OPEN

Compute Summit

January 28–29, 2014 San Jose

Engineering Workshop

OPEN Compute Summit

Microsoft’s cloud server specification

SW Management Overview

Badriddine Khessib,
Director
Microsoft cloud server Firmware Development

Microsoft

Microsoft cloud server spec features

EIA 19” Standard Rack Compatibility

Chassis 12U

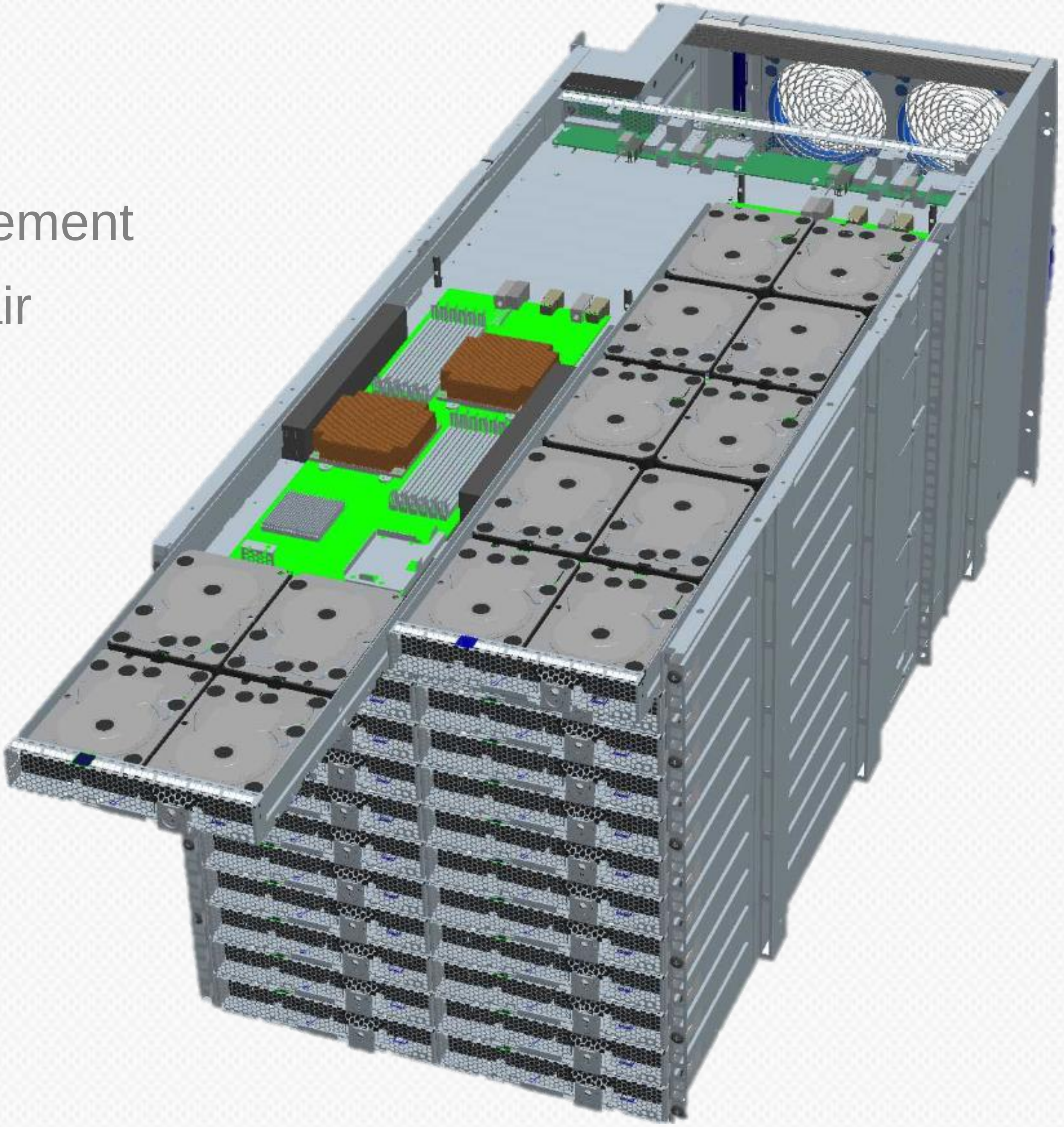
- Highly efficient design with shared power, cooling, and management
- Cable-free architecture enables simplified installation and repair
- High density: 24 blades / chassis, 96 blades / rack

