compliant Camden, Ebbets or Wembley SAS enclosures.

1.1.2.Assumptions

- It is assumed that simplex mode is not required to be supported.
- The document 44540-00, Pikes Peak controller – IOVM specific changes, is prerequisite to this document.

1.1.3.Related Documents

Document Number	Name of Document
349-1051890	SYMbol Specification – Internal Master
37021-00	Field Replaceable Host Interface Card FAM
41692-00	XBB-2 FRU Failure Reporting

Document Number	Name of Document
41598-00	XBB-2 Controller FRU Diagnostics
44327-00	Pikes Peak FPGA Management
	Hypervisor AAD
42286-00	Camden Ebbets Enclosure Support FAM
44338-00	Wembley SAS Enclosure FFD
45787-00	Wembley SAS Enclosure FAM
42061-00	Zebulon FPGA - Hardware Functional Specifications
42047-00	Pikes Peak RAID Controller - Hardware Functional Specifications
41115-00	Pikes Peak - Architectural Requirement Document (ARD)
	OSA CAS (Open Storage Architecture Concept Architecture Specifications)
	USP PAS (Unified Storage Platform Product Architecture Specifications)
349-1057240	Component Location Reporting FFD
349-1056330	Controller Management FFD
349-1053460	Controller Type Mismatch FFD
349-1057430	Persistent Cache Backup FFD
349-1063100	Field Replaceable Cache Memory FFD
349-1053340	Fan and Thermal Sensor Management FFD
46722-00	Orion SOD and Component Architecture AAD

1.1.4.Open Issues

- NTB specific details to be covered.
- ECC/PCIe error handling requirements are clear, but detailed architecture is still open due to POCs around it.
- Wake On LAN requirements and detailed architecture.
- Default value of Smart Battery Parameters is TBD.
- Define different seven-segment codes in BIOS for various boots such as PXE, USB, SATA flash etc?
- LSI specific signature to differentiate LSI specific USB boot drive and SATA flash drive. Corresponding checking in BIOS needs to be done to lockout USB, SATA flash drive not having LSI specific signature. In other words, BIOS must ensure secure boot from devices such as USB, SATA flash, and PXE. It should stop booting if it does not recognize LSI firmware image.
- Do we need to support 1G mode in Glen Cove?
- AAD does not cover NAS (CIFS/NFS) functionality.

- CPU temperature threshold and fan failure - It seems XBB2 has this behavior. This is something when a fan fails in a normal running controller. Is it something beyond what is specified in this AAD as part of CPU temperature monitoring?
- CPU temperature Threshold # 1 and Threshold # 2.
- There are many open issues related to SATA flash disk requirements and corresponding functionalities. Most of those issues are documented with “Architecture Note” section in this AAD.

1.1.5.AAD References

The intent of this section is to document functionality that is referenced in this AAD that belongs to other AADs.

Functionality	Owned AAD/FFD Element	Referenced in this AAD
Boot Process, especially PXE	Serviceability AAD - PXE ER	Push Button Support Section
Dual Controller Swap	Serviceability AAD - ACS and Controller Sparing ER	Dual Controller Swap
Lockdown behavior due to HIC and baseboard mismatch	Serviceability AAD - Controller State Management	Start-of-Day Mismatch Detection
IB HIC LED indicators	Controller Module Indicators FFD	InfiniBand Host Interface Card (Elk Park)
SATA Flash disk management and replacement	Serviceability AAD - System Diagnostics ER	Disk Management and Replacement
Power Supply SBB validation	SBB2 VPD Data Support FFD	Power Supply SBB validation and PMBus support
Cache Backup Device Diagnostics	Serviceability AAD - System Diagnostics ER	Cache backup device diagnostics

1.2.Aspect High Level Requirements

1.2.1.Product Requirements

This document covers the element requirements and detailed architecture for following PRs:

ClearQuest PR Number	Feature Name
LSIP200009298	Orion: Support 40Gb IB HIC on Pikes Peak SBB
LSIP200009299	Orion: Support 8Gb FC HIC on Pikes Peak SBB
LSIP200011980	Orion: Pikes Peak SAS Expander checks power supply at power up
LSIP200012568	Orion: Management Ethernet ports should use Auto-MDIX
LSIP200012572	Orion: SATA Flash disk requirements

ClearQuest PR Number	Feature Name
LSIP200012575	Orion: Support LSI's PMBus implementation for power supply management
LSIP200012577	Orion Use Linux's support for ECC/PCI error detection
LSIP200038257	Orion: Wake On LAN Support WOL for Pikes Peak

In addition to above PRs, this AAD also describes the controller type mismatch behavior on Pikes Peak.

1.2.2.AAD Overview

This AAD discusses (or plan to discuss) following major topics:

- Pikes Peak SBB hardware,
- The host interface cards supported
- SBB validation
- Controller type mismatch handling
- SATA Flash disk - Layout, SATA flash disk management, and disk replacement
- Ethernet Auto MDIX support
- ECC/PCIe error handling in OSA environment
- Wake On LAN support on Pikes Peak
- NTB support on Pikes Peak
- DIMM configuration verification
- CPU temperature monitoring
- Controller and Host Board ID computation and necessary infrastructure

2. Pikes Peak Controller - Element Requirements

2.1. Pikes Peak hardware platform - Element Requirements

2.1.1.Introduction

This document provides the following controller specific information:

- Supported hardware configurations.
- Supported memory capacities.
- Handling field replaceable memory and persistent cache backup devices.
- Supported IP address Reset Push button.

2.1.2.Operational Behavior

The Pikes Peak controller is a SBB 2.0 compliant controller supporting heterogeneous host connections such as iSCSI, FC, and InfiniBand (IB). It also supports drive side connections to SAS 2.0 disk drives. Pikes Peak base controller has four on-board Fibre Channel (FC) ports for host connectivity. Each host port supports data rate of 2, 4, or 8 Gb/s and the connectivity to the host ports is provided through SFP+ interfaces. In addition to the FC ports, base board also has a slot to install an additional Host Interface Card (HIC). The Pikes Peak controller supports FC, IB, and iSCSI host connections through optional HICs. For a supported configuration HICs on both the controllers in an array must be of same type.

The Pikes Peak Controller is offered in two versions to cover different price points and performance requirements. The two versions are referred to as the PP400 and the PP200. The PP400 is the high end controller and the PP200 is the low end controller. Both the versions support existing tiered performance behavior. The PP200 and PP400 do not support simplex mode.

2.1.2.1. Host Interface Characteristics

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDACCP5B

Table 1: Pikes Peak Host Interface (On base board) Characteristics

Parameter	Characteristic/Value	Reference
Host Interface	1 host interface card slot and four 2/4/8 Gb/s FC connections on the base controller	1
References: Document Number 349-1057870, the Feature Function Definition for Controller Host Interface Card Management		

2.1.2.2. Drive Interface Characteristics

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDACKP5B

Table 2: Pikes Peak Drive Interface Characteristics

Parameter	Characteristic/Value	Reference
Interface Type	SAS	1
Supported Link Rates (Gb/s)	6 Gb/s	1
Number of Drive Channels	1	
Drive Channel Port Location	Controller	
Drive Channel Expansion Ports per Controller	1 (x4 port via integrated Expander)	
References: 1. Document Number 349-1048660, the Feature Function Definition for SAS Drive Interface		

2.1.2.3. General Controller Characteristics

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDA5UP5B

Table 3: Pikes Peak General Controller Characteristics

Parameter	Characteristic/Value	Reference
Memory System	Three unified processor/cache memory DIMM on PP400 Two unified processor/cache memory DIMM on PP200	
Cache Persistence	Persistent cache backup	1, 11
Cache Backup Device	SATA flash drive (2x8G on PP200 and PP400)	
Cache Backup Power Source	Batteries	11
Cache Backup Power Coordination between controllers	Not Required	11
Dedicated Mirror Channels	Yes (NTB based - TBD)	
Ethernet Management Ports	Two 10/100/1000 external ports, One 10/100 internal port	2, 3
Seven-Segment Display	Yes	4
Seven-Segment Display Sequences	Supported	4
Controller Enclosure Tray ID	Soft-set (Displayed on seven-	5

Parameter	Characteristic/Value	Reference
	segment display)	
Audible Alarm on Controller Module	No	
Performance Tier	Supported	7
Board Sub-Model Identifier	Supported	7
Field Replaceable Cache Memory	Yes (Shared memory)	9
Processor/Cache DIMM Lockdown Recovery	Manual	
Discrete Line Test Capability	No	10
CPU Die Temp Sensor	Yes (CPU temperature monitoring circuitry available)	
Serial Port	miniUSB	
USB Port	Yes	
Supported Enclosures	Camden, Ebbets, Wembley SAS	12, 13
IP Address Reset Push Button Support	Yes	
Auto MDIX support	Yes	
Wake On LAN (WOL) support	Yes (TBD)	

References:

1. Document Number 349-1056620, the Feature Function Definition for Cache Management
2. Document Number 27013-00, the Feature Function Definition for Gigabit Ethernet Management Port
3. Document Number 27806-00, the Feature Function Definition for Ethernet Management Port IPv4 Configuration
4. Document Number 34533-00, the Feature Function Definition for Controller Module Indicators
5. Document Number 349-1048590, the Feature Function Definition for Tray Identifier Management
6. Document Number 349-1053220, the Feature Function Definition for Audible Alarm Management
7. Document Number 349-1049530, the Feature Function Definition for Tiered Performance Management.
8. Document Number 349-1053830, the Feature Function Definition for Board Sub-Model Identifier
9. Document Number 349-1063100, the Feature Function Definition for Field Replaceable Cache

Parameter	Characteristic/Value	Reference
Memory		
10. Document Number 349-1056330, the Feature Function Definition for Controller Management		
11. Document Number 349-1057430, the Feature Function Definition for Persistent Cache Backup		
12. Document Number 42197-00, the Feature Function Definition for DE1600/DE5600 Enclosure Support		
13. Document Number 44338-00, the Feature Function Definition for DE6600 Enclosure Support		

2.1.2.4. Controller Identifiers

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAXEAIC

Table 4: Pikes Peak Controller Identifiers and Model Names

Model Name	Base Board ID (Controller ID)	Host Card ID	Description	Host Ports
5280	5268	None	PP200 baseboard with no HIC	Four 2/4/8 Gb/s FC
5288	5268	0801	PP200 baseboard with FC HIC	Eight 2/4/8 Gb/s FC
5281	5268	0101	PP200 baseboard with iSCSI-SFP+ HIC	Four 2/4/8 Gb/s FC and two 10 Gb/s iSCSI
5284	5268	0401	PP200 baseboard with IB HIC	Four 2/4/8 Gb/s FC and two 40 Gb/s IB
5480	5468	None	PP400 baseboard with no HIC	Four 2/4/8 Gb/s FC
5488	5468	0801	PP400 baseboard with FC HIC	Eight 2/4/8 Gb/s FC
5481	5468	0101	PP400 baseboard with iSCSI-SFP+ HIC	Four 2/4/8 Gb/s FC and two 10 Gb/s iSCSI
5484	5468	0401	PP400 baseboard with IB HIC	Four 2/4/8 Gb/s FC and two 40 Gb/s IB

2.1.2.5. Battery Management

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDA5YAIC

Parameter	Characteristic/Value
Smart Battery	Yes

Parameter	Characteristic/Value
Battery Identity Tracked	Yes
Battery Location	Controller Canister
Battery Removal Method	Remove controller in order to remove battery
Battery Replacement Method	Replace Self
Battery Packaging	Dual individual CRUs
Total Battery packs	2 (One per controller)
Default Expiration Age (90-day units)	13 units (1170 days or 3.21 years)

2.1.2.6. Smart Battery Parameters

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAZCBIC

Reference: Document Number 349-1057690, the Feature Function Definition for Smart Battery Management.

Smart battery configuration parameters are described in the reference. If a parameter is set to zero, the default value is used. The units and default values are shown in the following tables:

Table 5: Pikes Peak Controller Smart Battery Configuration

Parameter Name	Units	Default
Minimum Application Capacity	Seconds	80 (TBD)
Replacement (Full Charge) Capacity	Seconds	313 (TBD)
Learn Cycle Interval	Weeks	8 (TBD)

Table 6: Pikes Peak Controller Smart Battery Attributes

Attribute	Value
Cache during battery learn cycle	TRUE
Learn cycle terminated if critical BBU temperature threshold reached	FALSE
Charging disabled if critical BBU temperature threshold reached	FALSE

Table 7: Pikes Peak Controller Smart Battery Temperature Logging Ranges (deg C)

Minimum Temperature	Maximum Temperature
No minimum	35

Minimum Temperature	Maximum Temperature
36	40
41	45
46	50
51	55
56	60
61	65
66	70
71	No maximum

2.1.2.7. Seven Segment Display Support

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAO1BIC

Reference: Document Number 34533-00, the Feature Function Definition for Controller Module Indicators.

Pikes Peak controllers display category-detail sequences on a seven-segment display as described in the reference. The seven-segment display is used to identify controller component failures as indicated in table below. Component failures may also be identified through event notification mechanisms when the controller is in the operational state.

Table 8: Pikes Peak Component Failure Identification

Component (Code)	Seven-Segment Display used in this controller state		
	Power-on diagnostics	Suspended	Operational
Shared DIMM (Dx+)	Yes (Only on Cache portion)	Yes	No
Host Card (Hx+)	Yes	Yes	No
Flash Drive (Fx+)	Yes	No	Yes
Battery	No	No	No
Power Supply (Sb+)	Yes	Yes	No

Architecture Note: Do we perform Power-on diagnostics on Power Supplies on existing platforms? If so, does controller currently display 7-segment code for Power Supply power-on diagnostics failure on existing platforms?

2.1.2.8. Protection Information Capability

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDARICIC

Reference: Document Number 34472-00, the Feature Function Definition for End-to-End Data Protection Information (PI).

Table 9: Pikes Peak Protection Information Capability

Base Board ID/Host Board ID	Description	PI Capable
5268, 5468	Base Pikes Peak Controller with 4 FC host ports	Yes
0801	4-port 2/4/8 Gb/s FC HIC	Yes
0101	2-port 10 Gb/s iSCSI-SFP+ HIC	Yes
0401	2-port 40 Gb/s IB HIC	Yes

2.1.2.9. Field Replaceable Cache Memory

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAESCIC

All DIMM slots in Pikes Peak controller shall have the same capacity DIMMs installed:

- PP200: Two DDR3-1067 DIMMs of capacity 2GB with a total capacity of 4GB
- PP200: Two DDR3-1067 DIMMs of capacity 4GB with a total capacity of 8GB
- PP400: Two DDR3-1067 DIMMs of capacity 4GB with a total capacity of 12GB
- PP400: Two DDR3-1067 DIMMs of capacity 8 GB with a total capacity of 24GB

The PP200 has 2 DIMM slots whereas PP400 has 3 DIMM slots. It is required that controllers have DIMMs of same capacity installed in available DIMM slots. DIMMs of different capacity on a controller are treated as unsupported. The unsupported behavior is same as described in document # 349-1063100, Field Replaceable Cache Memory Feature Function Definition.

2.1.2.10. Field Replaceable Cache Backup Device

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDARVCIC

The PP200 and PP400 have two cache backup devices, each of capacity 8 GB, populated in slot 0 and slot 1. Any configurations outside of these are considered invalid.

The controller has following behavior in case of invalid cache backup device configurations:

- The two controllers switch all volumes configured for write-back caching with mirroring to write-through caching.
- The controller continues to report all the installed cache backup devices.
- The controller logs a critical event and raises needs condition to add/remove the extra cache backup device.

Refer to SATA Flash Disk Element Requirement document for detailed information.

2.1.2.11. Field Replaceable HICs

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAPYCIC

The HICs on Pikes Peak controllers are field replaceable but HICs on both the controllers must match for the supported configuration.

Refer to document # 349-1057870, the Feature Function Definition for Controller Host Interface Card Management.

2.1.2.12. Push Button Support

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAWZCIC

In controllers prior to Pikes Peak, the serial port could be used to set the controller's management port IP addresses. This is useful if the IP address is unknown, and the controller is inaccessible via network. The Pikes Peak controller does not support traditional Serial Port Recovery Interface (SPRI) and instead contains a hardware push button. Pressing the pushbutton immediately resets the external Ethernet ports IP configuration to the "Quick Start" IP configuration. This allows the management application to communicate with the platform and perform further configuration. Refer to document # 44340-00, Hypervisor AAD - Virtualized hardware resources for detailed information.

In addition to IP address reset, push button when pressed continuously for 5 seconds, during reboot, allows controller to boot in a pre-defined order as follows:

1. Preboot eXecution Environment (PXE) - This option requires a Trivial File Transfer Protocol (TFTP) server running somewhere in the network with required boot image.
2. Boot from the installed SATA flash disks - This is the default mode and does not require push button to be pressed.

Architecture Note: Internal NIC IP address configuration is not affected by Push button.

Architecture Note: Boot order is still in an early stage and may change as we get more information. I have removed booting from USB option as we want to keep it as a hidden option.

Architecture Note: PXE boot is not committed for stage 1 yet.

2.1.2.13. USB Port

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAN3CIC

The USB port on controller is unsupported and a device plugged into USB port is not recognized.

Architecture Note: It is possible, though not desired, to PXE boot via USB if no tftp servers are found on the network. This port can also be used to boot the controller from LSI provided bootable USB drive, but this option is hidden and should be used by LSI designated support personnel. The USB port must be disabled by firmware in Domain0.

2.1.2.14. miniUSB Port

Topic ID: 2010-03-11T16:21:00Z-2258-12872-IDANZSMF

The Pikes Peak controller has one mini-USB style connector providing an external RS232 serial connection. The behavior of this port is same as serial port on platforms prior to Pikes Peak.

2.1.2.15. Power Supply

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAOADIC

The power supply on Pikes Peak controller supports PMBus protocol. Refer to document # 42253-00, the Feature Function Definition for SBB 2.0 VPD Data Support, for detailed information.

2.1.2.16. Wake on LAN Support (WOL)

Topic ID: 2009-11-02T15:32:00Z-2419-13791-IDAQBDIC

The Ethernet management ports on Pikes Peak controller support WOL.

Architecture Note: It is yet to be determined whether firmware needs to support WOL in first out.

