# SSI System Management Guide

*September 2009*

*Revision 1.0.0*

### *Important Information and Disclaimers:*

1. THE SERVER SYSTEM INFRASTRUCTURE PROMOTERS (“SSI PROMOTERS”) MAKE NO WARRANTIES WITH REGARD TOTHIS SSI SPECIFICATION (“SPECIFICATION”), AND IN PARTICULAR DOES NOT WARRANT OR REPRESENT THAT THIS SPECIFICATION OR ANY PRODUCTS MADE IN CONFORMANCE WITH IT WILL WORK IN THE INTENDED MANNER. NOR WILL SSI PROMOTERS ASSUME ANY RESPONSIBILITY FOR ANY ERRORS THAT THE SPECIFICATION MAY CONTAIN OR HAVE ANY LIABILITIES OR OBLIGATIONS FOR DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, INCIDENTAL, INDIRECT, PUNITIVE, OR CONSEQUENTIAL DAMAGES WHETHER ARISING FROM OR IN CONNECTION WITH THE USE OF THIS SPECIFICATION IN ANY WAY.

2. NO REPRESENTATIONS OR WARRANTIES ARE MADE THAT ANY PRODUCT BASED INWHOLE OR PART ON THE ABOVE SPECIFICATION WILL BE FREE FROM DEFECTS OR SAFE FOR USE FOR ITS INTENDED PURPOSE. ANY PERSON MAKING, USING OR SELLING SUCH PRODUCT DOES SO AT HIS OR HER OWN RISK.

3. THE USER OF THIS SPECIFICATION HEREBY EXPRESSLY ACKNOWLEDGES THAT THE SPECIFICATION IS PROVIDED AS IS, AND THAT THE SSI PROMOTERS MAKE NO REPRESENTATIONS, EXTENDS NO WARRANTIES OF ANY KIND EITHER EXPRESS OR IMPLIED ORAL OR WRITTEN, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, OR WARRANTY OR REPRESENTATION THAT THE SPECIFICATION OR ANY PRODUCT OR TECHNOLOGY UTILIZING THE SPECIFICATION OR ANY SUBSET OF THE SPECIFICATION WILL BE FREE FROM ANY CLAIMS OF INFRINGEMENT OF INTELLECTUAL PROPERTY, INCLUDING PATENTS, COPYRIGHTS AND TRADE SECRETS NOR DO THE SSI PROMOTERS ASSUME ANY OTHER RESPONSIBILITIES WHATSOEVER WITH RESPECT TO THE SPECIFICATION OR SUCH PRODUCTS.

4. A NON-EXCLUSIVE COPYRIGHT LICENSE IS HEREBY GRANTED TO REPRODUCE THIS SPECIFICATION FOR ANY PURPOSE PROVIDED THIS “IMPORTANT INFORMATION AND DISCLAIMERS SECTION (PARAGRAPHS 1-6) IS PROVIDED IN WHOLE.

5. UPON REQUEST FROM AN ADOPTER, THE SSI PROMOTERS WILL GRANT A NON-EXCLUSIVE, WORLD-WIDE LICENSE UNDER ANY NECESSARY CLAIMS, DEFINED IN THE ADOPTERS AGREEMENT, TO MAKE, HAVE MADE, USE, IMPORT, SELL, OFFER TO SELL, AND OTHERWISE DISTRIBUTE AND DISPOSE OF COMPLIANT PORTIONS, DEFINED IN THE ADOPTERS AGREEMENT, PROVIDED THAT SUCH LICENSE NEED NOT EXTEND TO ANY PART OR FUNCTION OF A PRODUCT IN WHICH A COMPLIANT PORTION IS INCORPORATED THAT IS NOT ITSELF PART OF THE COMPLIANT PORTION. SUCH LICENSE WILL BE GRANTED ON REASONABLE AND NONDISCRIMINATORY (“RAND”) TERMS AND MAY BE CONDITIONED UPON ADOPTER’S GRANT OF A RECIPROCAL LICENSE TO THE SSI PROMOTERS.

6. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY OTHER INTELLECTUAL PROPERTY RIGHTS IS GRANTED.

Revision 1.0.0

# Contents

## Figures

## Tables

# Revision History

The following table lists the revision schedule based on revision number and development stage of the product.

| Revision | Project Document State | Date |
|---|---|---|
| 1.0.0 | Initial public release. | 9/20/2009 |

Note: Not all revisions may be published.

This page intentionally left blank

# 1 Introduction

The intent of this document is to provide an overall systemic view of an SSI system from a management perspective. Other specifications will be referenced in this document. The adherence to SSI compliance is governed by those specifications.

## 1.1 Terms and Abbreviations

The following terms and acronyms are used in specific ways throughout this document:

### Table 1-1. Terms and Abbreviations

| Term | Definition |
|---|---|
| Base Management Interface (BMI) | This is the IPMB-based management interface used by the Chassis Manager to communicate with the blade management controllers. |
| blade | This is resource module that plugs into the blade chassis. A blade can provide many different types of resources to the chassis, including compute functions, storage capabilities, additional I/O interfaces and switching capabilities and special purpose capabilities. A blade can be a single-wide module (assumed) or a double-wide module, occupying two adjacent slots in the chassis. |
| Blade server | A system comprised of a chassis, chassis resources (power, cooling, Chassis Manager), compute blades, and communication (switch) blades. The chassis may contain additional modules, such as storage. |
| CFM | Cubic Feet per Minute. A measure of volumetric airflow. One CFM is equivalent to 472 cubic centimeters per second. |
| chassis | The mechanical enclosure that consists of the mid-plane, front boards, cooling devices, power supplies, etc. The chassis provides the interface to boards and consists of the guide rails, alignment, handle interface, face plate mounting hardware, and mid-plane interface. |
| Chassis Management Module (CMM) | Dedicated intelligent chassis module that hosts the Chassis Manager functionality. |
| Chassis Manager (CM) | Set of logical functions for hardware management of the chassis. This may be implemented by one or more dedicated Chassis Management Modules or by one or more blade management controllers and/or payload processors. |
| Chassis Management Controller (CMC) | Set of logical functions for hardware management of the chassis, implemented by one or more blade management controllers and/or payload processors. |
| cold start | Cold start is the time when blades receive the payload power for the first time. |
| face plate | The front-most element of a PBA, perpendicular to the PBA, that serves to mount connectors, indicators, controls and mezzanines. |
| Intelligent Platform Management Bus (IPMB) | IPMB is an I$^2$C-based bus that provides a standardized interconnection between managed modules within a chassis. ftp://download.intel.com/design/servers/ipmi/ipmb1010ltd.pdf |

| Term | Definition |
|---|---|
| Intelligent Platform Management Interface (IPMI) | IPMI v2.0 R1.0 specification defines a standardized, abstracted interface to the platform management subsystem of a computer system. ftp://download.intel.com/design/servers/ipmi/IPMIv2_0rev1_0.pdf |
| interconnect channel | An interconnect channel is comprised of two pairs of differential signal. One pair of differential signals for transmit and another pair of differential signal for receive. |
| managed module | Any component of the system that is addressable for management purposes via the specified management interconnect and protocol. A managed module is interfaced directly to the chassis BMI. |
| Management Controller | An intelligent embedded microcontroller that provides management functionality for a blade or other chassis module. |
| may | Indicates flexibility of choice with no implied preference. |
| mezzanine | The mezzanine is a PBA that installs on a blade PBA horizontally. It provides additional functionality on the blade PBA and provides electrical interface between the blade PBA and the mid-plane PBA. Both the blade PBA and mezzanine PBA are contained inside the blade module. |
| mid-plane | Equivalent to a system backplane. This is a PBA that provides the common electrical interface for each blade in the chassis and on both the front and back of the PBA. |
| module | A physically separate chassis component that may be independently replaceable (e.g., a blade or cooling module) or attached to some other component (e.g., a mezzanine board). |
| open blade | A blade that conforms to the requirements defined by the Open Blade standard set of specifications. |
| out-of-band (OOB) | Communication between blades that does not need the host or payload to be powered on |
| payload | The hardware on a blade that implements the main mission function of the blade. On a compute blade, this includes the main processors, memory, and I/O interfaces. The payload is powered separately from the blade management subsystem. Payload power is controlled by the blade management controller. |
| Peak power | The peak power drawn by a blade is defined as the maximum power that a blade can draw for a very short period of time during a hot insertion, hot removal, or during a cold start. |
| Primary Channels | The 2 Baseboard Ethernet connections that every server blade must have that goes to the Primary Switch(es).  These primary Ethernet channels are also shared on the blade for management traffic. |
| Primary Switch | The switch connected to the primary channels |
| SAP | Service Applet Package |
| **shall** | Indicates a mandatory requirement. Designers must implement such mandatory requirements to ensure interchangeability and to claim conformance with this specification. The use of **shall not** (in bold) indicates an action or implementation that is prohibited. |
| **should** | Indicates flexibility of choice with a strongly preferred implementation. The use of **should not** (in bold text) indicates flexibility of choice with a strong preference that the choice or implementation be avoided. |
| slot | A slot defines the position of one blade in a chassis |

| Term | Definition |
|---|---|
| U | Unit of vertical height defined in IEC 60297-1 rack, shelf, and chassis height increments. 1U=44.45 mm. |
| WDT | Watchdog timer |