Flexible Blade Support

- Compute blades – Dual socket, 4 HDD, 2 SSD
- JBOD Blade – scales from 10 to 80 HDDs

Scale-Optimized Chassis Management

- Secure REST API for out-of-band controls
- Hard-wired interfaces to OOB blade management

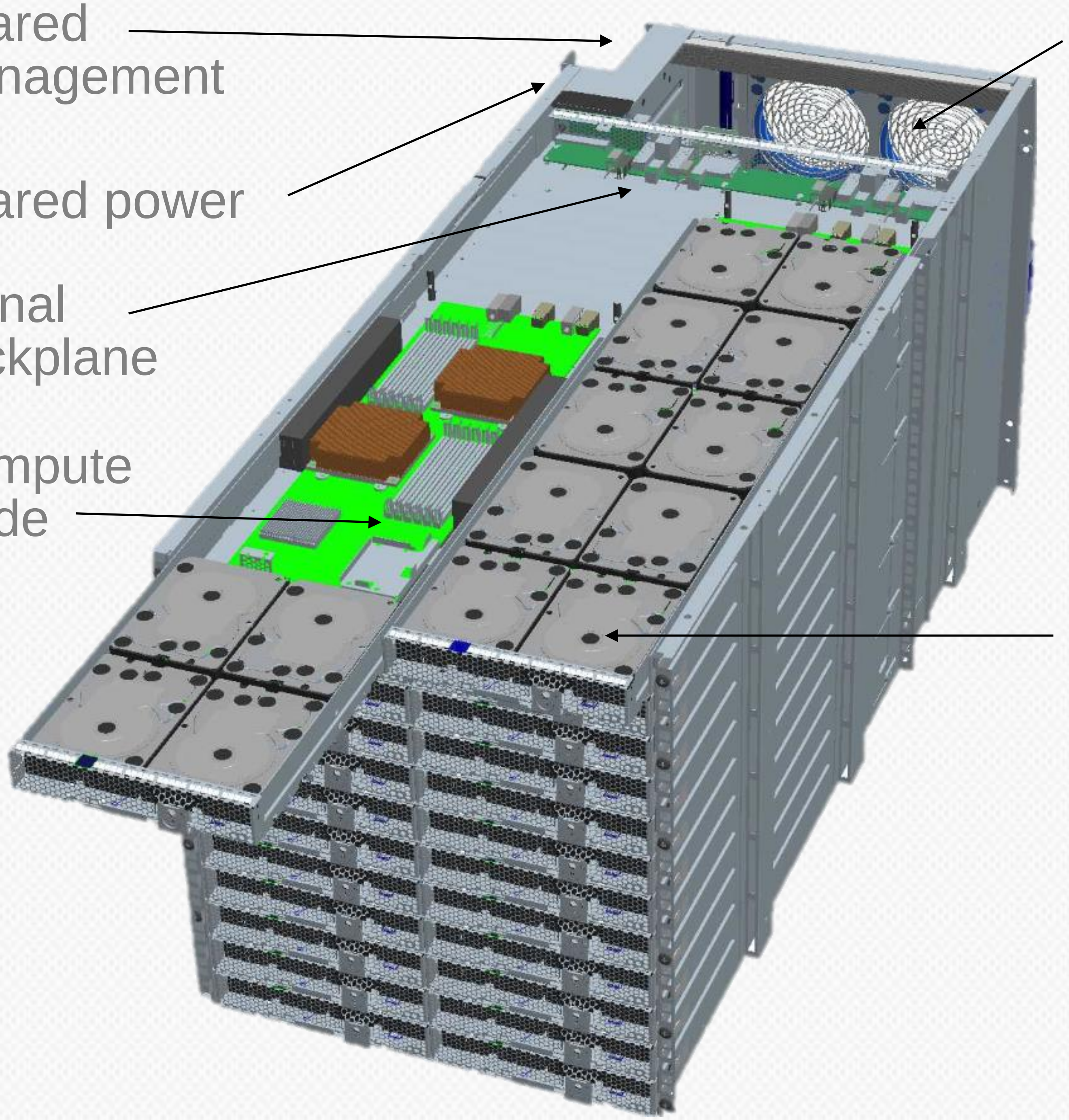


Key features

Shared infrastructure for efficiency and TCO optimization

- Power delivery, mechanicals, thermals/cooling, management
- Optimized for mass contract manufacturing and assembly
- Up to **40% cost savings** and **15% power efficiency** benefits
- **Saves 10,000 tons of metal** per one million installed servers

12U Shared Chassis
EIA Rack Mountable



Shared management

Shared power

Signal backplane

Compute blade

Shared fans

JBOD expansion

Shared management optimized for scale

A requirement of shared infrastructure (fans, PSUs)

Reduction of overall manageability solution cost (ratio is 24 to 1)

Improvement of the scalability of the manageability solution

Enabling scenarios other than server management:

- Assembly and deployment: diagnostics/servicing, cable check
- Asset management
- Power capping at Chassis level

Guiding principles and implications: 1

Simplicity:

- Most management operations should be in-band whenever possible (barebones OOB support)
- Reuse existing solutions and technologies whenever possible