2.2. Controller Type Mismatch - Element Requirements

2.2.1.Introduction

The Pikes Peak controller can be replaced with the same controller type. There are two versions of Pikes Peak controllers: a). PP200 and b). PP400. These two versions of controllers are treated as two different controller types and results in controller type mismatch if inter-mixed in same enclosure. In addition to this, host interface card on both the controllers must match for a supported configuration.

Architecture Note: Some of the information in the Element Requirement is from existing FFD, but most of the sections have some changes.

2.2.2.Operational Behavior

2.2.2.1. Controller Hot-Swapped with Incorrect Controller Type

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAUPP5B

Let's assume that controller A is inadvertently replaced with the incorrect controller type while controller B is operational. The outcome is as follows:

- Controller A continues to boot either from the functional partition in SATA flash on Pikes Peak or from on-board flash on platforms prior to Pikes Peak.
- Controller A enters the suspended state as a result of board ID mismatch.
- Upon insertion of controller A, the controller B detects the board ID mismatch and generates event notification as described in Section 2.2.4.1. Controller Type Mismatch.

2.2.2.2. Controller Hot-Swapped with same Controller Type

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDASSP5B

Let's assume that controller A is replaced with the same controller type while controller B is operational. The outcome is as follows:

- The controller A continues to boot from the functional partition in SATA flash.
- The controller A performs auto code synchronization if it is running different firmware than the firmware running on incumbent controller. Auto Code Synchronization rules are defined in Serviceability AAD - ACS and Controller Sparing element.
- The controller reboots once the firmware is re-flashed from the incumbent controller.

2.2.2.3. Controller A or B Cold-Swapped with Incorrect Controller Type

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDA20P5B

This scenario occurs when one of the controllers is inadvertently replaced with the incorrect controller type when the enclosure is powered off. The outcome is as follows:

- When power is applied to the enclosure, both the controllers continue to boot either from the functional partition in SATA flash on Pikes Peak or from on-board flash on platforms prior to Pikes Peak.
- Both controllers enter the suspended state due to board ID mismatch.
- The storage array is not capable of performing host I/O operations until one of the controllers is removed.

2.2.2.4. Power Cycle after Hot-Swap with Incorrect Controller Type

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAV3P5B

If the enclosure is power-cycled with mismatched controller types after one of the controllers had previously been hot-swapped with incorrect controller type, the outcome is the same as if a controller had been cold-swapped with the incorrect controller type.

2.2.2.5. Controller-Drive Enclosure Considerations

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAA5P5B

Controller canister slots in controller-drive enclosures are designed to accept either a storage array controller, or an Environmental Services Monitor (ESM). ESMs are used in drive expansion enclosures that are attached to controller and controller-drive enclosures. If a controller is present in one slot and an ESM is present in the other slot, the controller will behave as if the ESM were a controller of a different type. This will result in controller type mismatch.

2.2.2.6. Dual Controller Swap

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAURR5B

The dual controller swap behavior remains same as on previous platforms. Refer to existing FFD for more details.

Architecture Note: This behavior will change once NAS is introduced as cluster configuration needs to be handled. There might be few other things. The details should be handled in Serviceability AAD - ACS and controller sparing aspect element.

2.2.2.7. Start-of-Day Mismatch Detection

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAEVR5B

After each cold boot, the controller attempts to obtain the controller baseboard ID and host interface ID(s) from the alternate controller. If there is a mismatch in any pair of identifiers (base controller-alternate base controller, host card-alternate host card), the booting controller enters a suspended state and performs the following actions:

- The controller performs a self-lockdown and suspends normal operations.
- If the controller has a seven-segment display, a mismatched controller type lock-down code is displayed. The lock-down codes are defined in document # 34533-00, the Feature Function Definition for Controller Module Indicators.
- Management services (via SYMbol, CORBA etc.) are not available.

Architecture Note: Lockdown behavior is defined in serviceability AAD - Controller State Management.

2.2.2.8. Run-Time Mismatch Detection

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAJYR5B

When a controller is hot inserted, the in-place controller attempts to obtain the controller board ID and host interface ID of the hot inserted controller. If there is a mismatch in any pair of identifiers, the in-place controller generates event notification as described in Section 2.2.4.1. Controller Type Mismatch. The status for the newly inserted controller is set to be suspended.

2.2.2.9. Failure to Determine Alternate Board Identifier

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDA1ZR5B

If the alternate board identifiers cannot be determined, the in-place controller does not report the mismatched controller error. Instead, the controller reports:

- A Needs Attention condition.
- A critical MEL event indicating that the alternate board identifier could not be determined.
- A Recovery Guru procedure indicating that the alternate board identifier could not be determined.
- The replaced controller either enters the SBB validation failure state or controller type mismatch state if it is not able to validate the board identifier with the in-place controller. The SBB validation failure is described in document # 42253, the Feature Function Definition for SBB 2.0 VPD Data Support.

2.2.3.Administrative and Configuration Interfaces

2.2.3.1. Controller Mismatch Reporting

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDAD3R5B

Architecture Note: There is no change in existing reporting. Refer to existing FFD.

2.2.4.Error Handling and Event Notification

2.2.4.1. Controller Type Mismatch

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDA54R5B

As indicated in previous sections, a controller that detects a controller type mismatch during cold boot performs a self-lockdown and enters the suspended state. A controller that is suspended due to a type mismatch will automatically leaves the suspended state and resumes normal operations if the alternate controller is removed.

Architecture Note: This is an existing behavior.

2.2.4.2. Device Reported Events

Topic ID: 2009-11-02T15:32:00Z-1780-10151-IDARQPAC

The Pikes Peak controller will continue to have existing MEL events and Recovery Guru.

2.3. Host Interface Cards - Element Requirements

2.3.1.Introduction

The Pikes Peak base controller has four Fibre Channel (FC) ports for host connectivity. Each port supports data rate of 2, 4, or 8 Gb/s and the connectivity to the host ports is provided through SFP+ interfaces. In addition to the FC ports on base controller, Pikes Peak controller also has a provision to support InfiniBand (IB), iSCSI, or additional FC ports through an optional Host Interface Card (HIC). The HIC form factor for Pikes Peak controller is not compatible with HICs developed for platforms prior to Pikes Peak.

2.3.2.Operational Behavior

Following sub-sections describe the behavior for different HICs on Pikes Peak controller.

2.3.2.1. Fibre Channel Host Interface Card (Manitou)

Topic ID: 2009-11-02T15:15:00Z-516-2943-IDAUPBRC

Pikes Peak controller has a provision to attach an additional FC HIC, Manitou. The Manitou card supports four FC ports where each port supports data rate of 2, 4 or 8 Gb/s. With this HIC, there are a total of 8 FC ports available on Pikes Peak. The FC ports only support FCP upper-layer protocol. Only direct and FC fabric connections are supported; FC-AL attached to an FC fabric is not supported.

The LED behavior for the HIC is same as described in document # 34533-00, the Feature Function Definition for Controller Module Indicators.

2.3.2.2. InfiniBand Host Interface Card (Elk Park)

Topic ID: 2009-11-02T15:15:00Z-516-2943-IDA4QBRC

Pikes Peak controller also has a provision to attach an InfiniBand HIC, Elk Park. The Elk Park card supports two IB ports where each port supports data rate up to 40 Gb/s. The IB host card interconnects via an InfiniBand switch fabric. The LED behavior for the HIC is same as described in document # 34533-00, the Feature Function Definition for Controller Module Indicators.

Refer to Feature Function Definition document # 349-1061360 for more information about InfiniBand host port functionality.

Architecture Note: Document # 34533-00, the Feature Function Definition for Controller Module Indicators needs to be updated for IB LED indicators.

2.3.2.3. iSCSI-SFP+ Host Interface Card (Glen Cove)

Topic ID: 2009-11-02T15:15:00Z-516-2943-IDASSBRC

Pikes Peak controller also has a provision to attach an Ethernet/iSCSI HIC, Glen Cove. The Glen Cove card supports two Ethernet ports where each port supports data rate of 10 Gb/s. Each Ethernet port is implemented by separate PCI functions for unaccelerated Ethernet protocol and accelerated iSCSI Initiator/Target protocol. The iSCSI Initiator and iSCSI Target functions may optionally be implemented by separate PCI functions.

The LED behavior for the HIC is same as described in document # 34533-00, the Feature Function Definition for Controller Module Indicators.

2.4. SATA Flash Disk Support - Element Requirements

2.4.1.Introduction

The SATA Flash disk is a new component on Pikes Peak. It has several partitions: storage for functional code, storage for logs, storage for configurations, storage for core files, and storage for cache offload.

The PP200 and PP400 have two SATA flash disks of capacity 8 GB for a total of 16 GB.

Architecture Note: if a single PP type supports more than 1 flash configuration, then it affects functionality associated with upgrade from block only to block + file. in essence, we are trying to ensure that when a user moves from b-o to b+f, we also do not have to worry about flash drive replacements.

2.4.2.Operational Behavior

2.4.2.1. Partitioning

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDALQBRC

Pikes Peak uses Master Boot Record (MBR) partitioning scheme. This is the traditional layout used in PC operating systems. Broadly, Partitions can be categorized into two categories: Raw partitions and the partitions with the file systems

Raw partitions:

- Partition for IOVM Cache Offload
- TBD

File system partitions:

- Flash layout partition - Contains metadata and flash layout
- TBD

A suggested layout is described in Table 1.

Table 10: Cache Backup Device Partition

Index	Partition Use	Mirrored, if dual disk system	Capacity allocated	
			PP200 (2x8 GB total)	PP400 (2x8 GB total)
	Flash Layout	No	128K	128K
	Other partitions	TBD	TBD	TBD
	Cache Offload	No	4 GB	8 GB

Architecture Note: This table will undergo changes in future as things evolve. In this Rev of the document, just concentrate on cache offload partition. In fact depending upon POCs around this area, it is possible that this partition also undergoes changes.

Architecture Note: The flash layout has to be thought through to ensure that we accommodate PP100 with minimal effort which has one flash drive.

2.4.2.2. Default boot partition

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDAWBCRC

The default partition to boot is the Domain0 OS partition. The controller will always boot from bootable image on SATA flash disk in slot-0 (0 relative). There is no plan to boot controller from SATA flash disk in slot 1.

2.4.2.3. Mirroring

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDADDCRC

There is no plan to support mirroring between two SATA flash drives; however, cache offload will continue to work as it does today where it mirrors the metadata on a mirror partition whether on same drive or on a different drive.

2.4.2.4. Disk Configuration Information

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDA0ECRC

Each SATA flash disk drive maintains a layout partition to keep the metadata and the layout information. The metadata information includes some kind of signature to identify valid SATA disk. The layout information can include partition information and offset of various partitions from the start of flash. The information in this partition is called flash disk configuration information.

Part of this configuration information also includes a unique identifier that lets the controller firmware determine the layout of the partitions on this drive. This identifier is created by controller firmware during initialization if the drive does not contain any recognizable information.

Architecture Note: All the VMs dealing with SATA flash disk need to request Domain0, using rmi, for the disk layout and any specific information about metadata. We are yet to identify the complete list of metadata and supporting use-cases.

Architecture Note: Is it possible to insert any off-the-shelf SATA flash on our controller? If yes, then we need BIOS to validate some kind of signature before it boots off of the SATA flash.

2.4.2.5. Disk Management and Replacement

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDAOGCRC

Architecture Note: "Serviceability AAD - System Diagnostics" should handle the SATA flash disk replacement scenarios. If needed, this section can be populated in the future or can refer to serviceability AAD.

2.4.2.6. SATA flash disk configurations

Topic ID: 2010-03-11T16:15:00Z-1218-6945-IDA3NFDF

The PP200 and PP400 have two SATA flash disks, each of capacity 8 GB. Following are valid configurations on PP200 and PP400. Any configurations outside of these are considered invalid.

- PP200 and PP400 have both the slots (slot 0 and slot 1) populated with 8 GB SATA flash disks.

The controller has following behavior in case of invalid SATA flash disk configurations:

- The controller continues to report all the installed SATA flash disks.
- The controller logs a critical MEL event and raises needs condition to correct the configuration.

2.4.3.Administrative and Configuration Interfaces

2.4.3.1. Management Interface

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDAYNCRC

The SATA flash disk reporting will use existing management interface defined for Cache backup device.

Architecture Note: IOVM retrieves the SATA disk INQUIRY data from Domain0, so let IOVM continue to populate cache backup device info into object graph.

Architecture Note: Instead of reusing CacheBackupDevice data structure in SYMbol, do we need to add a new data structure? SATA flash is used for various purposes than just cache backup? Essentially, this new data structure may be a complete duplication with just a name change. Though this new data structure can provide flash layout as well, but is it really useful for end-user?

2.4.4.Error Handling and Event Notification

2.4.4.1. Invalid SATA flash disk configurations

Topic ID: 2009-11-02T15:15:00Z-1064-6067-IDAZPCRC

A new critical event and Recovery guru will be defined to cover the invalid configuration scenarios described in Section 2.4.2.6. SATA flash disk configurations.

Existing MEL events defined in document # 349-1057430, the Feature Function Definition for Persistent Cache Backup, cover rest of the error cases.

2.5. Auto MDIX - Element Requirements

2.5.1.Element Functional Behavior Changes

Existing LSI controllers require a straight through cable for connecting to a switch, and a crossover cable to connect to a PC. An appropriate cable is necessary to connect the transmit pin on one interface to the receive pin on the other interface. With an inappropriate cable, the transmit pin on one interface would be connected to the transmit pin on the other interface, resulting in failed communication.

2.5.2.Introduction

Auto-MDIX (automatic medium-dependent interface crossover) is a technology that automatically detects the required cable connection type (straight-through or crossover). It swaps transmit and receive pins in

the Ethernet controller if necessary to ensure an appropriate connection.

2.5.3.Operational Behavior

2.5.3.1. Ethernet Ports

Topic ID: 2009-11-02T15:14:00Z-269-1536-IDALRHTB

On Pikes Peak, The management and host iSCSI ports which use relevant cabling will use Auto MDIX technology, where supported by hardware. In particular, unshielded twisted pair (UTP) interfaces should use Auto-MDIX. For Auto-MDIX to operate correctly, the speed on the interface and duplex setting must be set to "auto".

2.6. ECC and PCI Error Handling - Element Requirements

2.6.1.Introduction

The controller detects ECC and PCI/PCIe errors and takes corrective actions as described in subsequent sections.

2.6.2.Operational Behavior

2.6.2.1. ECC Error Detection and Recovery

Topic ID: 2009-11-02T15:16:00Z-607-3462-IDAPPBRC

The controller memory is Error Correcting Code (ECC) protected. The ECC allows data that is being read or transmitted to be checked for errors and corrected, if possible, on the fly. The ECC errors can be correctable (single-bit) or uncorrectable (multi-bit). The controller employs following behavior for ECC errors:

- The controller resets when the single-bit ECC error threshold (Currently set to 10) is exceeded or when one multi-bit ECC error is encountered. If one of the thresholds for single-bit ECC errors has been exceeded within a certain time period, the controller enters the Lockdown state.
- The controller scans the cache memory during Start of Day and enters the Lockdown state if there are 100 transient uncorrectable ECC errors or 1 persistent uncorrectable ECC error.

2.6.2.2. Patrol Scrub

Topic ID: 2009-11-02T15:16:00Z-607-3462-IDAVRBRC

The Patrol scrub sequentially reads every memory location on the controller. By reading memory, potentially bad locations can be detected even if in an area infrequently accessed. The concept is similar to disk scrubs in RAID controllers, which can verify that data can be accessed and the parity and data are consistent. It can prevent uncorrectable errors and resultant reboot by detecting a single bit error and correcting before another bit in the field toggles and field become uncorrectable.

Patrol scrub is configured to read every memory location once every day. At this speed, performance impact is negligible. Patrol scrub, by default, is enabled.

Errors detected in patrol scrub are reported identically to other ECC errors as described in section .

Architecture Note: We need to measure the performance impact due to patrol scrub and provide a way to disable if it results in significant performance degradation.

2.6.2.3. PCI/PCIe Error Detection and Recovery

Topic ID: 2009-11-02T15:16:00Z-607-3462-IDAXUBRC

The controller employs following behavior for PCI/PCIe errors:

- The controller resets for any PCI/PCIe errors except the PCI/PCIe errors caused by master abort.
- The controller performs error threshold (currently set to 3) check during SOD and Lockdown if error count exceeds the threshold.

Architecture Note: Verify

2.6.3.Error Handling and Event Notification

2.6.3.1. ECC Error

Topic ID: 2009-11-02T15:16:00Z-607-3462-IDAZXBRC

A critical MEL event is logged when one of the thresholds for single-bit or multi-bit ECC errors in the memory exceeds within a certain time period. The controller also displays a code on seven-segment display. Refer to document Number 34533-00, the Feature Function Definition for Controller Module Indicators for the seven-segment code.

2.6.3.2. PCI/PCIe Error

Topic ID: 2009-11-02T15:16:00Z-607-3462-IDA4YBRC

A critical MEL event is logged when one of the thresholds for single-bit or multi-bit ECC errors in the memory exceeds within a certain time period. The controller also displays a code on seven-segment display. Refer to document Number 34533-00, the Feature Function Definition for Controller Module Indicators for the seven-segment code.

Architecture Note: Verify.

2.7. Wake on LAN Support - Element Requirements

2.7.1.Introduction

TBD

3. Pikes Peak hardware platform - Detailed Architecture

3.1. Pikes Peak hardware platform - Detailed Architecture

3.1.1.High Level Design

Current hardware platforms run RAID controller firmware on bare metal hardware having VxWorks as operating system. The Pikes Peak controller is the first controller platform based on Open Source Architecture (OSA). The Pikes Peak controller runs multiple VMs (Virtual Machines) on a hypervisor. One of the VMs runs existing VxWorks based RAID firmware. This VM is referred as IOVM (IO Virtual Machine) in rest of this document.

3.1.1.1.Details

3.1.1.1.1.Local Controller Baseboard ID

The PP200 and PP400 have same baseboard ID in hardware; however, they cannot inter-operate in same enclosure having PP200 in one slot and PP400 in other slot. This requires controller firmware to construct a unique baseboard ID, based on some unique information. The PP200 has dual core CPU whereas PP400 has quad core CPU. The CPU information along with the hardware baseboard ID (Retrieved from FPGA via discrete line) is used to construct the unique baseboard ID in firmware. The controller firmware must use this constructed baseboard ID on Pike Peak as opposed to hardware baseboard ID on existing platforms. The constructed baseboard ID for PP200 and PP400 is 5268 and 5468 respectively.