## 1.2 Reference Documents

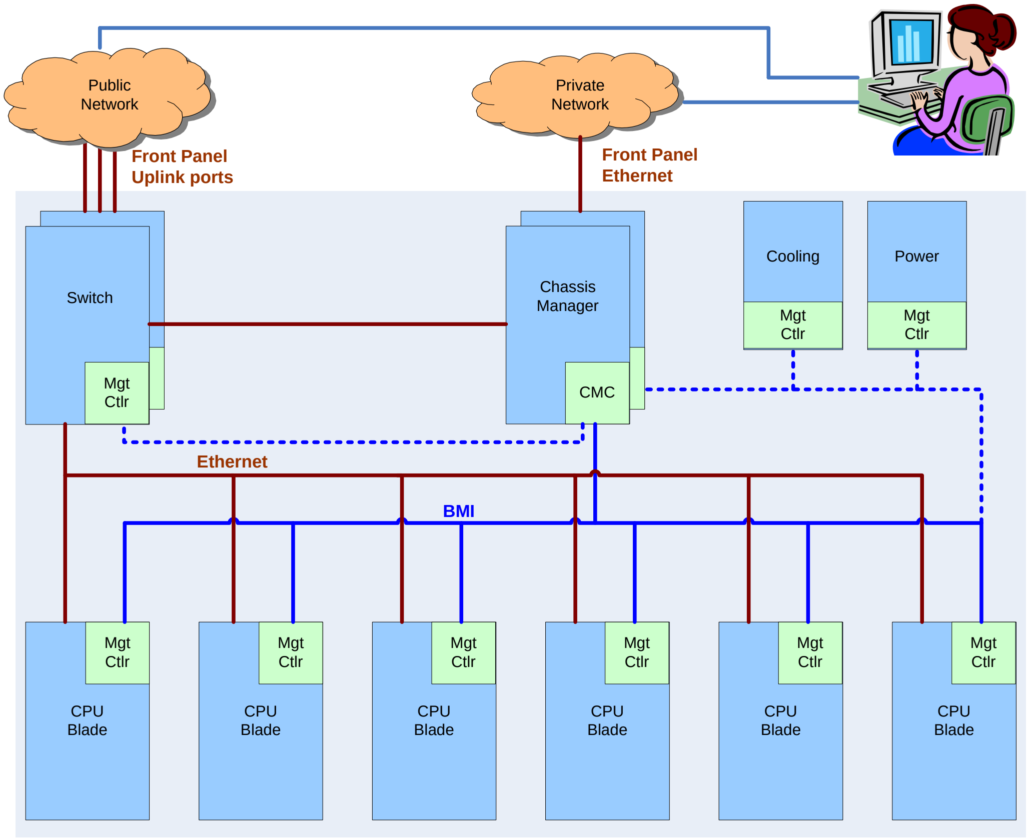
- SSI Compute Blade Specification
- SSI Chassis Management Module (CMM) Specification
- SSI Switch Base Specification
- SSI Switch VPD Specification
- SSI Switch Management Specification
- IPMI – Intelligent Platform Management Interface Specification, v2.0 rev 1.0E3, February 16, 2006, Copyright © 2004, 2005, 2006 Intel Corporation, Hewlett-Packard Company, NEC Corporation, Dell Inc., All rights reserved.
- IPMI – Platform Management FRU Information Storage Definition, V1.0, Document revision 1.1, September 27, 1999 Copyright © 1998, 1999 Intel Corporation, Hewlett-Packard Company, NEC Corporation, Dell Inc., All rights reserved.
- IPMI – Intelligent Platform Management Bus Communications Protocol Specification, V1.0, Document revision 1.0, November 15, 1999 Copyright © 1998, 1999 Intel Corporation, Hewlett-Packard Company, NEC Corporation, Dell Inc., All rights reserved.
- I2C Bus Specification Version 2.1 or later from NXP Semiconductors: This can be found at: http://www.nxp.com/acrobat_download/literature/9398/39340011.pdf
- SSI Mezzanine Card specification.

# 2 General SSI Blade System Architecture

## 2.1 Topology

Figure 2-1 illustrates the schematic for a typical blade system.

**Figure 2-1. Typical Blade System Management Interface Diagram**

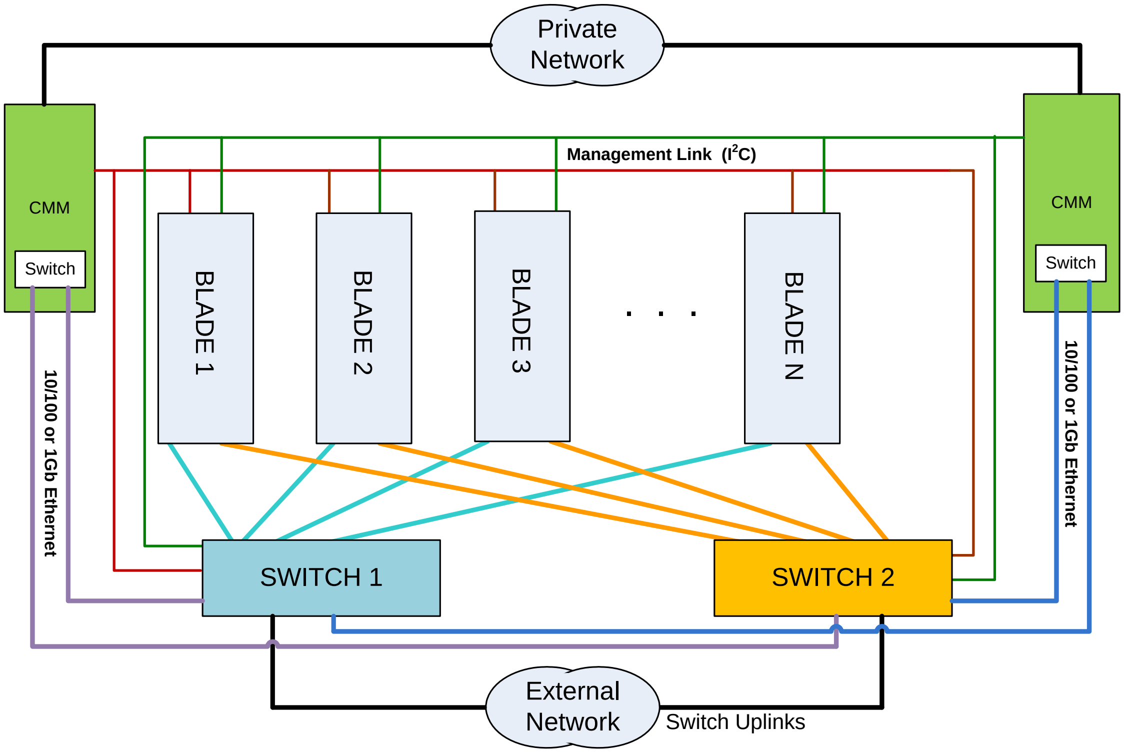


There are many different blade server product implementations. Cost, performance, ease of use, and high availability are a few key differentiators. Most bladed systems have a few things in common: all include a chassis that holds the servers or compute blades, communications switches, and integrated power and cooling sub-systems. Some include special function blades or modules, such as a load balancer blade or storage modules, and all bladed systems contain an interconnection network.

This physical interconnection network is called a mid-plane. Generally, the front side houses the compute blades, while the communication switches and other I/O modules are placed in the back.

For management connectivity, all of the blades are connected to two switches and to chassis management modules (CMMs) as shown in Figure 2-2. The blades are connected to the switches using both the First Primary and the Second Primary interconnect channels (Ethernet interfaces). These primary interconnect channels are also used for standard payload traffic to the blades.