The [IOVM] brdm (Board Manager) component is responsible for constructing the local baseboard ID. This component provides an interface to return the constructed baseboard ID. The [IOVM] brdm component uses BCM interface to get the baseboard ID from hardware and issues CPUID instruction (or equivalent) to processor to retrieve the basic and extended processor information. The [IOVM] brdm component then constructs the unique baseboard ID based on the information retrieved.

The [IOVM] brdm component needs to provide two interfaces: One to return the local controller's baseboard ID directly from PSOC (let's call this "hardware baseboard ID" from now on) and the other to return the constructed baseboard ID (let's call this "constructed baseboard ID" from now on) for the local controller. The [IOVM] hcvh component will transition from [IOVM] BCM to [IOVM] brdm component to get the hardware baseboard ID for SBB validation. Refer to later sections for more information about SBB validation.

3.1.1.1.2.Alternate Controller Baseboard ID

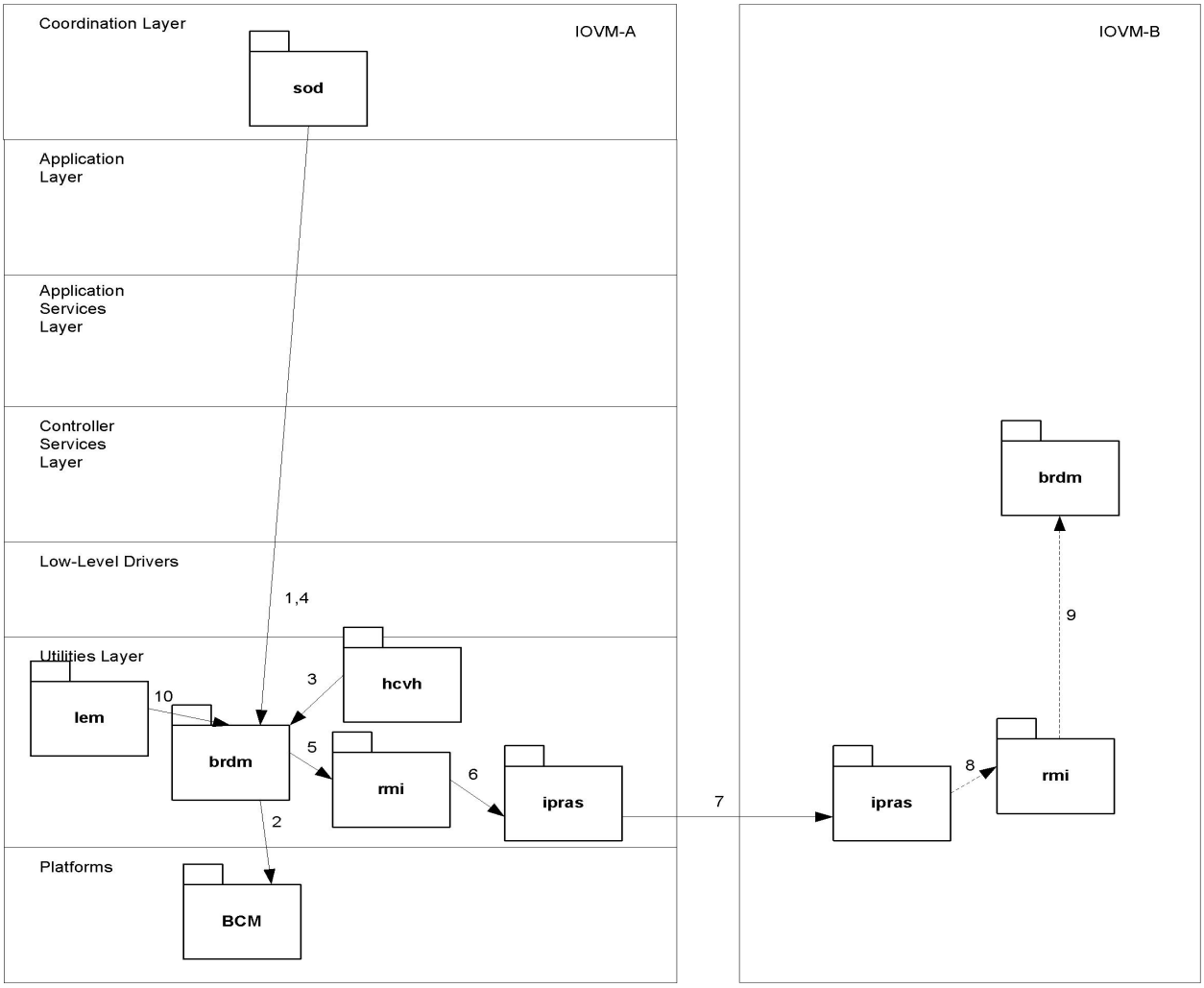
On existing platforms, controller firmware retrieves the alternate's hardware baseboard ID; however, on Pikes Peak, this information cannot be used in isolation (due to same hardware baseboard ID for both PP200 and PP400) as described in previous section. As a result of this, the controller firmware needs to communicate with alternate controller, using internal NIC, to retrieve alternate controller's constructed unique baseboard ID. During initialize phase, the [IOVM-A] brdm component calls [IOVM-B] brdm component, and vice-versa, to get the alternate's constructed baseboard ID.

The [IOVM] brdm component needs to provide two interfaces: One to return the alternate controller's hardware baseboard ID and the other to return the alternate’s constructed baseboard ID. The [IOVM] hcvh component will transition from [IOVM] BCM to [IOVM] brdm to get the alternate controller's hardware

baseboard ID for SBB validation. Refer to later sections for more information about SBB validation.

Architecture Note: We want to avoid using internal NIC until [IOVM] hcvh performs SBB validation in [IOVM] hcvh::instantiate phase. Hence, hcvh will use the local and alternate controller's hardware baseboard ID for SBB validation purpose.

Figure 2: Retrieve Alternate controller's Baseboard ID



The following are the call sequences:

1. [IOVM] sod instantiates brdm component.
2. [IOVM] brdm component constructs the local baseboard ID using hardware baseboard ID and processor information. The brdm component also gets the alternate controller's hardware baseboard ID. Note that hardware baseboard ID is same for both PP200 and PP400.
3. [IOVM] hcvh gets the local and alternate controller's hardware baseboard ID from brdm for SBB validation.
4. [IOVM] sod initializes brdm component.
5. [IOVM] brdm calls rmi.
6. [IOVM] rmi calls ipras.

7. [IOVM-A] ipras calls [IOVM-B] ipras.
8. [IOVM-B] ipras calls rmi.
9. [IOVM-B] rmi calls brdm component. The brdm component returns the constructed baseboard ID.
10. [IOVM-A] lem::initialize calls brdm to get the local and alternate controller's constructed baseboard to perform the board ID mismatch check.

Note that the above scenario works when the alternate controller is already up and running or at least has passed brdm::instantiation.

3.1.1.1.2.1.Error getting alternate controller's constructed baseboard ID

It is possible that either alternate controller is not up or there is an issue with internal NIC. To handle such scenarios, after SBB validation, the [Domain0] vmmgr will transmit a packet (Kind of keep alive packet) to alternate controller over the internal NIC and wait for 30 seconds (arbitrary number to start with) for a response from alternate. During this time, the IOVM will block on a semaphore. After the successful transmission of keep alive packet or expiry of the wait period, the [Domain0] vmmgr will notify IOVM via xenstore. The IOVM will then resume the SOD. The communication to alternate timeout may result in alternate board ID error during lem::initialize (unless the NIC is up after the timeout but before the lem::initialize). This will result in controller type mismatch scenario. Refer to subsequent sections for detailed information about controller type mismatch.

Note: The [Domain0] vmmgr will use ALT_INPLACE discrete line to find out whether alternate is physically present to send a keep alive packet.

3.1.1.1.3.Host Board ID

On existing platforms, the FPGA has information about host board IDs of HICs installed on local and alternate controller. However, on Pikes Peak, FPGA will only provide information about host board ID of local HIC. As a result, the controller will communicate with alternate, using internal NIC, to retrieve the host board ID of HIC(s) installed on the alternate. The [IOVM] brdm component will retrieve the host board ID from alternate controller during its initialize phase. The brdm component can call existing BCM interface to get the Host board ID for the local HIC. The error in retrieving the host board ID of HIC installed on alternate controller will result in controller type mismatch. Refer to subsequent sections for detailed information about controller type mismatch.

Note: The HIC mismatch results into deferred lock down until ACS. There is no change in this behavior.

3.1.1.1.4.SBB Validation

The IOVM will continue to perform SBB validation on Pikes Peak. During SOD, the Domain0 will launch IOVM as one of the very first item and will block until IOVM completes SBB validation or a timeout occurs. The [IOVM] hcvh component continues to perform following validations as part of SBB validation (New items are highlighted in bold):

1. Enclosure Validation
2. Enclosure/Canister match
3. Vent Configuration
4. **Vendor Pairing**- This is an additional step to the existing [IOVM] hcvh SBB validation functionality. As part of this, the [IOVM] hcvh component will validate the hardware baseboard ID of local and alternate controller. Mismatch will result in SBB validation failure.
5. Power Supply (PS) Validation

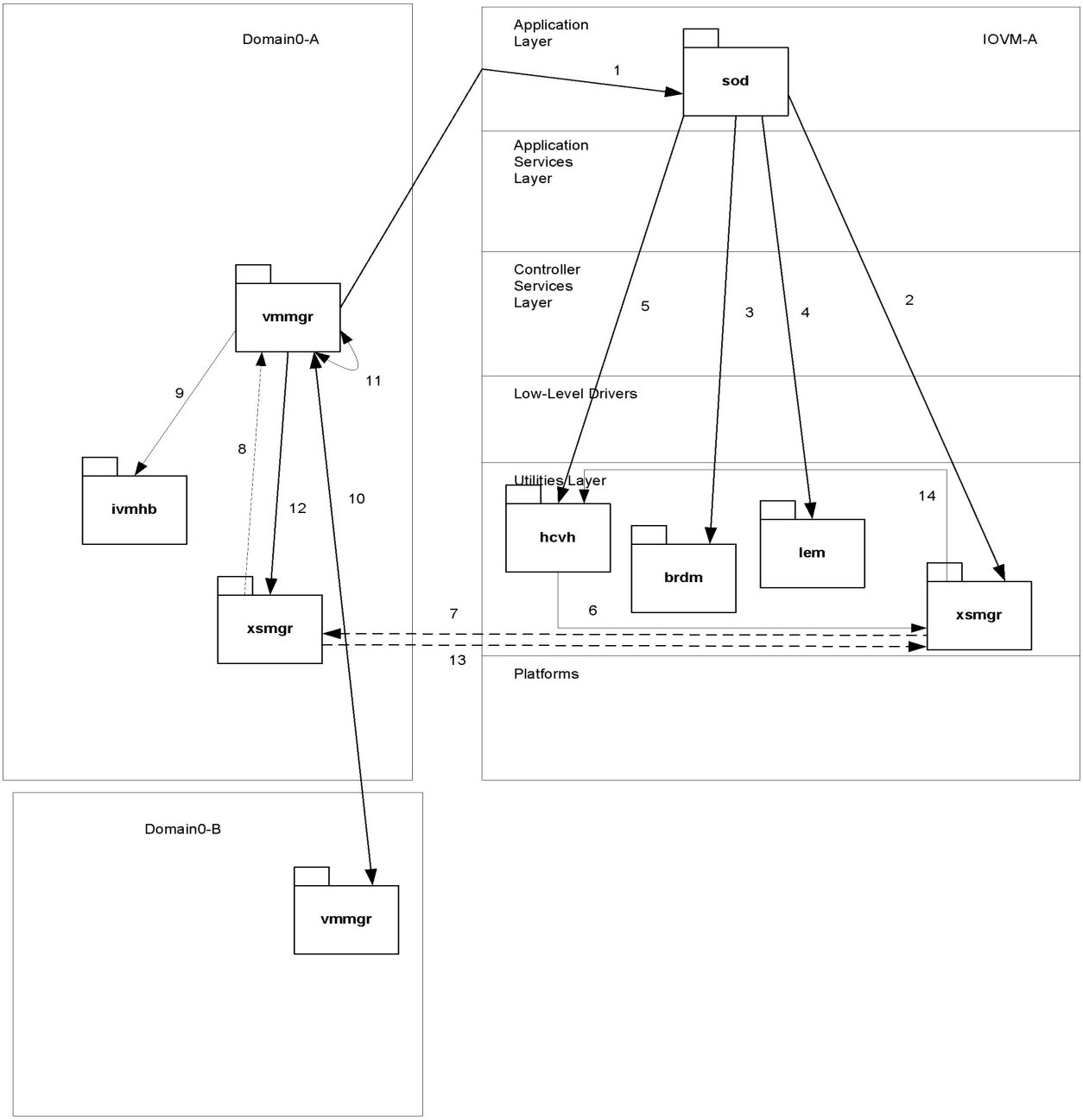
It is to be noted that there is a plan to perform first two steps (Enclosure and Enclosure/Canister validation) of SBB validation in SAS Expander firmware. After performing the SBB validation, the SAS Expander firmware will release the rest of the board. The quick switches will be enabled by the FPGA

when expander firmware releases the processor from reset after SBB validation. The enabling of quick switches will enable low-speed signals between the controllers over the mid-plane. It is to be noted that to perform PS SBB validation on Pikes Peak, quick switches must be enabled few seconds prior to the validation. The enabling of quick switches allows PS in-place line to stabilize before controller performs SBB validation. Having FPGA enabling quick switches before releasing the board provides sufficient window to stabilize PS in-place signal and allows controller firmware to validate the PS without any additional delay. As a result of this, the [IOVM] hcvh does not need to enable quick switches as done on Snowmass. The quick switches are enabled by the FPGA while releasing the controller board.

Refer to Enclosure specific FFD/FAMs for details about SBB validation.

3.1.1.1.4.1.SBB Validation Success

Figure 3: SBB Validation success sequence relative to SOD



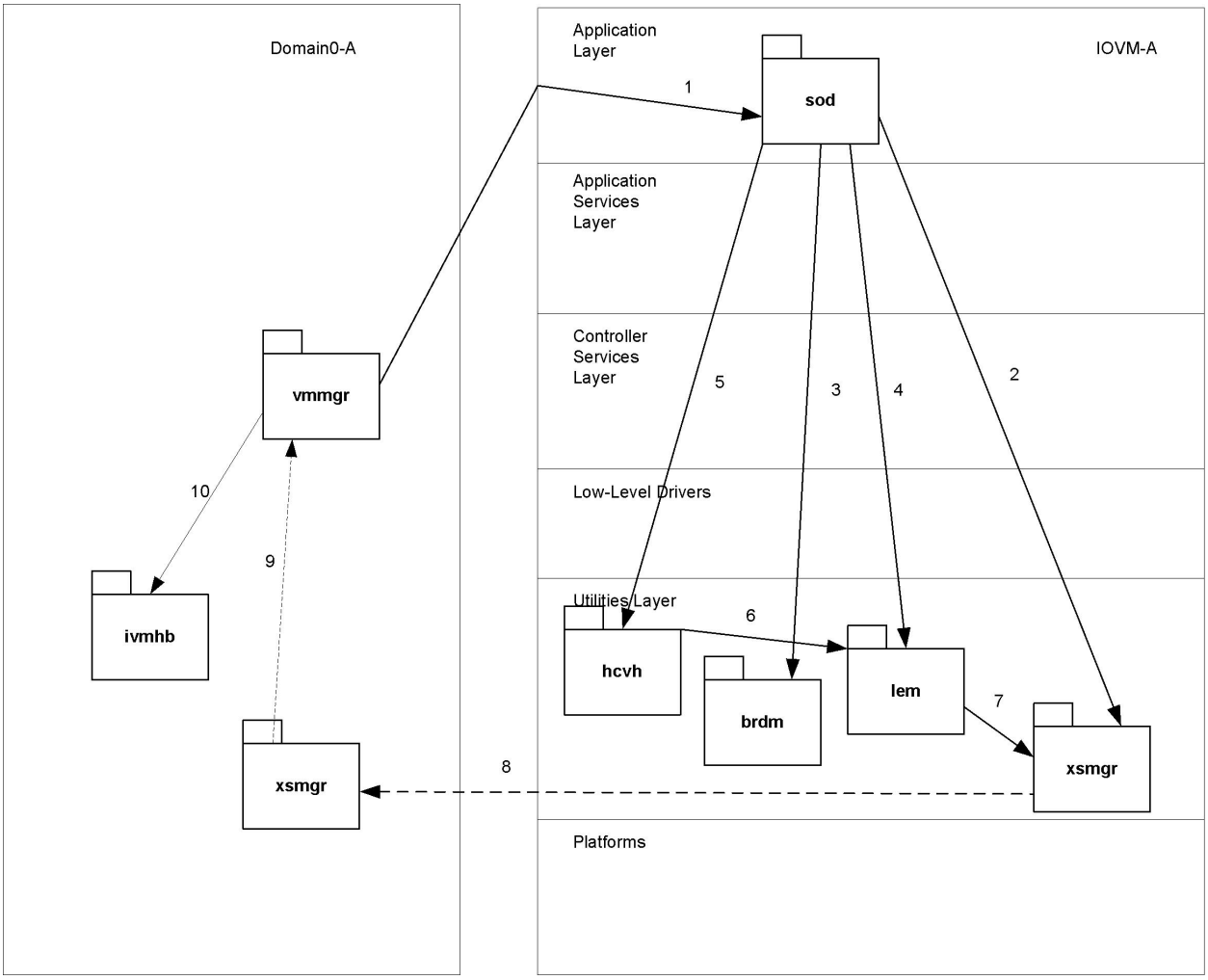
Following is the high-level sequence before and after SBB validation during SOD:

1. [Domain0] vmmgr launches IOVM. As a result of this step, the [IOVM] sodMain starts.
2. [IOVM] sod instantiates xsmgr component.
3. [IOVM] sod instantiates brdm component.
4. [IOVM] sod instantiates lem component.
5. [IOVM] sod instantiates hcvh component who performs SBB validation as mentioned above.
6. **[IOVM] hcvh updates xsmgr about SBB validation success by upda t ing Xenstore ke y /OSA/IOVM/ InitializationState with a value SBB_ VALIDATION_ SUCCESSFUL. The hcvh then blocks [IOVM] SOD processing by waiting on a semaphore.**

7. **[IOVM] xsmgr transmits the status to [Domain0] xsmgr via Xenstore key/value p air /OSA/IOVM/ InitializationState**. Refer to Section 3.1.2.11.2. XenStore Key/Values for more information.
8. **[Domain0] xsmgr updates [Domain0] vmmgr about the SBB validation status**.
9. **[Domain0] vmmgr calls ivmhb to stop the heartbeat timer.**
10. **In case of SBB validation success, the vmmgr checks the alternate in-place line and transmits a keep alive packet to alternate controller if it is in-place. The vmmgr waits for 30 seconds.**
11. **In case vmmgr fails to receive an acknowledgement from the alternate for the keep alive packet for the specified timeout, it suspends further SOD processing within Domain0.**
12. **In case vmmgr succeeds in receiving an acknowledgement from alternate controller or alternate is not in-place, it notifies IOVM via xenstore to resume the IOVM.**
13. **The [Domain0] xsmgr notifies [IOVM] xsmgr , by using /OSA/IOVM/ InitializationState key/value with the value either ALT_BOARD_ID_COMPLETE or ALT_BOARD_ID_TIMEOUT .** Refer to Section 3.1.2.11.2. XenStore Key/Values for more information.
14. **The [IOVM] xenstore component notifies hcvh . After this notification, the hcvh component releases its internal semaphore to resume SOD.**

3.1.1.1.4.2.SBB Validation Failure

Figure 4: SBB Validation failure sequence relative to SOD



Following is the high-level sequence before and after SBB validation during SOD:

1. [Domain0] vmmgr launches IOVM. As a result of this step, the [IOVM] sodMain starts.
2. [IOVM] sod instantiates xsmgr component.
3. [IOVM] sod instantiates brdm component.
4. [IOVM] sod instantiates lem component.
5. [IOVM] sod instantiates hcvh component who performs SBB validation as mentioned in section Section 3.1.1.1.4. SBB Validation.
6. **[IOVM] hcvh calls [IOVM] lem to lockdown controller as a result of SBB validation failure.**
7. **[IOVM] lem upda tes Xen store key /OSA/IOVM/ InitializationState with a value SBB_ V ALIDATION_FAILURE .** In addition to this, the [IOVM] lem performs following actions.
   - Illuminates controller and summary fault LED.
   - Displays seven segment code as described in "Controller Module Indicators FFD".
8. **[IOVM] xsmgr transmits the status to [Domain0] xsmgr via Xenstore key/value pair** /OSA/IOVM/InitializationState. Refer to Section 3.1.2.11.2. XenStore Key/Values for more information.
9. **[Domain0] xsmgr updates [Domain0] vmmgr about the SBB validation failure**.
10. **[Domain0] vmmgr calls ivmhb to stop the heartbeat timer.**

3.1.1.1.5.Controller Type Mismatch

On existing platforms, the lem component handles the controller type mismatch. A controller type mismatch typically results in suspending SOD processing. On Pikes Peak, the [IOVM] lem component will continue to perform controller type mismatch including the host card mismatch, suspend the [IOVM] SOD processing and report the status to [Domain0] olm via Xenstore.

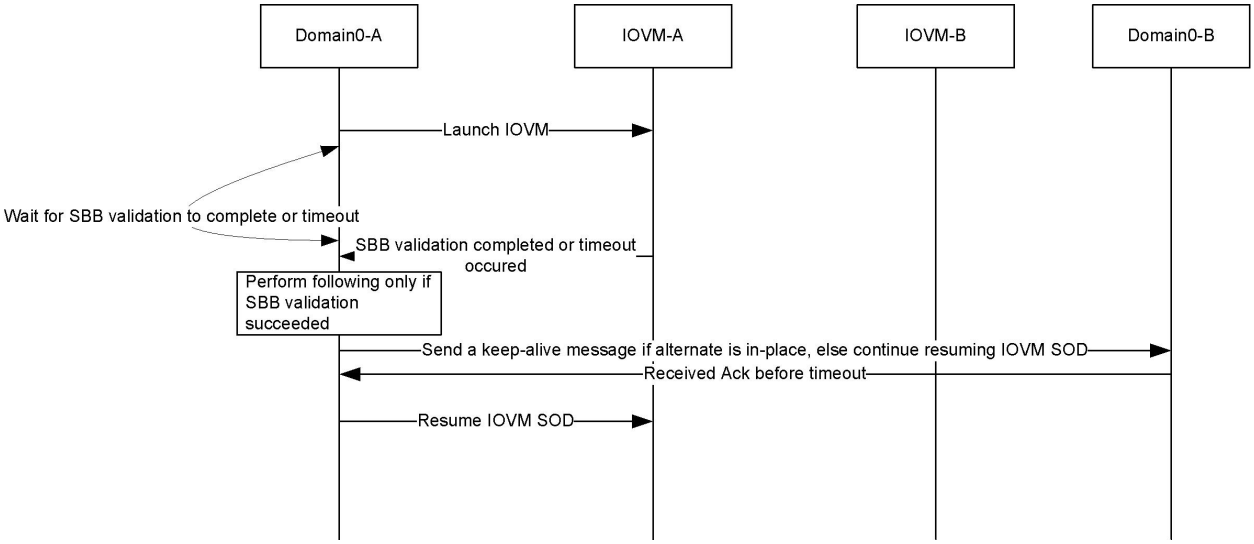
The Domain0 does not process anything further in the init sequence until IOVM returns SBB validation result. After getting SBB validation result, the Domain0 will send a message to Domain0 on alternate controller using the internal NIC and wait for a response unless timeout occurs. The Domain0 notifies [IOVM] sod once it receives an acknowledgement from alternate controller or timeout occurs. [IOVM] sod then resumes processing. Refer to below scenarios for detailed flow.

3.1.1.1.5.1.Scenario 1 (Alternate controller is up and internal NIC is working normally)

In such scenario, the alternate controller (Ctlr B) will respond to this controller's (Ctlr A) keep-alive message. The [Domain-A] will then notify [IOVM-A] to resume its SOD processing. In this scenario, it is expected that [IOVM] lem will succeed in retrieving the alternate controller's board IDs (baseboard and host board ID) unless internal NIC goes down afterwards. The [IOVM] will continue to perform existing controller type mismatch checks.

Following sequence diagram depicts this behavior at a high-level.

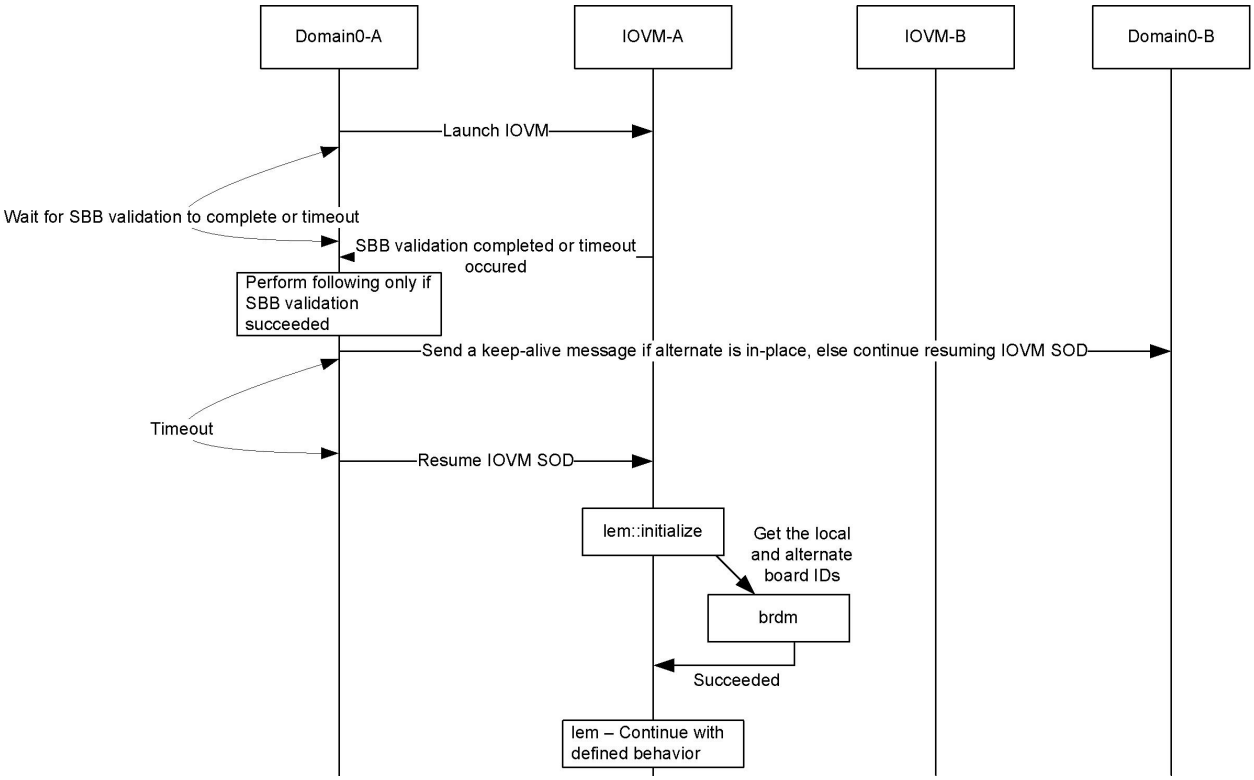
Figure 5: Controller Type Mismatch - Scenario 1



3.1.1.1.5.2.Scenario 2 (Alternate controller is physically in-place but running late)

In such scenario, the alternate controller is physically in-place but either has not reached a stage where it can respond to the keep-alive message or there is a problem with internal NIC. This controller's (Ctlr A) Domain0 will then experience a timeout. The [Domain0] vmmgr will continue to resume [IOVM] SOD after the timeout. Meanwhile, it is possible that alternate controller has now reached a stage where it can respond over the internal NIC. If so, [IOVM] brdm request for alternate board ID will succeed and [IOVM] lem will succeed in performing controller type mismatch check.

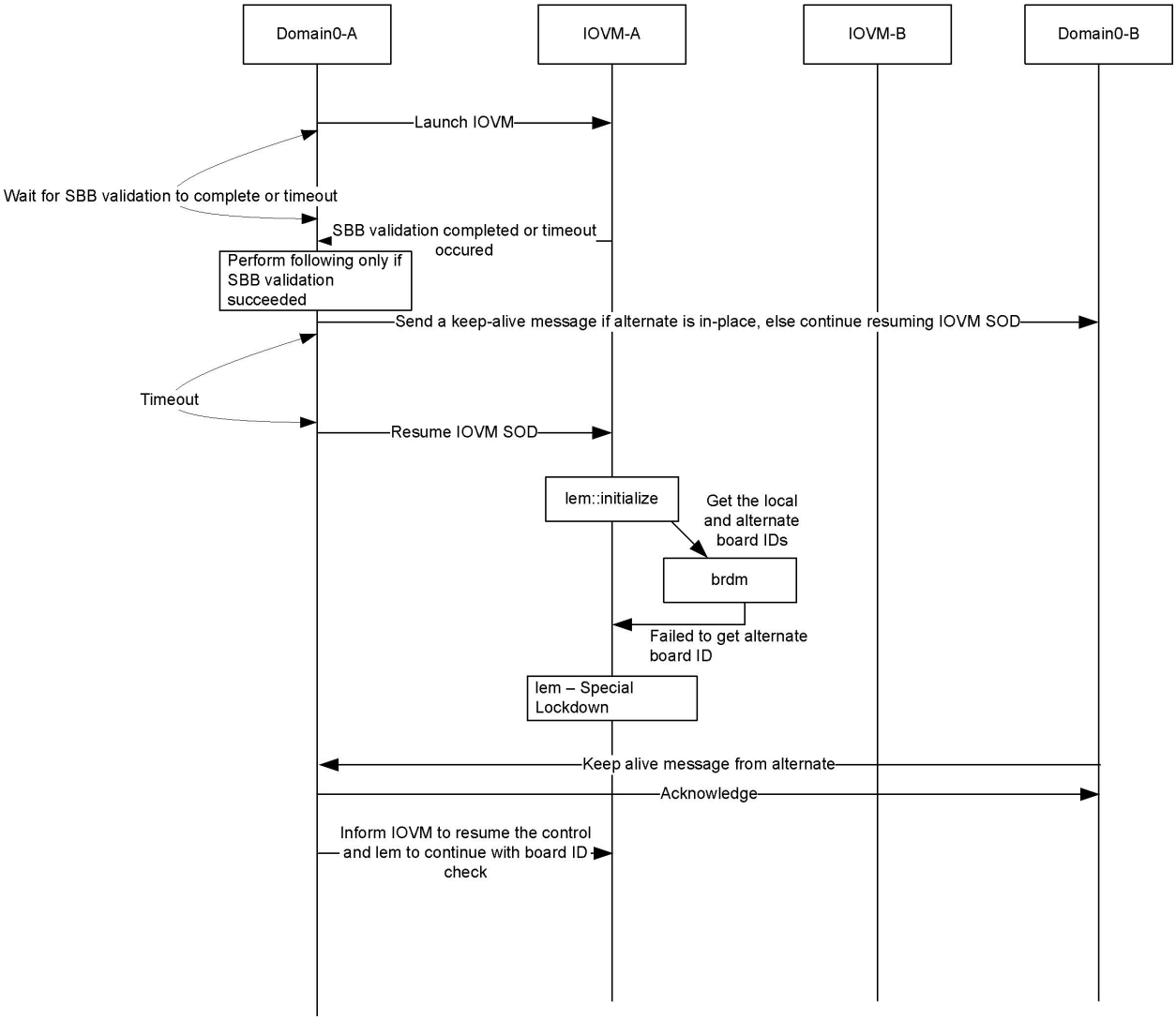
Figure 6: Controller Type Mismatch - Scenario 2



3.1.1.1.5.3.Scenario 3 (Alternate controller is physically in-place but not responding in a timely manner)

In such scenario, the alternate controller is physically in-place but either has not reached to a stage where it can respond to alternate controller's keep-alive message or there is a problem with internal NIC. This controller's (Ctlr A) Domain0 will then experience a timeout. The [Domain0] vmmgr will continue to resume [IOVM] SOD after the timeout. The [IOVM] lem component will fail the controller type mismatch check and will result in special lockdown. In this special lockdown, the IOVM suspends further SOD processing (Block on a semaphore) but listens for a message from Domain0 on local controller. The [Domain0-A] vmmgr will notify [Domain0-A] olm whenever there is a keep alive message from alternate controller (Ctlr B). The [Domain0-A] olm will then notify [IOVM-A] lem via [IOVM-A] olm. The [IOVM] lem component will release the semaphore after this notification from olm and resume IOVM SOD processing.

Figure 7: Controller Type Mismatch - Scenario 3



3.1.1.1.5.4.Scenario 4 (Alternate controller is not in-place)

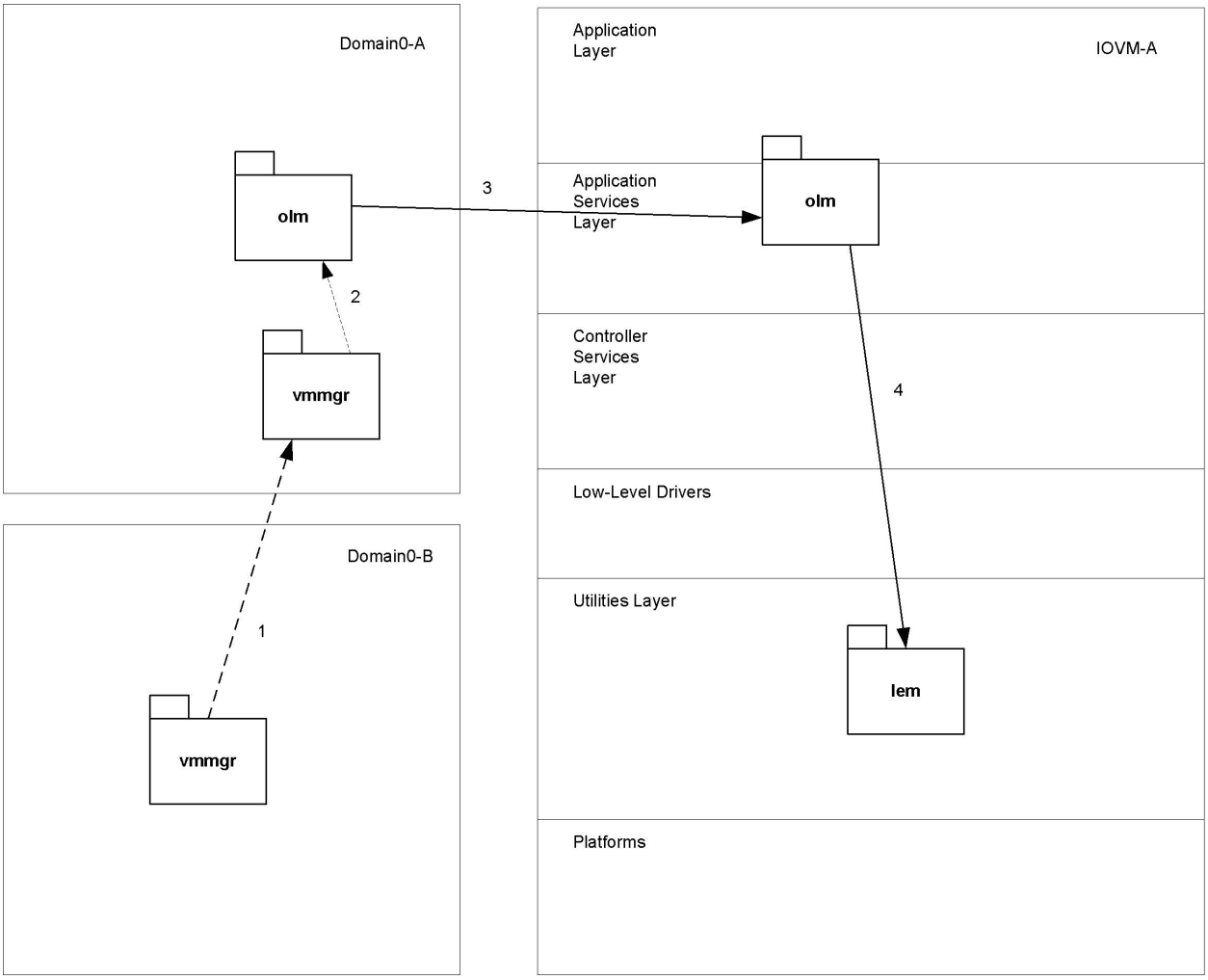
In such scenario, the [Domain0] vmmgr component will notify IOVM to resume its SOD processing. The incumbent controller will skip the controller type mismatch check and continue with SOD processing.