### Figure 2-2. Blade Server System Block Diagram

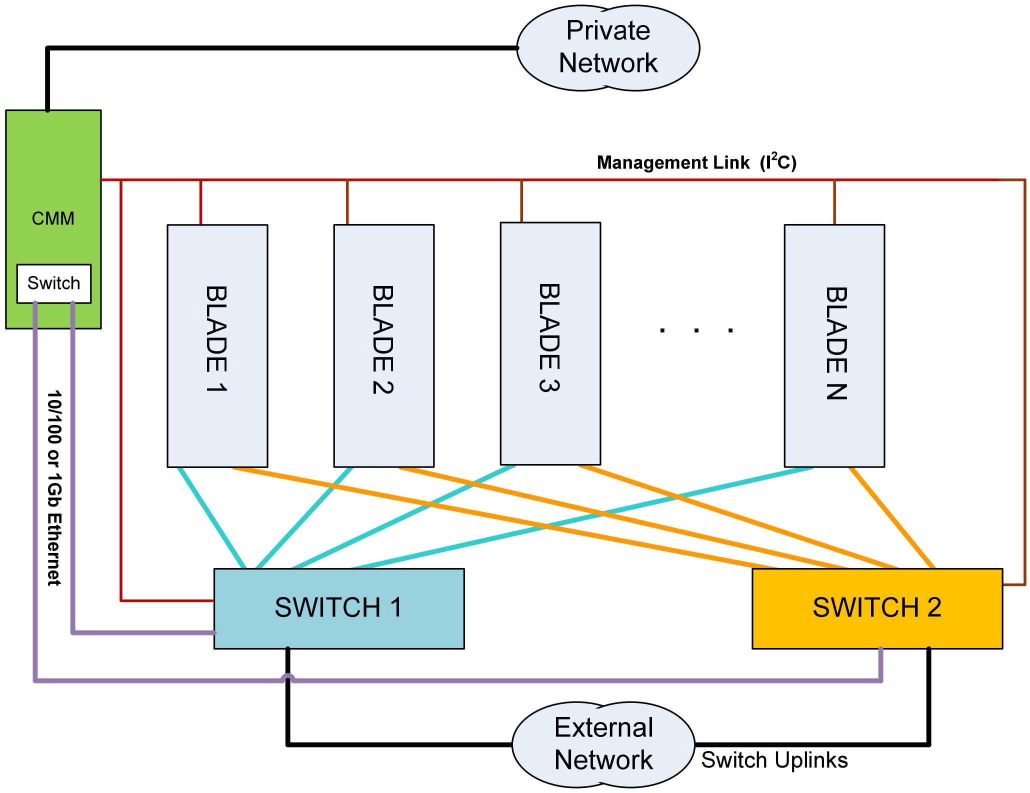


**Blade System Block Diagram (Dual Switch – Redundant Management)**

The configuration shown here is generally used in enterprise and high-availability systems in which redundancy is required. In this topology, multiple paths are available for the blades to communicate with Chassis Managers.

For cases in which redundancy is not of prime importance, such as in a small business, a simpler, non-redundant configuration can be used as shown in Figure 2-3. The second management module and all of the associated connections are not implemented or not used.

### Figure 2-3. Blade System with Non-redundant Management

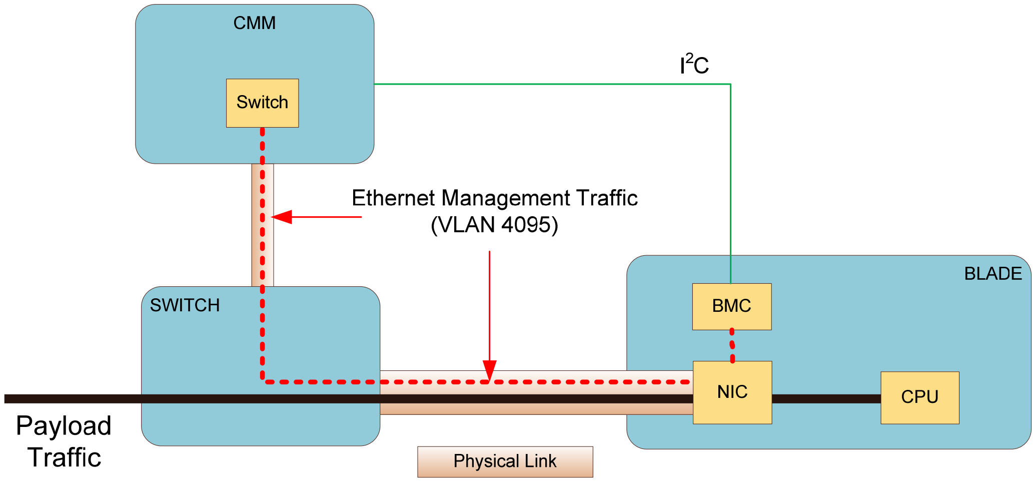


**Blade System Block Diagram**
**(Dual Switch – Non Redundant Management)**

The blades and the switches are connected to management modules using two distinctly different interconnect technologies - Ethernet and I$^2$C. Ethernet is used for IPMI Management, keyboard, video, and mouse (KVM) data from the blades to the management console. The other management interconnection is a slower speed serial connection (I$^2$C (BMI), as defined in the SSI Compute Blade Specification*)* between each blade and switch to the CMM. This interconnection is used primarily for power up, discovery, configuration, and asset management. Once the Ethernet interface is configured, the blade must be able to receive IPMI messages through both the BMI and Ethernet management interfaces.

As shown in Figure 2-4, for the Ethernet interface between the CMM and the management interface to the Primary Switch, the CMM must tag all packets with VLAN 4095; it **should not** be assumed that the Primary Switch tags these packets on ingress to the switch. In addition to the management interfaces, each of the interfaces in the switch that are connected to blades are members of VLAN 4095, therefore the CMM traffic can reach the blade NICs. The Blade NICs will then forward the appropriate traffic to the BMC through a sideband (e.g.: RM-II ) interface. Likewise, the BMC must tag all packets as 4095 to ensure that the management traffic (responses and asynchronous communications) reaches the CMM.

### Figure 2-4. Management Traffic Flow using a private VLAN



Management Traffic Flow

The Chassis Manager performs the discovery of the different components (subsystems) in the chassis, and it configures the blades, switches, and modules according to the policy set by the (remote) operations center. The Chassis Manager establishes the system configuration during power up, ensures compatibility among the blades and the switches, monitors system health, performs system error logging and reporting, and manages the overall system power and thermal capabilities.

The *SSI Switch Base Specification* has a provision for adding an additional VLAN for host (payload) in-band management as well. This could be used to manage the OS running on a blade.

In addition to the two Primary Ethernet Switches that also carry blade management traffic, a blade system may also have additional fabric switches that do not carry blade management traffic (e.g., Fibre Channel). In this case, the switch management interface may or may not be on VLAN 4095. This can be discovered through the switch VPD data.

# 3 System Functional Blocks

The SSI open blade server specification defines a modular decomposition of a blade server into the following components:

- Chassis
- Power
- Interconnect
- Compute
- Switching
- Management

## 3.1 Power

A chassis provides +12v power to all blade slots. This power is not switched or otherwise controlled at the slot, so the blades themselves are responsible for following the power budgeting protocol and limiting their power consumption. The *SSI Compute Blade Specification* does not make requirements relative to power subsystem decomposition (redundancy, number of modules, etc.), but it does identify manageability requirements.  The Chassis Manager **should** find the total power capacity of the system from the Power Supplies, and factor this into making power on policy decisions.

## 3.2 Cooling

An open blade server provides cooling capabilities that are scaled to the supported blade population. The cooling sub-system is managed by the Chassis Manager, which maps blade power budget and thermal state to cooling sub-system airflow. A chassis may be divided into multiple, separately controllable cooling zones, each with its own associated blades to be cooled.

Acoustics are strongly related to cooling because the noise generated is proportional to cooling airflow. The cooling sub-system reduces noise by producing only enough airflow to successfully cool the chassis. This may change depending on blade population, power consumption, and cooling policies.

The IDROM portion of the SSI Chassis Management Module (CMM) Specification contains the cooling zone map.

## 3.3 Compute Blades

Compute blades are nominally servers in their own right, containing the equivalent functionality of a standalone 1U or 2U server, without the power and cooling hardware. A compute blade can occupy one (single-wide) or two (double-wide) slots and may or may not contain storage devices.

Compute blades are hot-swappable. They are capable of being inserted into and removed from an SSI chassis without removing chassis power, allowing for on-line (from the over-all open blade server perspective) configuration changes and repairs.

The SSI Compute Blade Specification defines the mechanical, electrical, and management requirements for compute blades.

### 3.3.1 Connectivity

An SSI chassis provides the following connectivity for compute blades.

For more details than provided below, see the *SSI Compute Blade Specification*.

**Input Power**

The SSI mid-plane provides a separate power connector for use by the compute blade. Only +12v is provided and a compute blade may draw up to a maximum of 450W if it has been allowed by the Chassis Manager.

**Presence**

The SSI mid-plane implements a per-slot presence signal going to each CMM slot that allows the Chassis Manager to track chassis blade population independently of whether or not the blade management is working correctly.

**Slot Identity**

The SSI compute blade signal connector dedicates six slot identity pins to provide the blade with a chassis unique identity. The blade's management controller uses these ID pins to calculate its I2C address when communicating with the Chassis Manager.

**High-Speed Interconnections**

An SSI compute blade's signal connector defines three sets of high-speed signal interconnections to the mid-plane.