3.1.1.1.5.5.Scenario 5 (Alternate controller is not in-place but inserted later)

In such scenario, the incumbent controller [IOVM] lem will receive an event as soon as alternate is inserted (FW_RUNNING is asserted by alternate) but as opposed to existing behavior, the lem will not perform controller type mismatch. The Domain0 in incumbent controller (Ctlr A) will continue to wait for keep-alive message from alternate controller (Ctlr B) and after receiving that message, it will follow the below sequences to perform the controller type mismatch.

Following diagram depicts the call sequences in this scenario:

Figure 8: Controller Type Mismatch - Scenario 5



1. [Domain0-B] vmmgr sends a keep-alive message to [Domain0-A].
2. [Domain0-A] vmmgr notifies it to olm.
3. [Domain0-A] olm communicates to [IOVM] olm.
4. [IOVM-A] olm notifies [IOVM] lem about this. The [IOVM] lem performs controller type mismatch.

It is to be noted that xenstore will not be used in this scenario.

3.1.1.1.5.6.Special Lockdown

As described above, the following behavior is expected as part of this lockdown:

- IOVM suspends SOD processing in lem::initialize. The [IOVM] lem can just block on a semaphore.
- [Domain0] vmmgr notifies [IOVM] lem via olm if there is a keep-alive message from alternate.
- [IOVM] olm notifies lem who then releases semaphore, if in SOD.
- [IOVM] lem performs controller type mismatch and resumes SOD processing if in SOD (as described in scenario Section 3.1.1.1.5.3. Scenario 3 (Alternate controller is physically in-place but not responding in a timely manner), otherwise just performs controller type mismatch (as described in scenario Section 3.1.1.1.5.5. Scenario 5 (Alternate controller is not in-place but inserted later)).

A new lockdown which will allow Domain0 to listen for keep alive message from alternate and if so, then it

will release this controller from the lockdown.

3.1.1.1.6.Power Supply SBB validation and PMBus support

Pikes Peak is an SBB controller. SBB mandates that controllers may only draw 60 watts of power until verifying that the power supply can supply more power. The Pikes Peak controller's SBB implementation will relax this requirement and may draw maximum 100 watts of power until power supply validation. Pikes Peak at full capacity draws 150 watts.

Pikes Peak uses PMBus to communicate with the power supplies. PMBus is an open standard protocol used to communicate with power supplies.

The Power supply validation is a two step process as described below:

1. Power Supply validation in hardware: After power on, the Pikes Peak SAS expander processor checks the power supply for adequate capacity. To do this, it configures the I2C buses to allow access to both the power supplies. Using PMBus, the expander validates that the power supplies can supply at least 150 watts of power. The expander then allows controller to boot if both the power supplies are found to be adequate. The controller will lock down for PS hardware validation failure and a code "SE Sb" will be displayed on seven segment display.
2. Power Supply validation in Firmware: Before drives are spun up at power-on, the controller firmware ensures that the power supplies are valid for the enclosure. The controller firmware will verify the power supply using PMBus command for vendor ID (using MFR_SPEC_32 PMBus command) to be LSI and the capacity (using MFR_POUT_MAX PMBus command) to be **725** W.

The controller firmware must set the appropriate power supply register (0xD0h) to "1" to instruct power supply to respond to PMBus commands only. This must be done before power supply SBB validation. The hcvh and pwsplm components will be changed for the above mentioned functionality.

Architecture Note: Part of this section will be moved to existing FFD.

3.1.1.1.7.HIC Support

The Pikes Peak controller supports various Host Interface Cards (HICs):

- Dual-port 40 Gb/s InfiniBand HIC (Elk Park) - This HIC uses Mellanox's next generation, ConnectX based, IB chip from Mellanox.
- Quad-port 2, 4, or 8 Gb/s FC HIC (Manitou) – This HIC uses the same chip, Tachyon QE8, as was used by FC HIC on Snowmass.
- Dual-port 10 Gb/s (iSCSI-SFP+ optical) (Glen Cove) - This HIC uses the same chip, BE2, as was used by iSCSI HIC on Matterhorn.

The Pikes Peak controller baseboard has four 8 Gb/s FC ports. The Pikes Peak controller also has provision for pluggable, field replaceable HICs. The field replaceable HICs can support FC, InfiniBand, or iSCSI protocols in addition to the file based protocols such as CIFS, NFS.

3.1.1.1.7.1.FC and InfiniBand

The FC and InfiniBand HICs will be owned by IOVM. The FW changes required to support IB and FC HICs include the following:

- The changes in ssm component to support the host boards and report these into object graph.
- Feature Model changes to support the new HICs.
- The IB driver changes to handle the new Host Interface Cards. IB HIC uses ConnectX based new generation Mellanox chip requiring new IB driver to be introduced in controller firmware.

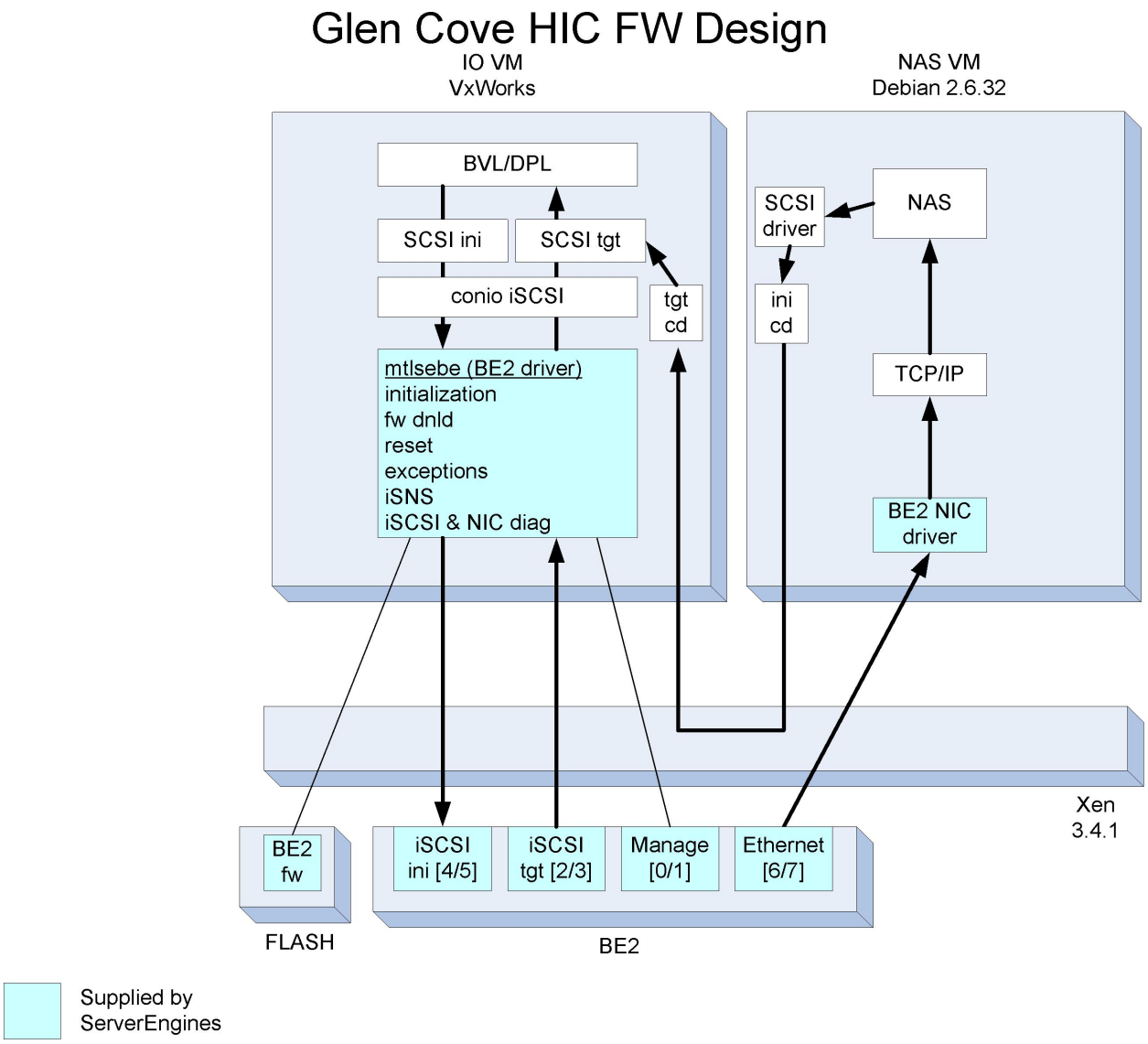
- Since the HICs are field replaceable, a basic diagnostic test run during SOD for newly installed HIC.
- SYMbol changes to add the 40 Gb/s Interface speed for IB.
- Model Config table updates due to the new host interface cards.
- It is to be noted that firmware will see one device for each PCI function.

3.1.1.1.7.2.Glen Cove (iSCSI and Ethernet HIC)

The Pikes Peak controller also has a provision to attach an Ethernet/iSCSI HIC, Glen Cove. The Glen Cove card supports two Ethernet ports. As a result, a two controller array system can support up to four Ethernet ports. This section describes the parameters specific to this host interface card. For more information on the iSCSI host interface, see the appropriate FFD.

Each Ethernet port is implemented by separate PCI functions for un-accelerated Ethernet protocol and accelerated iSCSI Initiator/Target protocol. The iSCSI Initiator and iSCSI Target functions are implemented by separate PCI functions due to BE2 implementation details, but they share one MAC address. Figure 9 shows the PCI assignments. The PCI function numbers are shown in []s, formatted as [port0 fn / port1 fn].

Figure 9: PCI function assignments for Glen Cove



The LED behavior for the HIC will be same as described in document # 34533-00, the Feature Function Definition for Controller Module Indicators.

Since Xen 3.4.1 does not support exposing multi-function devices to a single domain, each function will have a unique subsystem id using function 0. This is only important for the IOVM because six PCI functions for each BE2 will be assigned to it. So, the IOVM will actually see six different single-function PCI subsystem ids. The values for these six ids will be mtlsebe compile-time constants. The BE2 will route incoming Ethernet frames to the appropriate PCI functions in a manner so as to preserve each function's operational independence and to support the NAS VM's requirement to implement LACP (with aggregated MACs) in its linux Ethernet stack.

The BE2 IPL file will be formatted so as to reserve 4 MAC addresses per port (3 of which will be used). The MAC addresses for each port will be assigned such that all of the addresses assigned to each port are sequential. When/if the 4th MAC address becomes used, it should not require that the IPL be modified.

Each Ethernet port supports data rate of 10 Gb/s.

The iSCSI initiator, target, and iSNS functions will use a VxWorks mtlsebe driver which operates similar to that of the Snowmass system. The Ethernet function which is assigned to the NAS VM will use a driver compatible with linux 2.6.31. Since the Ethernet functions are being used in a stateless manner, the combination of iSCSI and Ethernet functions will not affect the resources available for iSCSI processing. The iSCSI target accepts up to TBD connections per session on any iSCSI portal, as negotiated during login Connections within a session are identified by a Connection ID (CID). An informational event log entry is made whenever a connection is unexpectedly terminated or lost, or if a TCP time-out occurs. A maximum of 506 simultaneous iSCSI connections can be accommodated per Ethernet port. Up to 1012 sessions per controller or 2024 sessions total are supported in a two controller array. The iSCSI target supports up to 506 simultaneous sessions per port, with a maximum of 2024 sessions for a two controller array. The number of sessions that can be supported may be reduced if multiple connections per session are used. The number of sessions may also be limited by the maximum number of initiators the target can accommodate. If the controller has insufficient internal resources, new logins will be refused until sufficient system resources become available.

Architecture Note: Double check these resource numbers and discuss the allocation between initiator and target modes.

The iSCSI initiator and target do not support T10 PI data protection first out but will support in future releases as per following description. Type 1 and 2 PI are required for both initiator and target modes. The CRC is calculated by the hardware. The tags are provided by the application software. The following operations with indicated options provided by the application software are supported for all data transfers (initiator and target, both ingress and egress):

- noPI
- insertPI(app tag, initial incrementing ref tag)
- removePI(check CRC flag, check app tag flag, check ref tag flag, app tag, initial incrementing ref tag)
- forwardPI(check CRC flag, check app tag flag, check ref tag flag, replace CRC with hw-generated flag, replace app tag flag, app tag, initial incrementing ref tag)
- Orion and the other iSCSI-only programs which use the BE2 will use the same mtlsebe source code from ServerEngines. Conditional compilation flags will be used to distinguish the products at compile time. As has been done in the past, as LSI integrates the mtlsebe code into the LSI source code control system, appropriate use of Gears controls will be added within makefiles and/or added header files to control the conditional compile directives. The source files provided by ServerEngines will not be modified to contain Gears-controlled directives.

3.1.1.1.7.2.1.Scenarios

Existing iSCSI based controller platforms does not support iSCSI HIC diagnostics; however HIC diagnostics is required to be supported on Pikes Peak.

Initialization

- The Domain0 VM assigns iSCSI Initiator, iSCSI Target, and Chip Management PCI functions to IOVM.
- The Domain0 VM assigns Un-Accelerated Ethernet PCI functions to File VM.
- The Domain0 VM starts IOVM (performs SOD sequencing just like the existing iSCSI/SAS product).
  1. The sod component instantiates the pmi component.
  2. The sod component instantiates the ioni component. a). The ioni component determines if an iSCSI HIC is installed
  3. The sod component instantiates the ioni component. b). If an iSCSI HIC is installed, the isni component loads the iSCSI.dl FLASH file, which contains the iSCSI components, including the mtlsebe component.

4. The sod component instantiates the pm component.
5. The sod component instantiates the b_isn component.
6. The sod component instantiates the mtlsebe component. a). The mtlsebe component registers for Mtl-level parameters with the pmi component. b). The mtlsebe component registers with isni as an Mtl
7. The ioni component assigns all the IO interface channels, using the supportsDeviceId interface.a). The mtlsebe component responds with true for the proper PCI ids.
8. The ioni component initializes all the IO interfaces. a). The b_isn component calls the mtlsebe's instantiateMtlInterface interface. b). The mtlsebe component does nothing with this.
9. The sod component enables host interface channels, via the ioni component. a). The ioni and hcvh components determine if any of the installed Glen Cove HICs have not been tested before in this system. b). If not, the ioni component requests that the b_isn component run an internal diagnostic for each channel on the HIC. Note that this diagnostic is performed prior to updating the BE2 firmware. Although this may seem counter-intuitive (to avoid running the diagnostic using "old" firmware), this sequence is consistent with usage with other HICs, and changing it was deemed unnecessary. This is particularly true since the only time the diagnostic will be executed is after a system power cycle during which the HIC is added. Therefore, it would not normally be the case that a new HIC and new firmware would be seen during the same SOD sequence. c). The ioni component requests that the b_isn component enable the interface for each channel:
10. i). The b_isn component relays the request to the mtlsebe component
11. ii). The mtlsebe component registers with pmi for MtlInterface-level parameters
12. iii). The mtlsebe component reports its interfaceCreated to the isni component
13. iv). The mtlsebe component checks the version of firmware in the BE2 FLASH
14. v). If it does not match what is in the BladeEng2.dl FLASH file, the mtlsebe component updates the BE2 FLASH with what is in the FLASH file.
15. The sod component initializes the b_isn component.
16. The sod component initializes the pm component. a). As the pm component reads the parameter database, pmi events are relayed to the mtlsebe component. b). This causes the mtlsebe component to validate the iSCSI operational parameters, within the context of a sod transaction.
17. When the sod transaction is completed successfully, the pm component relays a pmi event to the mtlsebe component. This causes the mtlsebe component to activate the iSCSI operational parameters.

- The IOVM reports the status of HIC diagnostics to Domain0.
- The Domain0 VM starts File VM once IOVM completes HIC diagnostics.

Architecture Note: It is assumed that current behavior is to just log the MEL event and continue with SOD. It is assumed that NAS VM will independently handle if there are any issues with Ethernet HIC. The NAS VM should not depend on diagnostic results.

BE2 firmware download

The BE2 firmware is packaged with the controller firmware in the BladeEng2.dl FLASH file. It is downloaded to the BE2 FLASH during initialization, as described above.

BE2 reset

No changes are required to reset handling. Either VM may cause a BE2 reset, or the other VM's driver will detect that it has occurred.

There is a new reason for a BE2 reset to occur. As part of diagnostics, the BE2 may be reset at least once. This will not cause any problems because diagnostics are never executed while the BE2 under test is processing IOs.

BE2 exceptions

- The IOVM's mtlsebe component will monitor for BE2 chip-level exceptions.

- The mtlsebe component will relay them to the isni component via the isni::MtlEvents::interfaceError() interface with new interfaceErrorType values to be defined during implementation. This will require instantiating a temporary MtlInterface object (for which there will be no MtlInterfaceCreated event).It will handle any required controls using the BE2's management PCI function (such as requesting a chip reset).
- The isni component will relay them to b_isn.
- The b_isn component will handle all aspects of the error, including logging.

BE2 diagnostics

1. Manufacturing

- Performed on linux systems, using standard tools. The version of linux will be specified by ServerEngines.

Deployed System

- ServerEngines will be responsible for implementing this: The code will be part of the implementation of the MtlInterface class, which already contains most of the required interfaces. It can leverage existing diagnostics code in the mtlsebe/diag directory, so very little actual coding will be required.

Level-0

1. This diagnostic is run every system boot. It is covered by the POST which is already running during boot, so no additional work is required.
2. POST is defined in the :Storage Blade Diagnostics-HWI APIs" document (version v.0.2) as follows:
3. "POST is entirely ARM firmware based. It is initiated from Redboot and performs the following tests:

- DDR calibration
- DDR Tests
- SEEPROM CRC test
- Flash ROM CRC test"
- *Internal*

1. This diagnostic tests each HIC the first time it is installed on a particular controller. It is part of the initialization procedure, as described above.
    - The b_isn component requests that the mtlsebe component run the diagnostic, using the MtlInterface::runInternalDiagnostic interface.
    - When the diagnostic is complete, the mtlsebe component reports the status, using the ioni:: internalDiagnosticDoneEvent interface.
    - If the diagnostic does not complete within 90 seconds, the mtlsebe component will force it to complete with an appropriate error status at that time.
2. This diagnostic tests both iSCSI I&T as well as Ethernet PCI functions. Since boot has just completed, POST has just been performed, so loopback testing is all that is required. The following tests will be included (other bediag calls will be required, such as bediag_initialize, bediag_chip_reset, but the sequence of these will be determined by ServerEngines):
    - "bediag_change_phy_loopback_setting" to loopback at the PHY
    - "bediag_network_loopback_test" with setting to test for about one minute. Upon completion, the chip and PHY will be returned to a normal operating configuration.

- *Extended*

1. The Service VM implements the RPC function and relays the request to the IOVM, using the existing RPC mechanism.
2. The IOVM bcdm component implements the RPC function and relays the request to the ioni component.

3. The ioni component relays the request to the appropriate ion:Lld (implemented by the b_isn component for iSCSI channels).
4. The b_isn component reports that the diagnostic is started, using the ioni::IonLld::updateHICDiagnosticStart interface.
5. The b_isn component relays the request to the mtlsebe component, using the MtlInterface::startHICDiagnostic interface.
6. The mtlsebe component performs the requested diagnostic.
7. During the diagnostic, the mtlsebe component will report percentages completed at intervals, using the isni::MtlEvent::onDiagPercent interface:
   - Upon completion of POST, it will report 10% complete.
   - For the loopback test (which should last about one minute), it will report another 10% complete every 7 seconds, until it is 90% complete (after 56 seconds), at which point it will stop reporting percentages.
   - The b_isn component will report this progress using the ioni::IonLld::updateHICDiagnosticProgress interface.
8. At any time, if the test is completes, the mtlsebe component will stop reporting percentages and report the status, using the isni::MtlEvent::onDiagStat interface.
   - The b_isn component will report this, using the ioni::IonLld::updateHICDiagnosticFinish interface.
9. If the diagnostic does not complete within 90 seconds, the mtlsebe component will force it to complete with an appropriate error status at that time.
10. At any time during the diagnostic, another RPC function may be received which requests that the diagnostic be cancelled. This is implemented using the same components as the start diagnostic. To accomplish this, the mtlsebe component will reset the BE2. When the reset is complete, the mtlsebe component will report, using the isni::MtlEvent::onDiagStat interface. Since the controller is in service mode, all IOs are stopped. This test will run the equivalent of POST (bediag_run_post) followed by the same test procedure used for Internal diagnostic above. This will result in at least one chip reset, but that is ok. In fact, all of the drivers which implement this diagnostic currently reset their IOCs.

- *Runtime*

1. This diagnostic is executed upon request from the alternate controller (if it suspects things are not working for some reason).
2. There is no additional work needed for this because iSCSI does not allow our port to send itself an IO request, the way we do in Fibre Channel. Therefore, the runtime diagnostics code will just skip that part of the test, the same as it does in our current 1G iSCSI products.

Block IOs

- This functions exactly as on the existing iSCSI/SAS product, using the BE2 iSCSI I&T PCI functions which are directly assigned to the IOVM.

File requests

- This functions exactly as on a standard linux product, using the BE2 as the Ethernet interface.

3.1.1.1.7.3.Controller Lockdown due to mismatch or unsupported HIC

Controller will self-lockdown if it discovers an unsupported HIC during SOD. The [IOVM] lem component will continue to perform unsupported and mismatch host board ID check during initialize phase. The lem component calls brdm to get the local and alternate controller's host board ID. If the host board ID checks fail, lem sets a lockdown context allowing sodMain to transition to deferred lockdown SOD mode. If ACS determines that no auto code synch is required and controller is in deferred lockdown state, then [IOVM] lem initiates a lockdown. This will suspend the [IOVM] sodMain task and notify Domain0 for the lockdown.

Below table describes various lock-down scenarios. The [IOVM] lem component continues to perform host board ID mismatch check.

Table 11: Ctlr Lockdown scenario due to HICs

Ctlr A HIC	Ctlr B HIC	Behavior
Supported	Supported	Normal
Supported	Not Supported	Ctlr B self-lockdown
Not Supported	Supported	Ctlr A self-lockdown
Not Supported	Not Supported	Ctlr A and Ctlr B self-lockdown
Supported but differs from Ctlr B HIC	Supported but differs from Ctlr A HIC	Booting Ctlr enters in a suspended state

Architecture Note: The lockdown behavior will be defined in Serviceability AAD. There are no changes in external behavior for the unsupported and mismatched HIC.

3.1.1.1.7.4.Cache backup device diagnostics

Upon receiving diagnostic request for the cache backup device, the controller firmware will run diagnostics on the cache offload partition of cache backup device. If there are any issues with the portion outside cache offload, then either controller will not be able to boot successfully or will encounter issues while updating the information such as configuration, logs, etc., on cache backup device. Such errors must be handled by the owners of respective partitions.

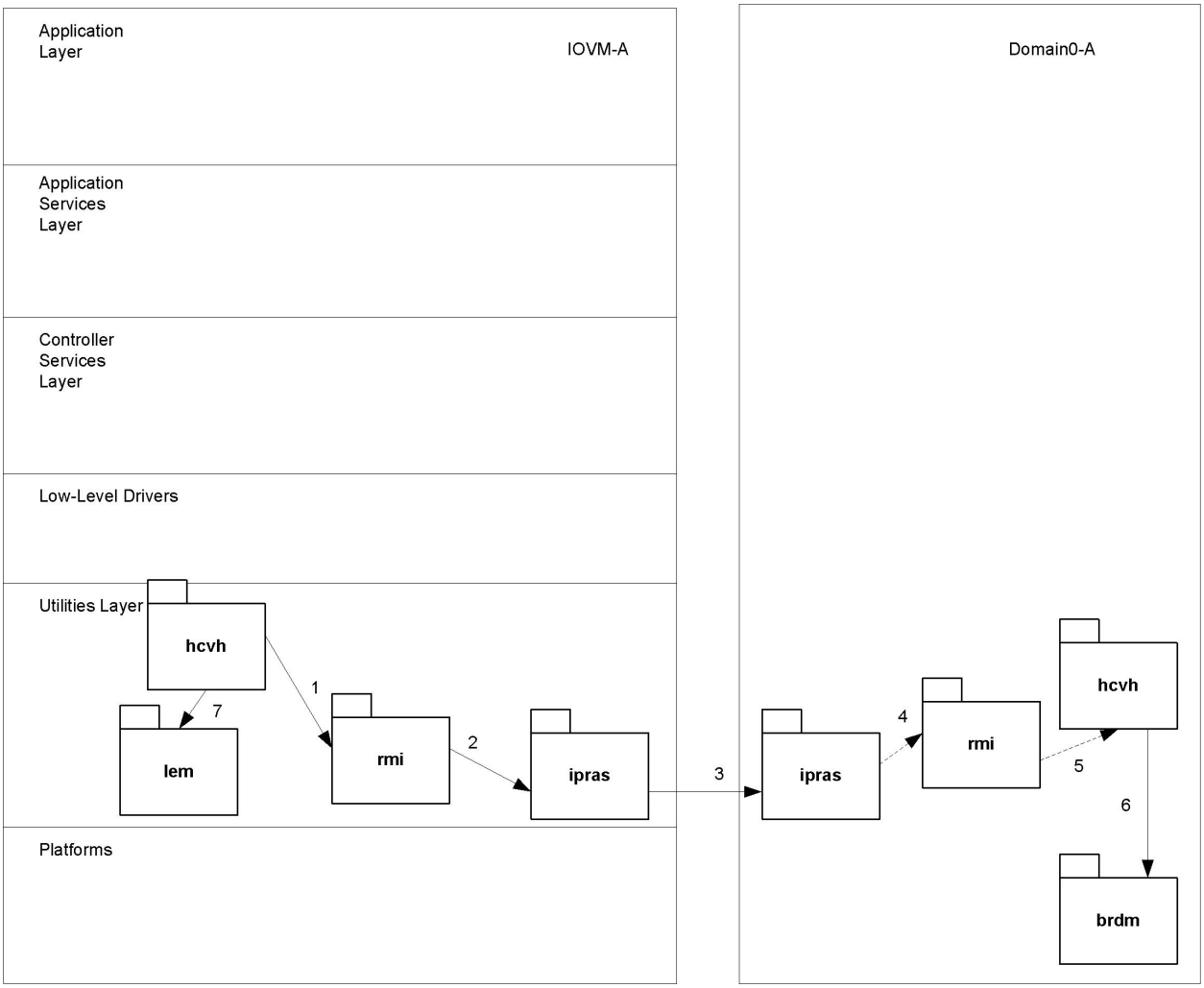
Architecture Note: This section will move to Serviceability AAD? We are not expecting any changes in our current implementation if we retain the behavior as described above. Controller does not perform any cache backup device diagnostics during SOD?

3.1.1.1.8.Verify DIMM configuration

On Pikes Peak, IOVM can access the portion of memory assigned to it. The memory can be part of one DIMM or can span across multiple DIMMs. The Memory assigned for RAID cache will be contiguous but that may not be the case with processor memory. On current platforms, the hcvh component uses BCM interfaces to retrieve the DIMM SPD data, but on Pikes Peak the [IOVM] hcvh can't access the DIMM SPD data. It needs to call [Domain0] hcvh to get the DIMM SPD data.

The hcvh component in IOVM will continue to perform DIMM configuration verification as it does for XBB2. However, on Pikes Peak every DIMM slot has to be populated with same capacity DIMMs. The PP200 will have 2 DIMMs each of either 2GB or 4GB, whereas PP400 will have 3 DIMMs each of either 4GB or 8GB. Any DIMM configuration outside this will be considered invalid. For an invalid DIMM configuration, the hcvh component will continue to display the existing seven-segment code, suspend the IOVM and report the configuration error to lem component which will further report the error to Domain0 using xenstore. The [IOVM] hcvh component performs DIMM verification in instantiate phase, before SBB validation.

Figure 10: DIMM Verification



Following are the call sequences:

1. [IOVM] hcvh component calls rmi to get the DIMM SPD data.
2. [IOVM] rmi calls ipras.
3. [IOVM] ipras calls [Domain0] ipras.
4. [Domain0] ipras calls rmi.
5. [Domain0] rmi calls hcvh.
6. [Domain0] hcvh calls brdm interfaces to get the DIMM SPD data.
7. The [IOVM] hcvh component then performs DIMM verification and calls [IOVM] lem to lockdown the controller if verification fails.

3.1.1.1.9.Disable USB Port

The USB port can be used to boot the bare metal controller and at that point, the actual controller firmware images can be transferred to SATA flash using some external utility. The controller can then reboot with the FW image on SATA flash drives. Once controller identifies valid firmware images in SATA flash, it will disable the USB port. The booting from USB drive is perceived in extreme field debugging or manufacturing scenario. The [Domain0] brdm component is responsible for disabling the USB port if it becomes a requirement.

Architecture Note: Is it really a requirement? This functionality is currently in-place except disabling the USB port in normal running. Do we really need to do anything in Domain0 to disable the USB port? It is possible that user may try to mount the USB drive and start using it for any undesired operation?

3.1.1.1.10.Push Button Support

The push button is used to choose the boot device and reset the IP address. If the user holds the push button during boot, the BIOS will detect this and follow through the boot sequence as described in earlier sections of this AAD. After normal boot, the same button can be used to change the IP address. Refer to Hypervisor AAD - Virtualized hardware resources for more information on resetting IP address. Refer to Serviceability AAD for more information about various boot options.

3.1.1.1.11.SATA Flash handling

3.1.1.1.11.1.Flash Layout record

The flash will use Master Boot Record (MBR) partitioning scheme. It will use first 128K bytes to maintain meta-data. The meta-data will describe the flash layout, and the other important metadata. It is required that controller firmware updates the metadata whenever it detects a new flash drive without correct signature. A new component "fpmgr" (Flash Partition Manager) in Domain0 is responsible to manage the metadata. This component will provide interfaces to other application components as required.

Architecture Note: What exactly is required in metadata is yet to be determined. It will evolve as we make progress. One of the use-case could be to synchronize this metadata with dacstore (using DRM) to handle flash replacement scenarios.

3.1.1.1.11.2.Flash partitioning

The SATA flash will have last 4GB reserved for cache offload.

Architecture Note: The partitioning of 1st 4GB is still TBD.

3.1.1.1.11.3.Default Boot Partition

The BIOS will boot from the first partition of the SATA flash disks in slot 0 (0 relative). The BIOS will display 7-segment code if it is not able to boot from disk in slot 0.

3.1.1.1.11.4.Invalid SATA flash disk configurations

Refer to "SATA Flash Disk Support" section for the supported configurations. The [IOVM] hcvh component is responsible for validating the SATA flash disk configurations. The [IOVM] hcvh component will call [IOVM] brdm component to retrieve the SATA flash related information. The [IOVM] brdm component will get the required information from Xenstore.

On Pikes Peak, Domain0 owns SATA flash disk(s). The disks are used for various purposes including cache offload which requires block I/O access to IOVM running VxWorks. The IOVM requires responses for SCSI commands such as INQUIRY, READ CAPACITY, etc., to validate the configurations.

Citrix agreed to implement following in Domain0 to accomplish the required functionality:

At backend device setup time, the storage backend in Domain0 issues the SCSI commands to get additional device information and puts that information in xenstore. The blkfront driver running in IOVM reads the information from xenstore and uses that information to implement the appropriate SCSI commands. The only limitation of this scheme is that the data is static and it needs to be pre-determined

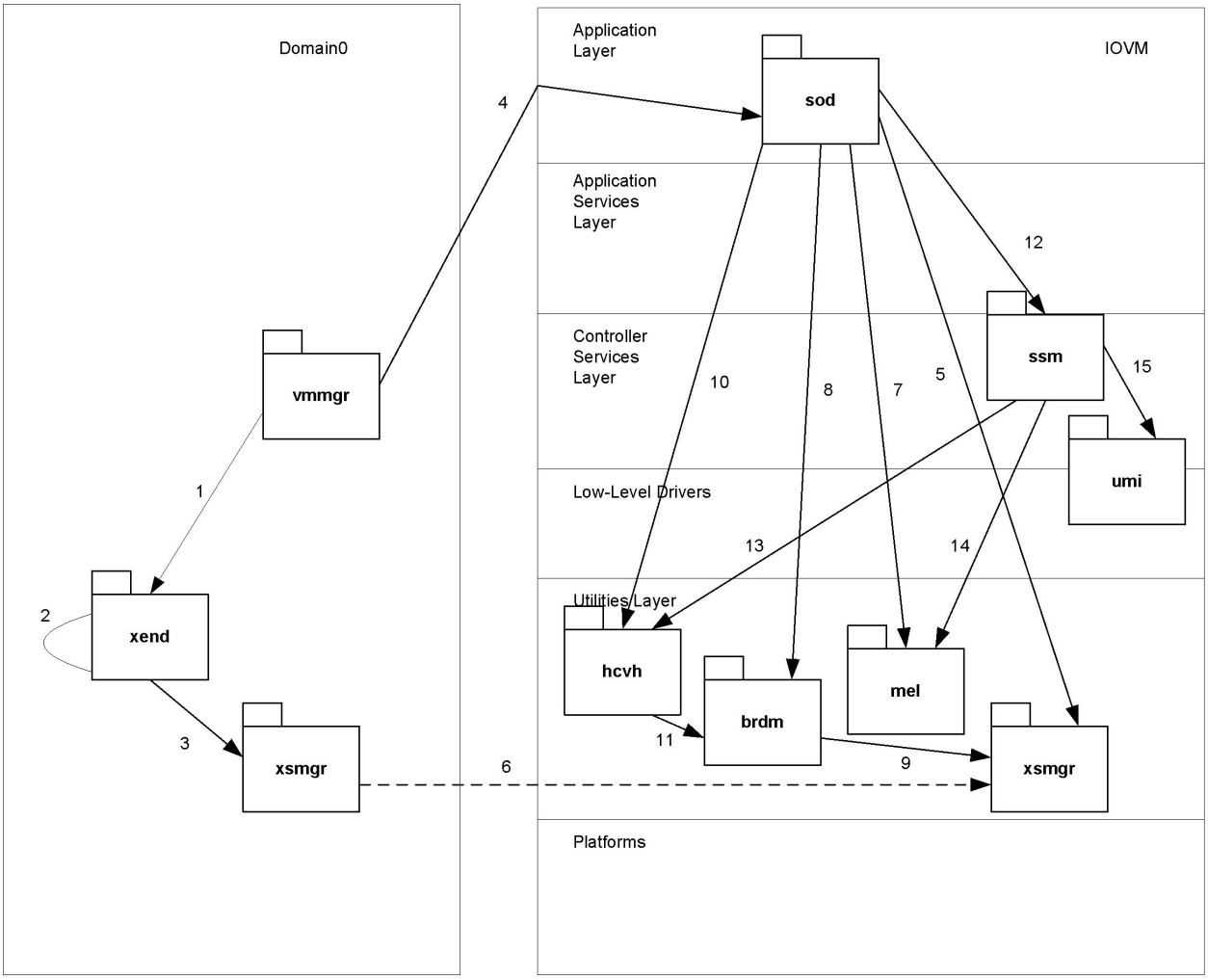
what queries will be made. Of course the scheme can be extended to do queries on demand, by having a daemon in Domain0 which puts additional data into xenstore when the frontend requests such information, but current requirements can be satisfied without this.