- **Primary Interconnect**
  Connections for two primary Ethernet Links that connect to Primary Switch 1 and Primary Switch 2 exist on the Compute Blade baseboard.
- **Optional Flexi-Interconnect**
  There are four x4 PCIe lanes going from the Blade to the Mezzanine connector. There are two x4 Lanes and two x2 I/O lanes coming out of the Mezzanine Card to the Midplane (and routed to the appropriate switch slots). The Chassis Manager queries the blade management controller to determine compatibility between the Blade's and Switch support for these channels.
- **OEM-defined Interconnect**
  A pair of blade signal connector channels are defined for OEM use. The SSI specification places no technology requirements on them.

**Blade Management Interconnections**

An SSI Compute Blade's signal connector defines a pair of $I^2C$ bus connections, which operate as a redundant management bus called the Base Management Interconnect (BMI). When redundant CMMs are present, one bus is connected to the first CMM and the second bus is connected to the second. This bus is used for communication by the Chassis Manager for blade discovery, configuration, and activation of the blade.

In the SSI Chassis Management Module Specification, the Midplane IDROM contents are specified. As part of that, a full I2C map to each component in the system also exists.

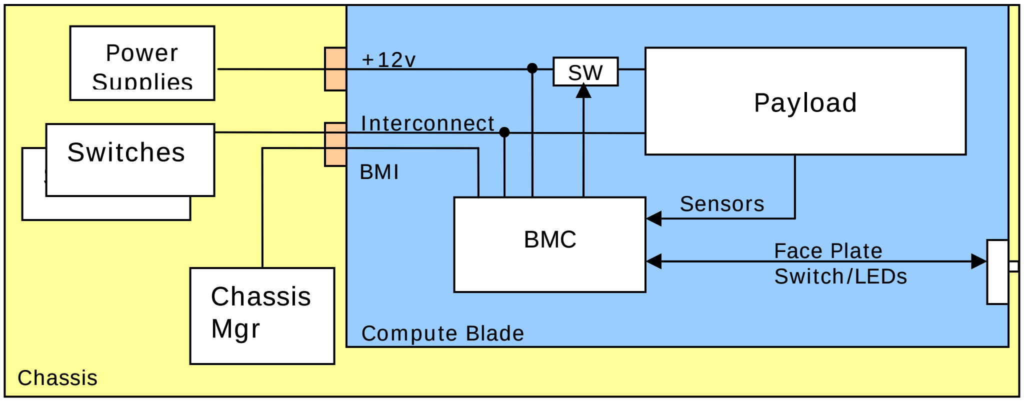
### 3.3.2 Compute Blade Management

An SSI chassis has a Chassis Manager function that communicates with blades, querying for inventory and state information. The Chassis Manager communicates with blades via the Base Management Interface (BMI) and/or the Primary Channel Ethernet link(s). The Chassis Manager uses the BMI for initial blade discovery and configuration. Blade activation and monitoring may continue over the BMI or via Ethernet. Special Ethernet-only services (such as KVM over IP may exist as well on a Blade).

Every SSI compute blade implements an intelligent management controller (BMC). This controller is always powered and active while the blade is in a chassis and the chassis has power, irrespective of the state of the blade's payload. A compute blade **shall** implement an interface to the Base Management Interface as well as management via the Primary Interconnect.

Figure 3-1 illustrates a block diagram of the management elements in a typical compute blade, also showing their relationship to chassis elements.

**Figure 3-1. Typical Compute Blade Management**



The basis for the management protocols and model is the *Intelligent Platform Management Interface* (IPMI) specification (v2.0). This specification qualifies the IPMI specification, making additions and exceptions to its requirements.

SSI compute blade management provides support for the following features:

- **Inventory**
  The Chassis Manager can determine a blade's identity, including manufacturer, product name, and unique identifier (GUID), allowing management software to track chassis-wide inventory. A blade's

attached modules can also be identified for further understanding of the blade's configuration.

- **Compatibility**
  A Compute Blade provides the Chassis Manager an inventory of its supported high-speed interfaces and their technologies. This allows the Chassis Manager to determine whether the blade is appropriate for the slot it's in and to alert the user if it is not.

- **Insertion/Extraction - Hot Swap**
  Compute blades are hot swappable; they can be added to and removed from a chassis without removing chassis power. The blade's BMC implements support for negotiating with the Chassis Manager for the blade's activation after insertion (including payload power-on) and deactivation before extraction (including payload shutdown and power-off). Compute blade management also supports face plate indicators and buttons to provide control and feedback to users.

- **Instrumentation**
  IPMI defines a sensor model for providing instrumentation of a blade's state. An SSI Compute Blade implements, at a minimum, the sensors defined in the Compute Blade specification. When conditions warrant, a blade's sensors generate events that are sent to the Chassis Manager.

- **Payload Control**
  The operating system and software that actually runs on the primary CPU(s) of the blade are called the "payload". The BMC provides out of band mechanisms for controlling power and resetting the blade. This allows for remote recovery of various configuration, application, and operating system failures. In some cases, even out of band remote KVM (keyboard, video, and mouse) and remote media (remotely hosted CD) may be available. The CMM will provide access to these services.

- **Cooling Management**
  An SSI chassis provides cooling services for the blades. For this to be effective, a blade's thermal state needs to be used to inform the CMM to perform the proper cooling sub-system actions (e.g., Increase Fan Speed). The *SSI Compute Blade Specification* defines specific a per-blade IPMI thermal sensor that the Chassis Manager can use for this purpose.

- **Power Management**
  An SSI chassis has limited power resources that may be dependent on its configuration or degraded as a result of failures. The blade activation process includes a negotiation between the blade and the Chassis Manager for amount of power that the blade may use. This prevents the chassis' power sub-system from being over committed, which may not otherwise show up until a confluence of maximum blade power utilizations overload it.

### 3.3.3 Compute Blade BIOS

SSI compute blades implemented with IA32 processors **shall** implement BIOS requirements to support integration with the management infrastructure and to guarantee a uniform environment for operating systems and applications. The *SSI Compute Blade Specification* provides specific requirements in the areas of:

- **BIOS interfaces**
  This includes Legacy BIOS, UEFI, SMBIOS, and ACPI interfaces.
- **Chipset and memory error handling**
  Generally, a platform can detect errors on various buses, such as the CPU front-side bus and I/O buses. In addition, memory device errors can be detected via ECC. Basic support for detecting and logging these must be supported.
- **IPMI management support**
  This includes support for the interface to the management sub-system as well as use of IPMI-defined features such as the IPMI Watchdog to detect and recover from boot failures/hangs.
- **Boot control (disk, configuration, update, PXE, etc.)**
  IPMI supports controlling payload boot type and boot device on a boot-by-boot basis.
- **Ethernet-based management**
  Various management activities (e.g., configuration) are required to be supported over Ethernet.
- **Trusted Platform Module (TPM) support**
  TPMs are used to effect security. They are used by both BIOS and Operating Systems.
- **Microsoft*-specific OS support**
  Microsoft* has requirements in the areas of error handling (WHEA) and OS activation (SLP 1.0 and OEM Activation).

## 3.4 Switch Modules

SSI Has two switch form factors. The 1x and the 4x. The 1x is a smaller form factor and has a single lane going to each blade. The 4x is a larger form factor that has 4 (aggregated) lanes going to each switch slot. The switch modules contain an SEEPROM containing Vital Product (VPD) data and status and control registers for determining the status and sending "commands". Details on Switch features and connectivity are located in the *SSI Switch Base Specification* and *SSI Switch VPD Specification.* Switch management requirements are found in the *SSI Switch Management Specification*.

## 3.5 Chassis Manager

The Chassis Manager has two high-level functions. The first is to monitor and control chassis physical resources, including shared resources like power and cooling, as well as blade modules such as compute blades and switch modules. The Chassis Manager supports a common module hot-swap model, and it implements the mechanisms and policies associated with that model. The interaction of the Chassis Manager with SSI Blades and switches are covered in the SSI standards.

The second high-level function is that of representing chassis management externally. This is commonly performed with a GUI hosted on the Chassis Manager. The Chassis Manager also implements the SSI External Management (Dashboard) interface for basic (programmatic) access to the common features of the system. External management software and administrators can use this interface to track chassis state and set policies and other configuration state. For more detailed and comprehensive management information, the DMTF Modular Computer System profile may be implemented.

An SSI Chassis Manager has basic requirements for implementing both of these functions. Vendors are free to extend them to add value.

The Chassis Manager may support redundancy for enhanced availability. Minimally, a Chassis Manager will have access to the BMI and Ethernet management networks.

## 3.6 Managed Elements

SSI managed modules are entities that have management states associated with them. State may be only inventory related – manufacturer, product name, and the like, or may include more involved state such as thermal, power, and interconnection configurations.