Algorithm:

1. When creating a domain, xend constructs the appropriate entries in xenstore for the front and backend devices. A hook will be added in the xend code to call out a script from a well-known location with a set of well-known command line arguments such as device name or major-minor code (exact arguments TBD). This script gathers the necessary information (by calling sg utilities and such) and returns the information to caller in a structured way. The xend will decode this information and place the right information in xenstore in well-defined locations before actually starting the domain (IOVM). The device name mentioned above is the name of the device specified in VM configuration file provided for the VM creation (xm create -c <conf.file>). The xend will extract the device name from the conf file, and pass that into the script.
2. It is possible that few command syntax may be incorrect or device does not support few commands or few/all commands may fail due to some transient issues. In such scenario, it is possible that script returns an error code indicating the type of error encountered. xend can then determine whether it should fail the domain (IOVM) creation, or it should propagate this error into xenstore and continue with the domain creation to allow hcvh and blkfront driver to take specific action. The specific actions for invalid SATA flash disk configurations are described in requirements section of this AAD. In case of error, the hcvh will consider the SATA disk drives as invalid and take the required actions based on the configurations.

Following sequence diagram illustrates the call sequences:

Figure 11: SATA flash disk configuration validation



1. [Domain0] vmmgr calls xend to create IOVM.
2. [Domain0] xend calls the script to collect SATA flash information as described above.
3. [Domain0[xend places the information in Xenstore.
4. [Domain0] vmmgr actually creates VM. This is as part of step 1 mentioned above but specified separately for the clarity.
5. [IOVM] sod initializes xsmgr.
6. [IOVM] xsmgr is notified by [Domain0] xsmgr about Xenstore entries. This step is performed indirectly, but mentioned here for clarity.
7. [IOVM] sod instantiates mel.
8. [IOVM] sod instantiates brdm.
9. [IOVM] brdm retrieves the SATA flash drives information from Xenstore. Refer to Section 3.1.2.11.2. XenStore Key/Values for more information.
10. [IOVM] sod instantiates hcvh.
11. [IOVM] hcvh gets the information about SATA flash drives from brdm and performs SATA disk configuration validation.
12. [IOVM] sod initializes ssm.
13. [IOVM] ssm calls hcvh for the SATA disk configuration validation results.
14. [IOVM] ssm logs a critical MEL event for invalid configurations.
15. [IOVM] ssm reports a Needs Attention condition for invalid configurations.

3.1.1.1.12.CPU Temperature Monitoring

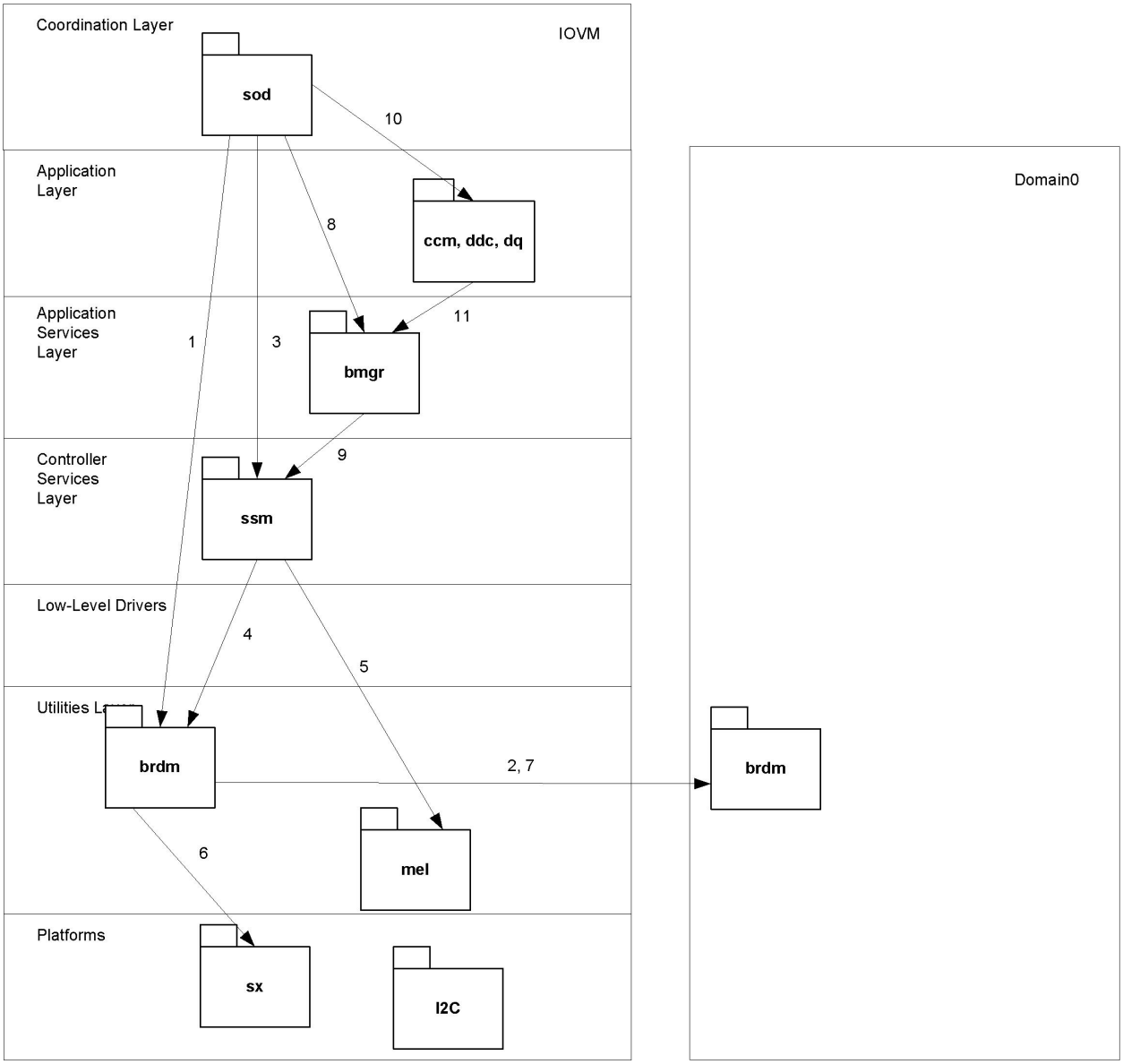
The Pikes Peak controller has same external behavior as on Snowmass if CPU temperature exceeds the threshold (Normal and Maximum). However, the mechanism to retrieve the CPU temperature is different on Pikes Peak as opposed to Snowmass. On Pikes Peak, the processor does not provide the absolute temperature, but provides the temperature relative to the TCC (Thermal Control Circuit) activation temperature (74.6 deg C). The [IOVM] brdm component will provide interfaces to [IOVM] ssm to set the temperature Threshold # 1 and Threshold # 2. Note that this Threshold is relative to the TCC and not an absolute temperature. The minimum resolution for Threshold is 1 deg Celsius. An interrupt will be raised if the CPU temperature exceeds the Threshold # 1 and Threshold # 2. Subsequent sections treat "Threshold # 1" as "Normal Threshold" and "Threshold # 2" as "Maximum Threshold" to align LSI terminology with Jasper Forest CPU terminology.

Architecture Note: Actual values of Threshold # 1 and Threshold # 2 is TBD and will be specified based on thermal measurements.

3.1.1.1.12.1.Setting CPU Threshold Temperature

Following diagram illustrates the component collaboration during [IOVM] SOD processing:

Figure 12: Setting CPU temperature threshold values



- 1. [IOVM] sod instantiates brdm component.
- 2. [IOVM] brdm component communicates with [Domain0] brdm component and gets the value of "Thermal Status Log" bit in IA32_THERM_STATUS MSR. The [Domain0] brdm reads the register and clears this bit. The [Domain0] brdm component returns the value of the bit just read to [IOVM] brdm. This bit indicates the activation of CPU TCC during previous boot. The brdm component also needs to enable the Low and High temperature interrupts by enabling the appropriate bits in IA32_THERM_INTERRUPT MSR.

Architecture Note: We don't need [IOVM] brdm to [Domain0] brdm communication if [IOVM] brdm can access the above mentioned MSR. As a result, the step 2 and 7 in this collaboration diagram will be accomplished by [IOVM] brdm. This step will be updated once we know whether this MSR can be accessed from IOVM.

Architecture Note: Platforms need to enable the thermal monitoring interrupt in Local APIC before brdm

instantiate.

- 3. [IOVM] sod instantiates ssm component.
- 4. [IOVM] ssm sets the CPU temperature Threshold # 1 and Threshold # 2. This threshold is relative to TCC. The ssm also checks with brdm whether previous boot was due to CPU TCC.
- 5. [IOVM] ssm logs an informational MEL event in case previous boot was due to CPU TCC. This is an optional step depending on the value of the bit.
- 6. [IOVM] brdm component registers with [IOVM] sx for any CPU temperature events. If temperature interrupts can only be raised in Domain0, then [Domain0] brdm component will handle the interrupt.

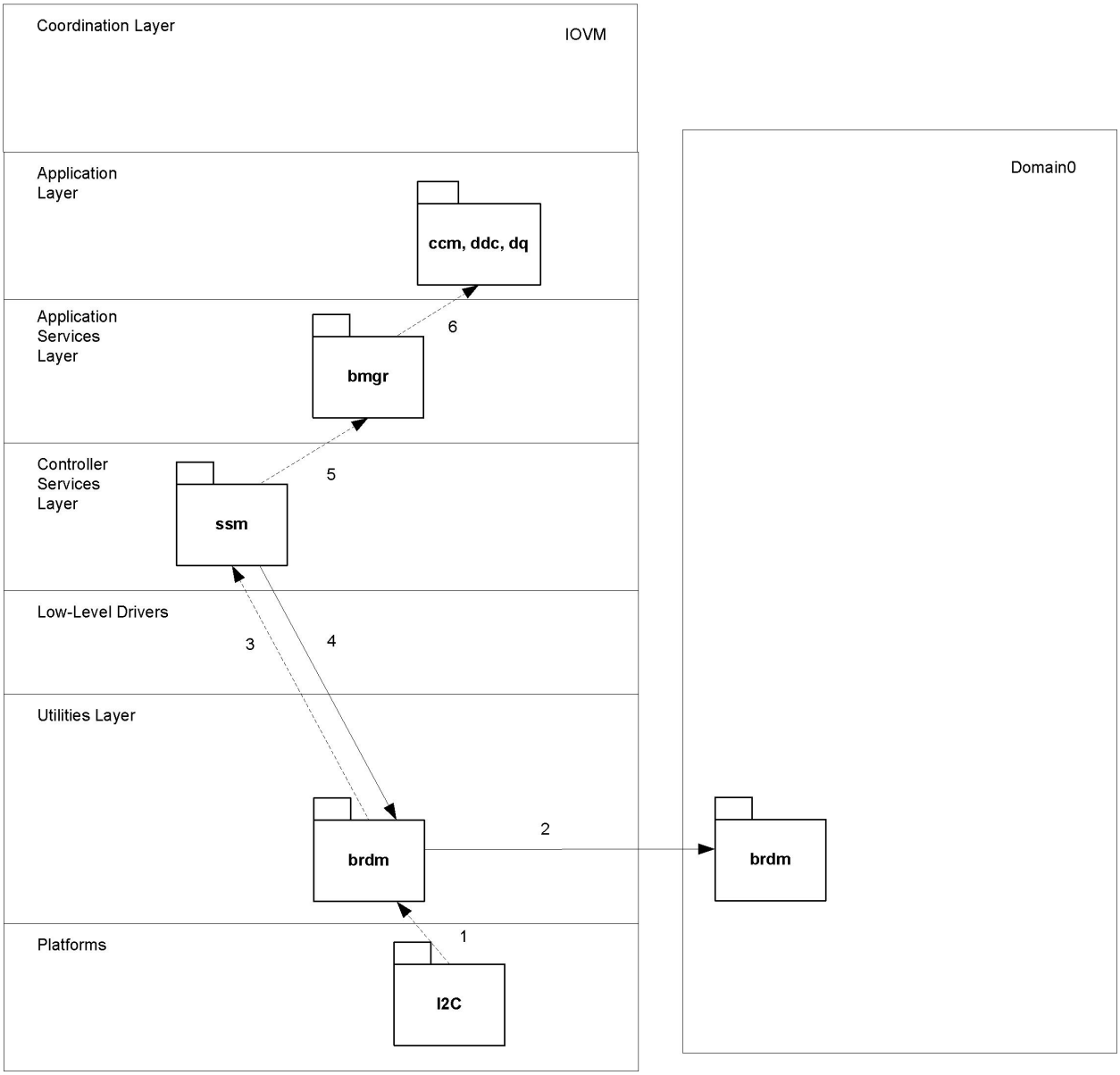
Architecture Note: This step is contingent on investigations in this area.

- 7. [IOVM] brdm communicates to [Domain0] brdm to actually set the thresholds. The [Domain0] brdm component sets the following fields in IA32_THERM_INTERRUPT register:
  - "Threshold # 1 Interrupt Enable"
  - "Threshold # 1 Value" with the value provided by [IOVM] ssm.
  - "Threshold # 2 Interrupt Enable"
  - "Threshold # 2 Value" with the value provided by [IOVM] ssm
  - "Overheat Interrupt Enable"
  - "Low Temp. Interrupt Enable"
  - "High Temp. Interrupt Enable"
- 8. [IOVM] sod instantiates bmgr component.
- 9. [IOVM] bmgr component registers with the ssm component for the CPU temperature related events.
- 10. [IOVM] sod instantiates ccm, ddc and dq components.
- 11. [IOVM] ccm, ddc and dq components register with bmgr for battery events.

3.1.1.1.12.2.CPU over temperature condition (Threshold # 1) detected

Following diagram illustrates the component collaboration to handle the CPU over temperature (Threshold # 1 and Threshold # 2) events:

Figure 13: CPU Over temperature condition detected



- 1. [IOVM] brdm or [Domain0] brdm component receives the interrupt. If it is [Domain0] brdm, then it needs to notify [IOVM] brdm using rmi.
- 2. [IOVM] brdm reads the digital temperature, in 1 deg Celsius relative to the TCC activation temperature. The [Domain0] brdm component can read the CPU register IA32_THERM_STATUS and provide following information. In addition to the information, it needs to clear various fields in this register.
  - Ensure that temperature is a valid temperature. For a valid temperature, the "Reading Valid" bit of the IA32_THERM_STATUS must be set to 1.
  - Actual temperature (aka digital readout temperature) relative to the TCC. A lower reading indicates a higher actual temperature. The "Digital Readout" temperature is the number of degrees the processor is away from hitting the activation temperature for the TCC (Thermal Control Circuit) logic. The activation temperature is not a fixed value. It does not indicate the

actual temperature, but indicates how far away the actual temperature is from hitting it.
- Whether actual temperature is currently higher than or equal to the value set in Thermal Threshold # 1 as indicated by bit "Thermal Threshold # 1 Status" in the above register.
- Whether actual temperature is currently higher than or equal to the value set in Thermal Threshold # 2 as indicated by bit "Thermal Threshold # 2 Status" in the above register.

Architecture Note: Investigate whether IOVM can access IA32_THERM_STATUS MSR. If so, then there is no need to communication between IOVM and Domain0 for this - update step 2.

- 3. [IOVM] brdm component notifies ssm about the CPU thermal monitoring interrupt.
- 4. [IOVM] ssm component calls brdm interfaces to get more information about CPU temperature.
- 5. [IOVM] ssm notifies bmgr.
- 6. [IOVM] bmgr posts a battery not usable event to registered listeners such as ccm, ddc and dq. These components will have the same behavior as on previous platforms monitoring CPU die temperature.

3.1.1.1.12.3.Abatement of CPU over temperature condition (Threshold # 1)

The collaboration for this scenario is same as described in above section with only exception to that bmgr listeners will now receive battery usable event.

3.1.1.1.12.4.CPU over temperature condition (Threshold # 2) detected

The [IOVM] ssm component will log a date stamp and the CPU temperature in a persistent Failure Analysis store on the controller associated with the over temperature CPU. It is possible that CPU can trip anytime now due to overheat. The call flow from brdm to ssm remains as described above.

Architecture Note: Assumed that [IOVM] ssm logs the date stamp on Snowmass whenever CPU hits Maximum temperature?

3.1.1.1.13.ECC error handling

TBD

3.1.1.1.14.PCI/PCIe error handling

TBD

3.1.1.1.15.Wake On LAN (WOL)

TBD

3.1.1.1.16.NTB

TBD

3.1.2.Core Assets

This section specifies the core assets affected by the feature. Each core asset is specified as its own subsection below. For those core assets that are components, diagrams may be used to illustrate new dependencies being added to an existing component or a diagram may also be used to illustrate how any new component(s) fit into the existing component layer(s) as well as their dependencies.

Any new core assets or updates to the overall CFW component dependency model will result in corresponding updates to the Controller Firmware Architecture Specification.

Also specified in this section will be which components will contain new variation points and a brief description why and how the variation points are being implemented.

The **Core Application Services** core asset team is responsible for coordinating implementation of this feature and submitting the Feature Request CR if one hasn't already been submitted.

3.1.2.1.Core Application Services

3.1.2.1.1.[IOVM] brdm - Board Manager

The brdm component is extended to:

- Construct baseboard ID for the local board and provide this to alternate controller on request.
- Provide the hardware and the constructed baseboard ID, and host board ID to higher level components.
- Retrieve the information about SATA flash drives from XenStore and provide it to hcvh when requested.
- Disable USB port.

3.1.2.1.2.[Domain0] brdm - Board Manager

The brdm component is extended to:

- Provide DIMM SPD data to [Domain0] hcvh.
- Notify [IOVM] brdm about the CPU thermal monitoring interrupts.

3.1.2.1.3.[IOVM] ssm - Sub System Monitor

Following changes are expected in ssm component in IOVM:

- The ssm component will need to be enhanced to provide support for the new HICs.
- The ssm component will need to read the Host Board ID information from brdm to set appropriate parameters for the new HICs.
- The SSMHostCardType needs to add any additional host card type as part of the supported host cards.
- The ssm component will also need to report the new host board correctly in the object graph.
- The ssm component will need to get the SATA flash disk information from [IOVM] brdm component and report into object graph as described in "Management Interface" section of SATA Disk Element ER. In brief, it will use existing cache backup device SYMbol data structure to report.
- The ssm component needs to add the necessary implementation for the Pikes Peak controller as described in component location FFD.
- The ssm component logs a critical MEL event and raises needs attention condition for invalid SATA flash disk configurations.
- The ssm sets the CPU temperature Threshold # 1 and Threshold # 2 as described in Section 3.1.1.1.12. CPU Temperature Monitoring.
- Log "CPU Thermal Control Circuit Activation" MEL event during SOD.