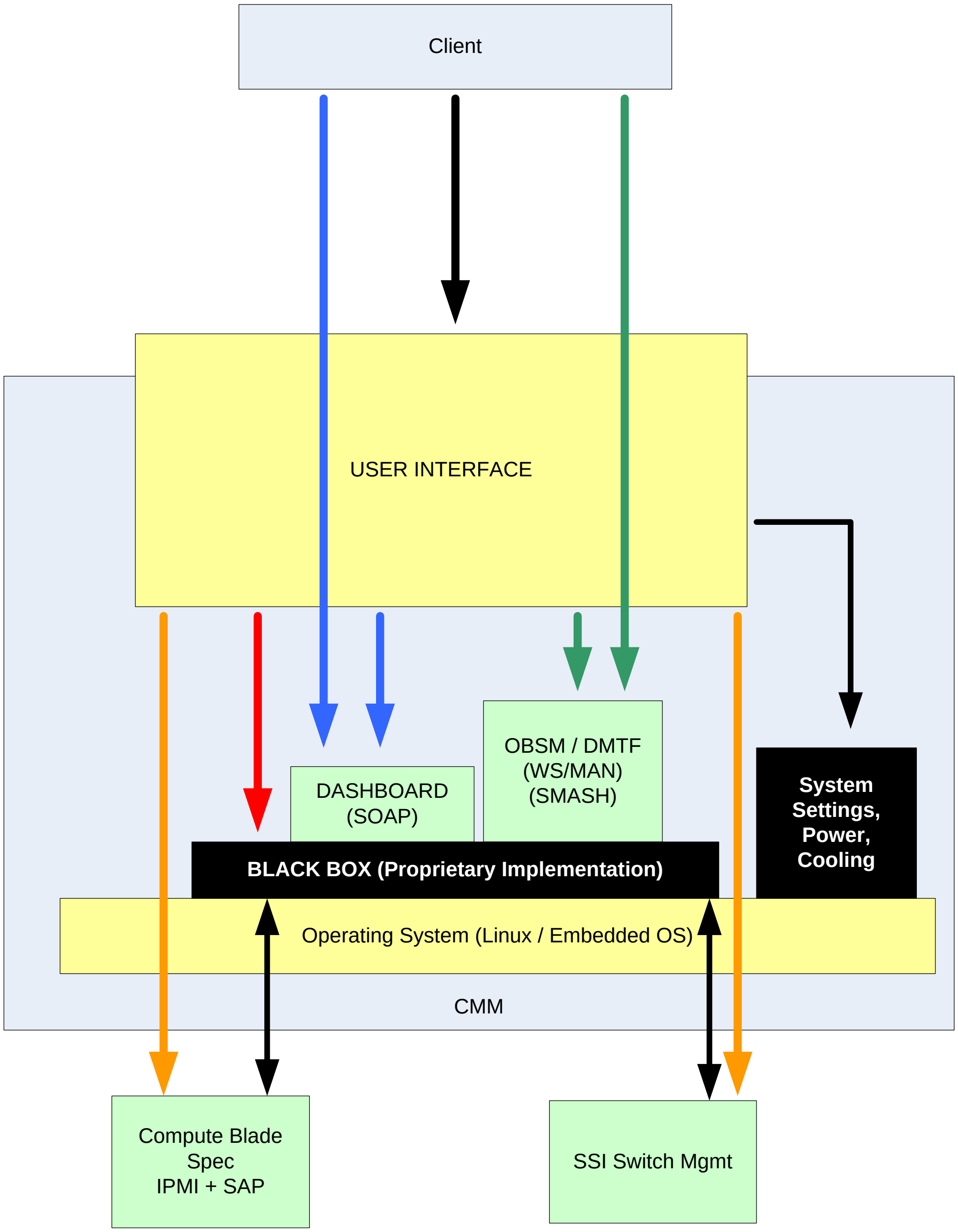
Managed modules, such as compute blades and switch modules, have intelligent management controllers that communicate with the Chassis Manager via the management network(s). They represent the blade's management state to the Chassis Manager. See Section 3.3.2 for an overview of compute blade management.

A module may also be a chassis module that is itself managed by some other managed module. As an example, consider a cooling management module that controls separate fan modules. The cooling management module is directly managed by the Chassis Manager, but the fan modules are managed only indirectly through the cooling management module. Similarly, a compute blade mezzanine board is a module represented by the compute blade.

# 4 Chassis Manager Software Architecture

Figure 4-1 illustrates the SSI software model. The individual components are described in this section.

**Figure 4-1. SSI System Software Model**


Client
USER INTERFACE
DASHBOARD (SOAP)
OBSM / DMTF (WS/MAN) (SMASH)
System Settings, Power, Cooling
BLACK BOX (Proprietary Implementation)
Operating System (Linux / Embedded OS)
CMM
Compute Blade Spec IPMI + SAP
SSI Switch Mgmt

## 4.1 Operating System

Depending on the system and licensing requirements, a variety of operating systems (that support a network stack) may be used.

## 4.2 User Interfaces

User interfaces include the standard interfaces (SNMP, SMASH, WS-MAN, SOAP), as well as a GUI. GUI specifications are outside the scope of this document. However, exposing basic controls such as firmware update, system inventory, system health, and module control **should** be implemented at a minimum.

## 4.3 Storage

The CMM specification provides a mechanism for a single or dual, centralized SD storage device on the mid-plane of a chassis. This can optionally be used to keep the CMM software on the chassis itself. This could be used to store databases only or the entire CMM software stack. When designing a midplane, support **should** always be made for this form of storage by providing an SD mounting device and properly pinning it to the CMM slot.

## 4.4 CMM Redundancy

In a redundant CMM system, only one can be the master CMM at any given time. Both CMMs **shall** be of the same make. Therefore the algorithms developed for redundancy only need to comprehend a software/hardware stack that is the same. The hardware tools available for redundancy are the following:

- CMM to CMM Ethernet connection (for coherent communications).
- CMM to CMM RS-232 serial connection (for heartbeat/health monitoring (and backup to Ethernet)).
- CMM select line (for asserting the master CMM).

**Takeover:** Upon takeover and asserting the CMM select line, only the new master CMM **should** send I2C commands to the managed devices, and the new master **should** take over the internal IP address of the old master as well as the SD card (in the case of a single SD). Each of the managed devices **should** also be pinged or gratuitously ARPed in order to flush the ARP cache in each device. Upon takeover, the switch in the backup CMM **should** be disabled.

Additional information on the CMM can be found in the *SSI Chassis Management Module Specification.*

## 4.5 Management Networks

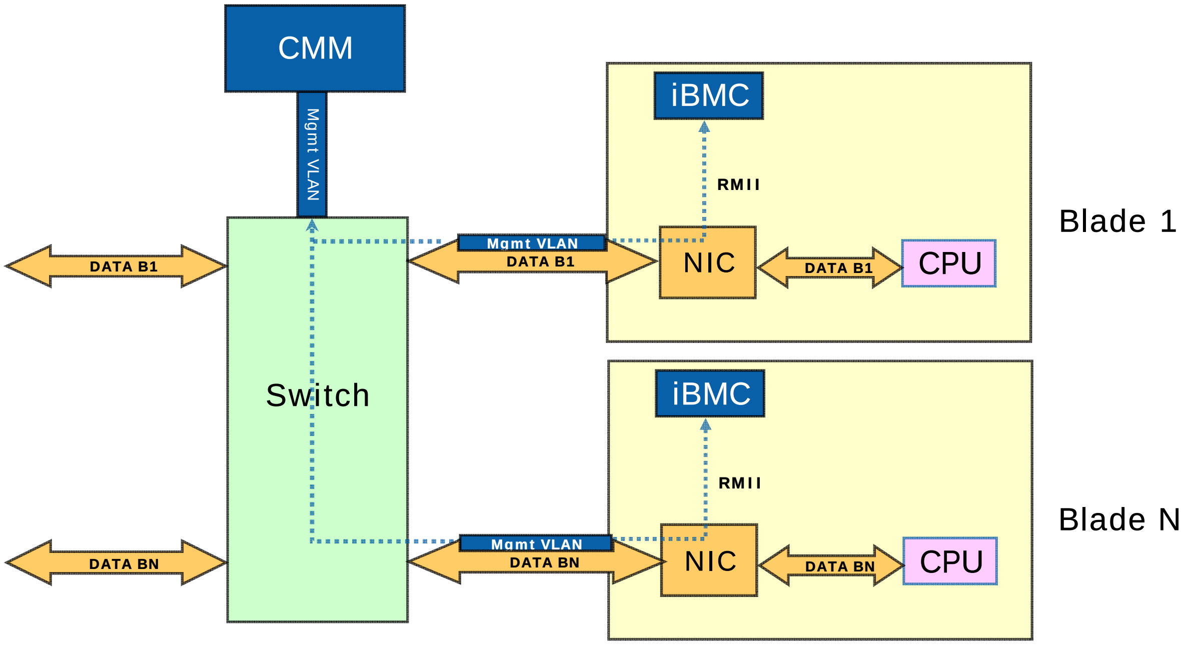
An SSI chassis is required to implement Ethernet management connectivity via the Primary Ethernet fabric.