The ssm component needs to re-evaluate the use of following gears variation for Pikes Peak. It is assumed that no major changes are required in existing implementation.

- APP_SnowmassEnclosure,

- APP_MercuryDevilsEnclosure,
- APP_CamdenEbbetsSupport,
- APP_QuadPort1GbiSCSIHIC

3.1.2.1.4.[IOVM] hcvh - Hardware Configuration Validation Handler

The hcvh component must be extended to:

- Perform vendor pairing as part of SBB validation. For vendor pairing, the local and alternate hardware baseboard ID is sufficient.
- The hcvh component needs to incorporate the changes in PS SBB validation as described in Section 3.1.1.1.4. SBB Validation.
- Perform PS SBB validation as described in Section 3.1.1.1.6. Power Supply SBB validation and PMBus support.
- Perform DIMM verification in instantiate phase. Get the DIMM SPD data from [Domain0] brdm via [Domain0] hcvh.
- Perform SATA disk configuration validation as described in Section 3.1.1.1.11.4. Invalid SATA flash disk configurations.
- Provide an interface to ssm and pbm to report SATA disk configuration validation results.
- The hcvh component needs to re-evaluate the use of following gears variation HW_SupportsUserDiagnostics, and HW_HostCardReplaceable for Pikes Peak. It is assumed that no major changes are required in existing implementation.

3.1.2.1.5.[Domain0] hcvh - Hardware Configuration Validation Handler

- Provide an interface to [IOVM] hcvh to return the DIMM SPD data.

3.1.2.2.Diagnostic Services

3.1.2.2.1.[IOVM] bcdm - Base Controller Diagnostics Manager

The bcdm component needs to revisit the use of following gears variation for Pikes Peak. It is assumed that no major changes are required in existing implementation.

- APP_HICDiagnostics
- APP_BaseControllerDiagnostics

3.1.2.3.Firmware Architecture

3.1.2.3.1.VariationMgmt – Variation Management Tools

3.1.2.3.1.1.Gears Variable

3.1.2.3.1.1.1.hostboard::Supported Boards

Enumeration `HB_SupportedBoards` will be updated for the new HICs as following:

`HB_0401` - Dual-port 40 Gbps InfiniBand HIC (Elk Park)

`HB_0801` - Quad-port 8 Gbps FC HIC (Manitou)

`HB_0101` - Dual-port 10 Gbps iSCSI HIC (Glen Cove)

3.1.2.3.1.2.Module and Mixin:

3.1.2.3.1.2.1.Hostboard

The hostboard mixin is used to define the feature model values for Host Interfaces supported by the baseboard and its host interface cards.

Following is the host board definition file for PikesPeak_MixedHIC:

```
/*
    Context: Pikes Peak baseboard with mixed HIC support.
    HostInterface: built in FC   - 4 ports
    HostInterface:  one of the following (detected at start up):
     - Elk Park 40 Gb IB          - 2 ports
     - Manitou 8 Gb FC           - 4 ports
     - Glen Cove 10 Gb iSCSI      - 2 ports
*/
HB_SupportedBoards =
{
   HB_None;
   HB_0801;
   HB_0401;
   HB_0101;
}
HB_MaxHostPorts
{
   FC = 8;
   IB = 2;
   iSCSI = 2;
   All = 8;
}
HB_SupportedSpeeds
{
InfiniBand =
{
```

```
40 Gb;
}
FC =
{
8 Gb;
}
iSCSI =
{
10 Gb;
}
}
HB_CacheAlignment = 8;
HB_MediaTransportProtocols =
{
   FibreChannel;
InfiniBand;
Ethernet;
}
HB_SCSItransportProtocols =
{
   FCP;
  SRP;
iSCSI;
}
HB_BinaryRequired =
{
ibHcaFw;
BladeEng2;
}
HB_LCLRequired = true;
HB_T10PI = true;
```

HB_Infusion = false;

HB_MixedHIC = true;

In addition to above, the IOVM needs to define new elements src_IB_mthca and src_IB_mlx under ib_hw directory. One would map to Arbel chip and other would map to connectX chip.

3.1.2.3.1.3.Recipe Process

The following table specifies recipe changes under ib_hw component.

Asset Recipe Variation	Element	Repository Mapping	Element Recipe Variation
None	src	Application/RAID/Infiniband/ib_hw/src_IB_mthca	None
	src	Application/RAID/Infiniband/ib_hw/src_IB_mlx	None

3.1.2.4.Foundations 1

3.1.2.4.1.[IOVM] SYMbol API

The SYMbol API changes related to {insert feature name} are summarized in the {insert FFD name including FFD at the end}, document number {insert FFD doc number}. Details of these changes are in or will be included in the SYMbol Specification – Internal Master, document number 349-1051890.

The SYMbol API is represented in an XML data structure, from which both the SYMbol specification and the public API file SYMbolAPI.x are generated. The SYMbolAPI asset does not directly contain this XML content, but rather it contains a generated SYMbolAPI.x file for each of the currently supported feature sets on the RAIDCore trunk. The Gears api.target variable is used to select the appropriate SYMbolAPI.x file for the feature set associated with the product being built.

The variation mechanism used to determine the generated content of the feature-set-specific SYMbolAPI.x files is outside the scope of this FAM. However, this FAM does specify the Gears variables defined in the feature model that allow the firmware assets to vary as needed to adapt to the contents of the feature-set-specific SYMbolAPI.x selected for the product being built. These variables are defined in section Section 3.1.2.3.1.1. Gears Variable and their use is noted throughout the FAM as appropriate.

- The `InterfaceSpeed` enumeration in SYMbol needs to be extended to add `SPEED_40GIG` for 40 Gbps IB.
- New RecoveryFailureType REC_INVALID_SATA_FLASH_CONFIGURATION and corresponding FailureTypeEntry must be added. The FailureTypeEntry is of type ControllerRef.

3.1.2.4.2.[IOVM] Meldb – Major Event Log Database

A new event group “SBB Validation” needs to be added.

3.1.2.4.2.1.Invalid SATA flash disk configurations

MEL Data Name	MEL Data Content
Event Name	Invalid SATA flash disk configuration
Event Group	Persistent Cache Backup Events

MEL Data Name	MEL Data Content
Event Priority	Critical
Log Group	Controller
Event Category	Error
Event Component Type	Cache Backup Device (MEL_COMP_CACHE_BACKUP_DEVICE)
Sense Key	None
ASC/ASCQ	None
Event Specific Data	Flash slot # related to invalid configurations

Architecture Note: Do we need to create a new "Event Group" instead of putting this under "Persistent Cache Backup Events" group?

The [IOVM] ssm component is responsible for logging this mel event.

3.1.2.4.2.2.CPU Thermal Control Circuit activated

MEL Data Name	MEL Data Content
Event Name	CPU thermal control circuit activated
Event Group	Sub System Monitor
Event Priority	Informational
Log Group	Controller
Event Category	Error
Event Component Type	Controller (MEL_COMP_CONT)
Sense Key	None
ASC/ASCQ	None
Event Specific Data	None

The [IOVM] ssm component is responsible for logging this mel event.

3.1.2.5.Foundations 2

3.1.2.5.1.[Domain0] vmmgr - Virtual Machine Manager

- The vmmgr component is extended to
- Incorporate the changes required for SBB validation as described in Section 3.1.1.1.4. SBB Validation.

3.1.2.5.2. [Domain0] olm - OSA Lockdown Manager

- This component registers with the vmmgr for lockdown notifications, especially lockdown due to SBB validation.
- Register with vmmgr to be notified whenever it receives keep alive message from alternate. Refer to Section 3.1.1.1.5.3. Scenario 3 (Alternate controller is physically in-place but not responding in a timely manner) and Section 3.1.1.1.5.5. Scenario 5 (Alternate controller is not in-place but inserted later) for more information.
- Handle special lockdown scenario. This lockdown is no different from existing lockdowns except that [Domain0] vmmgr will wait for a keep alive message from alternate controller.

3.1.2.5.3.[Domain0] fpmgr - Flash Partition Manager

The fmgr is a new component introduced in Domain0 to provide required interfaces to SATA flash. This component is responsible for detecting a valid SATA flash drive, formatting the layout partition it if it is a new drive, and providing information about SATA flash drive to clients as requested. This component resides at user-level. This component is also responsible to call [Domain0] NV interfaces to store some of the meta-data in controller NVSRAM to associate the flash drive with the controller.

3.1.2.5.4.[IOVM] cmgr – Controller Manager

- The cmgr component will report different Model Names into the object graph for different HICs in Pikes Peak. A new combination matching the Board ID (5268 for PP200 and 5468 for PP400) and the Host Board IDs (0801 for FC, 0101 for iSCSI, and 0401 for IB) will need to be added.
- The cmgr component needs to revisit the use of following gears variation, HW_FieldUpgradeableMemory, for Pikes Peak. It is assumed that no major changes are required in existing implementation.

3.1.2.5.5.[IOVM lem - Lockdown Error Manager

The lem component is extended to

- Incorporate the changes required for SBB validation as described in Section 3.1.1.1.4. SBB Validation.
- Handle the controller type mismatch scenarios as described in Section 3.1.1.1.5. Controller Type Mismatch.
- Provide an interface to olm component to be notified whenever there is a keep alive message from alternate controller. Refer to Section 3.1.1.1.5.3. Scenario 3 (Alternate controller is physically in-place but not responding in a timely manner) and Section 3.1.1.1.5.5. Scenario 5 (Alternate controller is not in-place but inserted later) for more information.
- Notify [Domain0] olm for the special lockdown.
- Remove vendor pairing part of SBB validation.

3.1.2.6.IO Interfaces 1

3.1.2.6.1.[IOVM] ioni - I/O Network Interface

The function enableHostInterfaces gets called as part of pre-initialization phase during IOVM SOD. The internal diagnostics are executed as part of this function. The ioni component needs to revisit the use of following gears variation for Pikes Peak. It is assumed that no major changes are required in existing implementation.

- APP_HICDiagnostics

3.1.2.7.IO Interfaces 3

3.1.2.7.1.[IOVM] isni - iSCSI Network Interface

New interfaceErrorType values will be defined for use with the isni::MtlEvents::interfaceError() interface to indicate new types of BE2 exceptions, some of which may be associated with problems on the Ethernet PCI function.

- The ioni::Lld::runInternalDiagnostic interface will be added to isni::MtlInterface.

3.1.2.7.2.[IOVM] b_isn - Breckenridge iSCSI network Manager

- Enhance to handle new isni::MtlEvents::interfaceError events. Add to existing MEL data, etc (this assumes that the existing NAS VM's syslog will contain information about the error from that point of view).
- Handle the general aspects of chip errors, including logging and handling the Ethernet aspects of the error.
- The ioni::Lld::runInternalDiagnostic interface will be implemented by b_isn::IscsiNetworkManager. It will relay the request to the mtlsebe component.
- Relay the HIC diagnostics controls from ioni to mtlsebe and handle status retrieval. – note this is already done.

3.1.2.7.3.[IOVM] mtlsebe

This code for this component is provided by ServerEngines and is integrated into the LSI source code control system in the same manner as is currently used. Although there is no currently-released program which uses this component, there has been much work done to date on potential I+T products which use this component. That work is leveraged here. The additional work required within the mtlsebe component for the Glen Cove HIC is divided into that performed by LSI and that performed by ServerEngines.

LSI

- Integrate code drops.
- First LSI-to-SE drop may require changes to the export script.
- Apply appropriate variation controls to makefiles.

ServerEngines

- Detect new error conditions, as appropriate, and use the isni::MtlEvents::interfaceError() interface with the new interfaceErrorType values.
- Implement all BE2 diagnostics as described above.

3.1.2.7.4.[IOVM] be2nic

This is a new component for the Glen Cove HIC which is provided by ServerEngines. It represents the binary Debian linux driver for the version which is used on the NAS VM. ServerEngines will make changes to the standard linux driver, as required, for it to operate in the Xen environment used by the Orion program. It must also support the NAS VM's requirement to implement LACP (with aggregated MACs) in its linux Ethernet stack.

3.1.2.8.IO Interfaces 4

3.1.2.8.1.[IOVM] ib_hw – IB Hardware Specific Driver

Currently, controller firmware has IB driver which is based on OFED-1.2. The existing IB driver can only support Tavor and Arbel IB chips. The Pikes Peak controller will have next generation of InfiniBand chip "ConnectX" and as a result OFED-1.3 or higher version (Pick latest GA release) of InfiniBand driver is required. There might be some customization required for this open source driver to work on Pikes Peak controller. The component also needs to accommodate MAPI related changes due to introduction of connectX based chip. It is also required to change the src directory under Images/ibHcaFw to bring in the images for connectX chip.

3.1.2.8.2.[IOVM] ib_core – InfiniBand Core Driver

This piece also comes from OFED, but this is mainly to deal with RDMA part of communication. With the upgrade in OFED version, this component is also required to be changed for Pikes Peak. The component also needs to accommodate MAPI related changes due to introduction of connectX based chip.

3.1.2.8.3.[IOVM] ibHcaFw – InfiniBand HCA Firmware Image

A new image for ConnectX chip is required which needs to be checked during SOD for any upgrade.

3.1.2.8.4.[IOVM] LCL – Linux Compatibility Layer

The LCL component needs to accommodate linux kernel-2.6 related constructs used by OFED-1.4 or higher. Current LCL is linux kernel-2.4 based.

3.1.2.8.5.[IOVM] STP_SRP – SCSI RDMA Protocol

SRP protocol interacts with SCSI driver on one side and InfiniBand driver on the other side. With the upgrade of InfiniBand driver, this component might require some changes. In particular, loopback diagnostic under mtfe needs to incorporate diagnostics for ConnectX.

3.1.2.9.Platforms

3.1.2.9.1.[IOVM] BCM – Board Configuration Module

The Pikes Peak variant of the modelConfig table in BCM needs to be updated to handle the new HICs. The BCM component needs to align itself with Zebulon FPGA for the quick switches.

3.1.2.9.2.[IOVM] common – Common PCI Definitions

New PCI IDs will be added in cmnPCI.h for the new HICs. New enumeration values also need to be defined for PCI_VENDOR_DEVICE_ID for the new HICs.

3.1.2.9.3.[IOVM] Diag – Diagnostic Platform Module

The diagnostics for various HICs will not be integrated into the Diag module. The manufacturing plan is for the ODM to use chip specific code from chip vendor on a PC to test the HICs, using a PCIe-to-HIC adapter to plug the HIC into the PC. The diagnostic test will run on the PC and test the HIC in isolation, with no Pikes Peak controller involved in the process. No component diagnostic will be added to Diag for various HICs.

3.1.2.9.4.[IOVM] mpm - Midplane Manager

The mpm component needs to align itself with the quick switches for SBB validation in Zebulon FPGA.

3.1.2.9.5.[IOVM] pwsplm - Power Supply Manager

The pwsplm component needs to be enhanced to support PMBus protocol for the power supplies in ESG enclosures Camden/Ebbets and Wembley SAS enclosures housing Pikes Peak controller. Refer to Section 3.1.1.1.6. Power Supply SBB validation and PMBus support for more information.

3.1.2.9.6.BIOS

Following changes are expected in BIOS:

- Display different 7-segment codes for controller boot using PXE, USB and SATA flash. Also, specify error codes for errors during these boots.
- BIOS must boot from the SATA flash disk on slot-0 if both the slots are populated. Display a boot error in 7-segment if it is not able to boot from disk in slot-0. It is possible that disk in slot-1 has valid firmware.
- BIOS to validate some kind of signature in SATA flash disk, USB drive before it allows controller to boot from these devices.

3.1.2.10.Volume IO Services

3.1.2.10.1.[IOVM] ccm - Cache Configuration Manager

The ccm component needs to revisit the use of following gears variation for Pikes Peak. It is possible that existing behavior continues.

- HW_FieldUpgradeableMemory

3.1.2.10.2.[IOVM] rpa - RAID Parity Assist

The rpa component needs to revisit the use of following gears variation for Pikes Peak. It is assumed that no major changes are required in existing implementation.

- HW_FieldUpgradeableMemory

3.1.2.10.3.[IOVM] cache - Cache Management

The cache component needs to revisit the use of following gears variation for Pikes Peak. It is assumed that no major changes are required in existing implementation.

- HW_FieldUpgradeableMemory

3.1.2.10.4.[IOVM] pbm - Persistent Backup Manager

The pbm component needs to get the information about available SATA flash disk from [IOVM] brdm component. The pbm component needs to call [IOVM] hcvh component to get the status of SATA flash disk configurations. In case of invalid SATA flash disk configurations, switch write-back caching with mirroring to write-through caching. Rest flow remains same as on existing platforms.

3.1.2.11.Hypervisor

3.1.2.11.1.[Domain0] ivmhb - Inter VM Heartbeat Manager

- This component needs to provide an interface to vmmgr to be notified after SBB validation.

3.1.2.11.2.XenStore Key/Values

The new additions are marked in bold.

Table 14: XenStore Key/Values

Xenstore Key	**Potential Values**	**Initiator**	**Target (Registrant)**
/OSA/IOVM/InitializationState	INIT = 0	[Domain0] VMMGR via XSMGR	
	SBB_VALIDATION_SUCCESSFUL = 1	[IOVM] HCVH via XSMGR	[Domain0] VMMGR via XSMGR
	SBB_VALIDATION_INVALID = 2	[IOVM] LEM via XSMGR	[Domain0] VMMGR via XSMGR
	ALT_BOARD_ID_TIMEOUT = 3	[Domain0] VMMGR via XSMGR	[IOVM] HCVH via XSMGR
	ALT_BOARD_ID_COMPLETE = 4	[Domain0] VMMGR via XSMGR	[IOVM] HCVH via XSMGR
	ACS_REQUIRED = 5		
/OSA/IOVM/SATAFlashInquiryInfo	TBD	[Domain0] VMMGR via XSMGR	[IOVM] HCVH via XSMGR