### 4.5.1 Management VLAN

Ethernet switch modules used as Primary Switches implement a management VLAN ID of 4095 that is used to communicate internally in the chassis between the Chassis Manager and Compute Blade management controllers. Both the Chassis Manager and the Compute Blade BMCs must tag the management traffic with the management VLAN ID. In

Figure 4-2, all traffic represented in blue is 4095 tagged traffic. The Compute Blade NIC must have the proper filters setup to ensure the management traffic gets to the BMC.

**Figure 4-2: Management VLAN Topology**



In Linux, sending out tagged VLAN frames is done simply by adding a virtual (802.1q) interface. This is accomplished with the "vconfig" command. Below is an example:

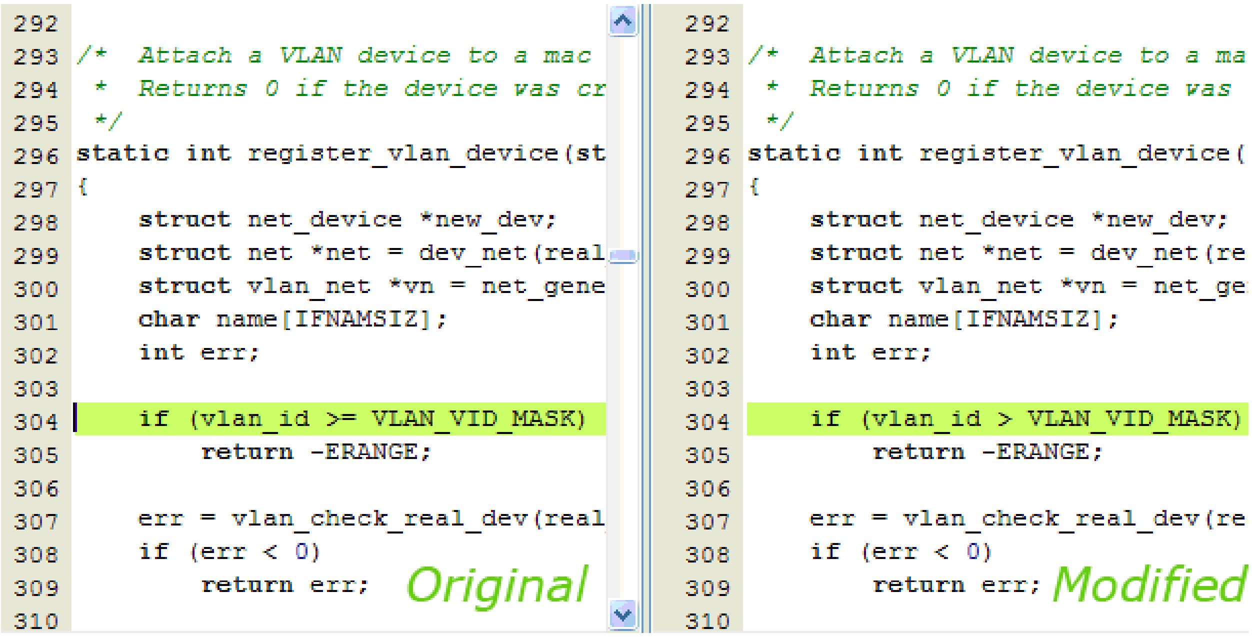
```
# vconfig add eth1 4095
# ifconfig eth1.4095 1.1.1.254 netmask 255.255.255.0 up
```

Some operating systems (including Windows® and Linux) will not be able to perform this out of the box. The reason is that VLAN 4095 is often reserved for special use (typically discarding frames). In fact, as of Linux kernel 2.6.30, the first line above will generate an error on an unmodified kernel.

Therefore, if the BMC or Chassis Manager runs Linux, a kernel modification may be required to configure the tagged VLAN interface. Figure 4-3 shows a diff reference to at least one of the changes that may need to be made to a 2.6.x

Linux kernel to support VLAN 4095 (/usr/src/linux//linux/net/8021q/vlan.c). The original source is on the left, and the modification is on the right. As you can see, this change effectively increments the supported VLAN ID by 1. his "hint" is only intended to be an example; it is not guaranteed to be free of side effects. Also, keep in mind of any obligations that GPL may require of such modifications:

**Figure 4-3: Linux VLAN Limit Workaround**



```

/*  Attach a VLAN device to a mac
 *  Returns 0 if the device was cr
 */
static int register_vlan_device(st
{
    struct net_device *new_dev;
    struct net *net = dev_net(real
    struct vlan_net *vn = net_gene
    char name[IFNAMSIZ];
    int err;

    if (vlan_id >= VLAN_VID_MASK)
        return -ERANGE;

    err = vlan_check_real_dev(real
    if (err < 0)
        return err;

```

```

/*  Attach a VLAN device to a ma
 *  Returns 0 if the device was
 */
static int register_vlan_device(
{
    struct net_device *new_dev;
    struct net *net = dev_net(re
    struct vlan_net *vn = net_ge
    char name[IFNAMSIZ];
    int err;

    if (vlan_id > VLAN_VID_MASK)
        return -ERANGE;

    err = vlan_check_real_dev(re
    if (err < 0)
        return err;

```

The Chassis Manager **shall** also support (non-primary) switches that do not use VLAN 4095 to communicate with the switch.

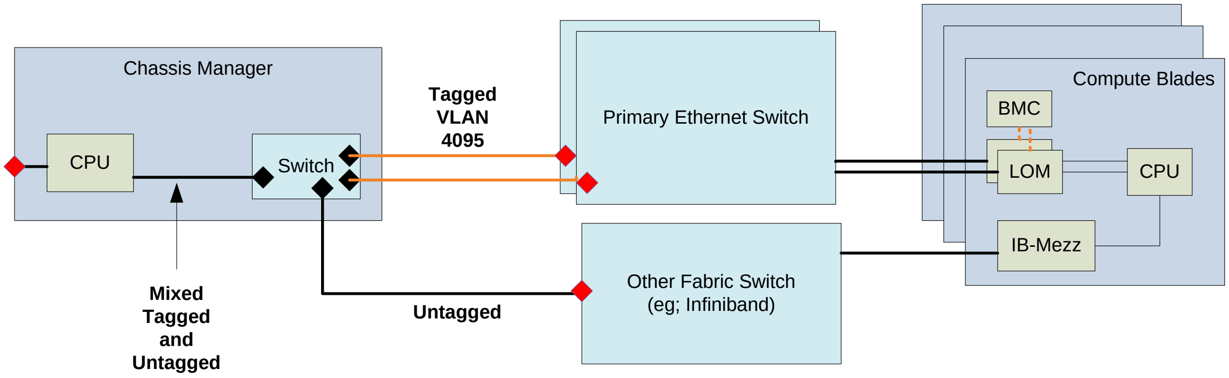
## 4.5.2 Switch Management Network Requirements

Web, SNMP, and SSH (CLI) interfaces are required in the SSI Switch *Base Specification*. These interfaces are intended to be directly exposed to the end user.

The management architecture is designed as such that advanced and detailed management tasks that are specific to the individual switch module **should** be provided by the interfaces integrated into the switch module. For instance, the SSI Switch Base Specification requires that the Web Interface includes features such as Firmware Update, Port Status, Network Configuration, SNMP Configuration, etc.

To support these features being available to the end user, the Chassis Manager must have a network configured in such a way that the internal Ethernet interface that is on the switch must be accessible directly by the external interface of the Chassis Manager. Therefore, each interface represented by a red diamond (in Figure 4-4) would actually be in the same (external) subnet.

**Figure 4-4: Switch Management Ethernet Topology**



This access must be so transparent that even the IP range that is used on the switch must be the same as the IP range on the Chassis Manager external uplink. Essentially the switch needs to appear as if it is in the same network as the external port of the Chassis Manager.

The Chassis Manager must also support connecting to switch management interfaces that **may** or **may not** be a 4095 VLAN tagged interface. Although Primary switches are required to use 4095 tagged VLANs, other switches may not use tagged interfaces. Since the CMM may only have a single connection to its integrated switch, this implies that the single interface on the CMM will need to support both 4095 tagged and untagged traffic.

In Linux, potential solutions that have been explored are iptables port forwarding, iptables IP forwarding of virtual interfaces, proxy-arp, and bridging. In the following examples, this document will explore the merits of each (with http as an example) and provide a conclusion. Each of these options were explored in a "Proof of Concept" (POC) effort to produce this document.

### 4.5.3 Bridging

Bridging is a network implementation that assembles the internal interfaces into a bridge that has its own IP address. In this configuration, IP addresses are not assigned to individual ports on the CMM (such as eth0 and eth1), but rather to the bridge (br0) that bridges the traffic between all interfaces (just like a general Ethernet switch would). All of the problems below in the other implementation ideas are solved in a bridged environment. However, a new problem is introduced. Since broadcast traffic from the outside is bridged inside, more "noise" than is desirable may appear in the internal interfaces. Additional security issues could be introduced here as well if the internal network is not protected. So, a tool like ebtables will have to be used to limit the traffic going across the bridge. Additionally, if two Chassis are plugged into the same external network, IP conflicts between the internal IP addresses can arise if the proper filters are not put into place. Bridging is a recommended implementation. Below is a Linux configuration example with a system that has bridge-utils installed:

```
# create bridge
brctl addbr br0

# add interface to bridge
brctl addif br0 eth0
brctl addif br0 eth1
brctl addif br0 eth1.4095

# clear IP configurations from individual interfaces.
ifconfig eth0 0.0.0.0
ifconfig eth1 0.0.0.0
ifconfig eth1.4095 0.0.0.0

# set external ip address of bridge
ifconfig br0 192.168.0.1 netmask 255.255.255.0

# set internal ip address for communications to blades
ifconfig br0:1 1.1.1.254 netmask 255.255.255.0

# add ebtables rules to isolate internal (private) network

ebtables -A INPUT -p ARP -i eth0 --arp-ip-dst 1.1.1.0/24 -j DROP
ebtables -A INPUT -p ARP -i eth0 --arp-ip-src 1.1.1.0/24 -j DROP

ebtables -A FORWARD -p ARP -o eth0 --arp-ip-dst 1.1.1.0/24 -j DROP
ebtables -A FORWARD -p ARP -o eth0 --arp-ip-src 1.1.1.0/24 -j DROP
ebtables -A FORWARD -p ARP -i eth0 --arp-ip-dst 1.1.1.0/24 -j DROP
ebtables -A FORWARD -p ARP -i eth0 --arp-ip-src 1.1.1.0/24 -j DROP

ebtables -A OUTPUT -p ARP -o eth0 --arp-ip-dst 1.1.1.0/24 -j DROP
ebtables -A OUTPUT -p ARP -o eth0 --arp-ip-src 1.1.1.0/24 -j DROP
```

### 4.5.4 Iptables port forwarding

Port forwarding is a method where the IP address of the external Ethernet port of the Chassis Manager is used to access the switch. This is done by taking an unused TCP port number and redirecting it to an internal IP address. For example, external IP:port 192.168.0.1:81 is redirected to 1.1.1.101:80 (internal IP of switch). The good thing about this approach is that a single IP address is used for all management of the system.

Unfortunately, this method will begin to fail if the web server on the switch ever does a "redirect" to a page that calls out :80 in the URL. Since many switches of this standard were designed around the IBM® Bladecenter® behavior (which uses external IP addressing for switches), this approach could be very risky. This is not a recommended approach.

### 4.5.5 Iptables IP forwarding

IP forwarding is similar to port forwarding except that the Chassis Manager sets up a virtual external interface that is a separate IP from the Management IP of the Chassis Manager. Traffic to that IP is redirected to an internal IP. One nice aspect of this approach is that it is immune to the port number redirect as described above, however it is not immune to a redirect to the IP that the switch thinks is the management IP. For example, if the browser is going to

192.168.0.101 and that IP is redirected to an internal switch IP of 1.1.1.101, then, if the web server on the switch ever does a redirect to http://1.1.1.101/test.html, the browser would not be able to reach that address. This too is not a recommended approach.

### 4.5.6 Proxy Arp

Proxy Arp is a means by which the external Ethernet interface will respond to ARP requests to the IP of the internal interface. In this scenario, the internal IP address of the switch is within the same range as the external interface of the Chassis Manager. In the POC experiments, this approach seemed to remedy both of the issues that were raised above. The only use case it did not handle is getting an IP address via DHCP. Since DHCP is a supported use case for SSI switches, this approach is not recommended.

## 4.6 Unmanaged Chassis (absent/faulty CMM)

Each module **should** employ a mechanism to eventually come up in the absence of a CMM. For instance, if a system has a single CMM, and it goes into a fault state, it may not be able to communicate with a blade to give a synchronized power up. In this case, the system designer **should** consider which services are “mission critical”.  In most cases, the “mission critical” component is the performance of the blade, switch, and I/0 payloads. Having a long timeout to wait for CMM communications is the recommended way to handle an absent or faulty CMM.

## 4.7 Security

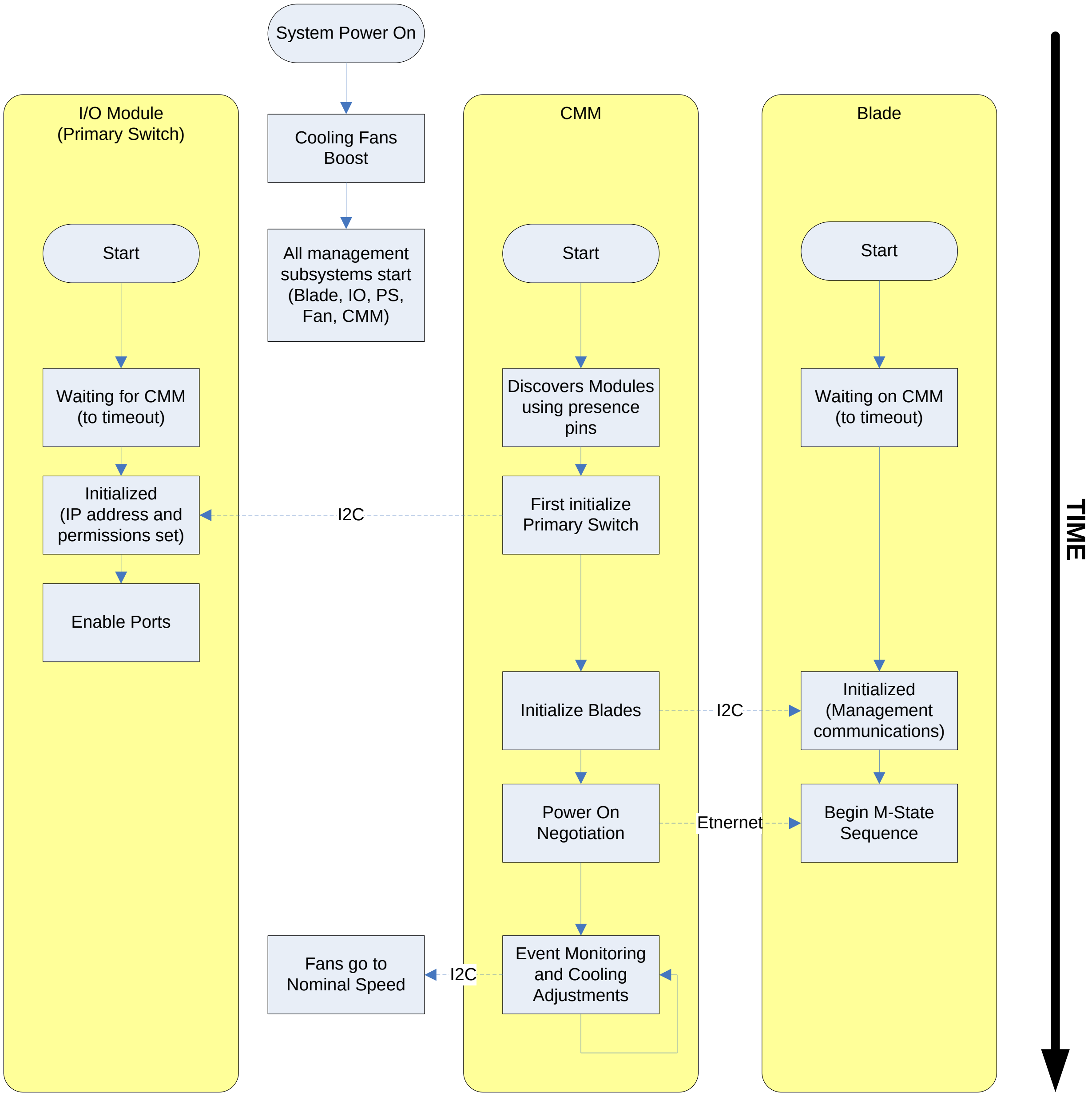
In the SSI management architecture, the blade management controllers are trusted entities, but blade payload software isn’t. Due to this, the blade management controllers are required to limit the ability of payload software to affect the management state of other blades or chassis resources. See the *SSI Compute Blade Specification* for details.

Similarly, management entities such as the Chassis Manager (and in some deployment models – blades), that communicate with software external to the chassis are required to implement authentication and encryption over those interfaces, e.g., Ethernet.

# 5 System Startup

The SSI blade system startup is coordinated by the Chassis Manager as represented in the example in Figure 5-1.

**Figure 5-1 System Startup Sequence Example**


System Power On
I/O Module
(Primary Switch)
Cooling Fans Boost
CMM
Blade
Start
All management subsystems start (Blade, IO, PS, Fan, CMM)
Start
Start
Waiting for CMM (to timeout)
Discovers Modules using presence pins
Waiting on CMM (to timeout)
Initialized (IP address and permissions set)
I2C
First initialize Primary Switch
TIME
Enable Ports
Initialize Blades
I2C
Initialized (Management communications)
Power On Negotiation
Etnernet
Begin M-State Sequence
Fans go to Nominal Speed
I2C
Event Monitoring and Cooling Adjustments

## 5.1 I/O Modules and Switches

In most cases, I/O Modules **should** be initialized first, since the blades may very well depend on them for their startup or management. In particular, the Primary switch **shall** be initialized first in order for the Ethernet-based BMC management to the blades to function. Initialization of the I/O Modules involves setting VPD data, and it is detailed in the *SSI Switch Base Specification* and *SSI Switch VPD Specification*.

## 5.2 Blades

Upon initial system power-on, the blades will power up their BMCs, which will control the power to the rest of the platform. The Chassis Manager, after first initializing the primary switch(es), will initialize the BMC (Ethernet) management through the BMI interface. Then, the M-State power-on sequence can begin as outlined in the *SSI Compute Blade Specification*.

## 5.3 Power Supplies

Power Supplies act in the very first part of system power on, but need to do little else than provide the prescribed power. The power supply management subsystem **should** be available early in the process to allow the Chassis Manager to properly gauge the available power available to negotiate with the other modules requesting power on.

## 5.4 Cooling Modules

On initial power on, the cooling modules **should** boost to full power (since at this point, the temperature and power consumption data is not yet available to the Chassis Manager. Once the Chassis Manager is fully aware of the system temperatures and cooling requirements, it can then instruct the cooling modules to reduce their speed and provide a quieter system.

# 6 *Event Handling*

## 6.1 Instrumentation

### 6.1.1 Presence Detection

A single presence pin must connect the CMM to each removable module in the system (including blades, power supplies, fans, and I/O modules). The first line of inventory presence detection is this pin. Additionally the management interface for each given module **should** offer additional information. Upon insertion of any module, an insertion handler on the CMM **should** handle the event. In some cases, the CMM will need to go into a state machine to first attempt to communicate with the module, and second to initialize the module.

### 6.1.2 Thermal

The SSI Specifications supports two cooling optimizations for a blade system. In environments that do not have acoustical requirements, the systems may be run cooler by running the fans a little faster for the purpose of maintaining a very conservative temperature level for maximum stability and longevity. Alternatively, many environments require acoustical considerations for the comfort of the operators in the facility. For such systems, running the systems with slower fan speeds, and therefore a little hotter, may be necessary.

Each module **should** have a means to determine its thermal condition at the most thermally critical location. In addition, the management subsystem on each module **should** communicate this condition to the Chassis Manager. On blades, this would typically be CPUs, chipsets, memory, and hard disk drives (HDDs). A means to measure the ambient temperature (near an airflow inlet) **should** also be employed. The *SSI Compute Blade Specification* defines an Aggregate Thermal Magnitude Sensor for the purpose of hinting the thermal condition of the blade. This sensor reports the condition according to the data in Table 1-1.

### Table 6-1. Thermal State Conditions and Values

| Value | State Description |
|---|---|
| 0 | Off<br>The blade will set this value when the payload power is off. |
| 1 | Cold<br>The blade is well below normal operating temperature. |
| 2 | Cool<br>The blade is operating at a temperature that is most optimal for hardware longevity and stability.<br>The CMM **should** attempt to hold the blade at this level if acoustics are not a factor and hardware stability and longevity is the only concern. |
| 3 | Warm<br>The blade is operating at a temperature that is within an acceptable envelope at the high end and may sustain operations at this temperature.<br>The CMM **should** attempt to hold the blade at this level for systems attempting to be acoustically optimized. This level is the optimal balance between acoustics and stability. |
| 4 | Hot<br>The blade is operating at a temperature slightly above Warm and outside of the required range of sustained reliability.<br>The CMM **should** increase the cooling system (fans) to reduce the temperature to 2 or 3. |
| 5 | Hotter<br>The blade is operating at the high edge of the threshold yet still in a non-critical state. Any increase in temperature will cause a degraded state.<br>The CMM **should** increase the cooling system (even more so than level 4) to reduce the temperature to 2 or 3. |
| 6 | Warning<br>The blade is operating outside of the proper operating range and may already be experiencing a degraded performance. If available, the blade **should** “throttle down” to reduce temperature.<br>The CMM **should** increase cooling to 100%. |
| 7 | Critical State -- action required<br>The blade is operating at a temperature that hardware damage may be imminent. The blade **should** attempt to shut down.<br>The CMM **should** attempt to shut down the blade in addition to increase cooling to 100%. |

The system designer **should** design the chassis cooling algorithm in a way that tries to keep the blade’s Aggregate Thermal Magnitude on a value of either 2 or 3. The blade must gather all required intelligence within the blade platform to determine the current (aggregate) thermal state (as indicated above). If the CMM is set to attempt to maintain an acoustically optimized system, then the CMM will try to keep the (hottest) blade at level 3. This **should** be done by continuously adjusting the fan speeds in the appropriate cooling zone as the sensor goes to 2 (slow fans down) or 4 or 5 (speed fans up). If the thermal state reaches level 6, the cooling system **should** go to maximum cooling power. At thermal state 7, the CMM **should** proceed to shut down the blade.

### 6.1.3 Power

The power supplies **shall** have a means to measure the total output and capacity, and this information **should** be available to the Chassis Manager for power budgeting as required by the *SSI Compute Blade Specification*. For power redundancy, an N+N (mirrored power), or N+1 system, may be implemented. The designer **should** keep in mind that if the target product is for a redundant electrical feed in a datacenter, a mirrored solution may be desired (to handle failures of an electrical feed). For a cost-optimized system, an N+1 configuration may be more desirable, since it requires fewer power supplies. Management of the power supplies is currently out of scope of the SSI specifications.

In some designs the fans on the power supplies may be used for cooling other components. Special care **should** be taken to understand the ramifications of a power supply taking in warm air, or using the warm exhaust air of a power supply for cooling.

## 6.2 Hot Swap

The presence pins **should** be used for hot-swap detection, and the CMM **should** run a hot-swap state machine for each type of device. The state machine for each type of device **should** contain all of the appropriate policies for removal or insertion. For example, if a power supply is removed, the proper amount of power capacity **should** be deducted, and the actions necessary to keep the system from going “over budget” **should** be taken. Likewise, on an insertion event, like a switch, the proper initialization routine **should** be started to ensure proper power-on.

## 6.3 Power On/Off

The blades, in particular, have a power on/off M-State machine for ensuring the proper power budgeting is handled. The Chassis Manager will have to respond to every power-on and -off event of the blade to maintain a proper M-State machine. Special considerations may be needed to handle an absent or unavailable CMM.

## 6.4 Critical Events

All potentially critical conditions and a policy for handling each condition **should** be maintained by the Chassis Manager. Some critical conditions will require drastic measures, like shutting down blades to preserve the rest of the system. The chassis manager **should** contain a policy system for handling every type of critical event.

## 6.5 Non-critical Events

At a minimum, non-critical events **should** be logged by the Chassis Manager. Some will require a policy (such as hot swap and power on/off events) response, while others will simply need to be logged.

# 7 *Systems Management*

## 7.1 User Interfaces

Although user interfaces are not specified in the specifications, it is expected that Chassis Manager designers would implement at least a web graphical user interface for an SSI System. For features like remote KVM/media, using a web interface would allow for launching the client-side applets required. In the *SSI Chassis Management Module Specification*, the IDROM portion of the spec identifies the count, location, and dimensions of the chassis and each of its modules. This data could be used for constructing a representational image of the chassis.

## 7.2 Launching Applets

The *SSI Compute Blade Specification* describes a way to launch rich applets, such as KVM over IP or remote media (remote CD/DVD). In general, the specification supports a single sign on approach where users who are logged into the CMM could launch the applets stored on the blade BMCs without having to login again. The design is to use a ticket (cookie-like) system, which allows the CMM to privately authenticate a session with the blade BMC. The BMC will present the CMM with a “ticket,” which can be presented to a client browser. The browser can use this ticket as a pre-authenticated pass string.

## 7.3 Programmatic Interfaces

At a minimum, a Chassis Manager **shall** implement the *SSI External Management Interface* (Dashboard). This SOAP interface provides for a general inventory of chassis componentry, as well as basic remote control functionality. It is intended to be relatively easy to implement and easy to write a client to interface with it. The general use case of a "Dashboard" was the driving logic behind the specification.

If a more comprehensive interface is desired, the *DMTF Modular System* profile may be implemented along with a SMASH/CLP/WS-MAN interface for a more comprehensive instrumentation and functionality of the system.

## 7.4 Information and Control

Through the various management interfaces, an SSI chassis **shall** (at a minimum) provide the following information for each module:

- Inventory information
- Logs
- Operational state
- Health state
- Sensor data

Additionally, management control (such as power control) **should** be exposed through these interfaces.

## 7.5 System Maintenance

Occasionally, the system may require hardware or software maintenance. Such duties include clearing and/or acknowledging logs, updating the configuration, replacing modules, and optimizing the system for power, cooling, and performance. At least the web user interface **should** provide a mechanism for these tasks, and some of these tasks may need to be approachable through the management interface as well.

Final Page