- Abstract infrastructure components for supplier flexibility

Implications

- Minimal set of OOB functionality: Power, Cooling, Blade/Chassis FRU/Logs, Serial Console
- Using industry standard solutions: X86 SOCs, Windows OS, UART communication, IPMI subset
- CM is abstracting Chassis, Blades (compute, JBOD), PDU, TOR

Guiding principles and implications: 2

Scalability

- Optimized for automated lights-out management at scale (> 1M servers)
- Flexible API to allow for easy integration with existing manageability tools

Implications

- Scalable solution: 24 to 1 ratio of servers to CMs (> 1M servers)
- REST API and CLI to allow maximum flexibility

Guiding principles and implications: 3

Security:

- Data center infrastructure and user data should be secure at all times
- Manageability solution should be secure from internal and external threats

Implications

- Security is built-in top to bottom: hardware, OS, application

Guiding principles and implications: 4

Cost Efficiency

- Manageability solution should be extremely low cost
- Reduce overall TCO by lowering deployment and operational costs

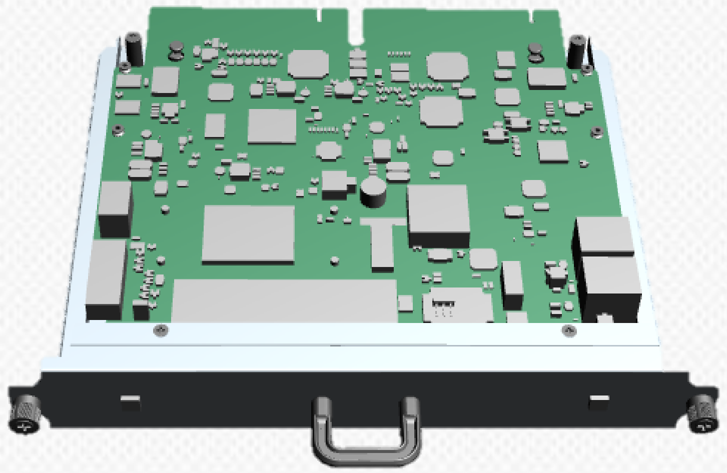
Implications

- Commodity hardware, OS, low speed UART
- CM features targets deployment and operational agility

Chassis manager

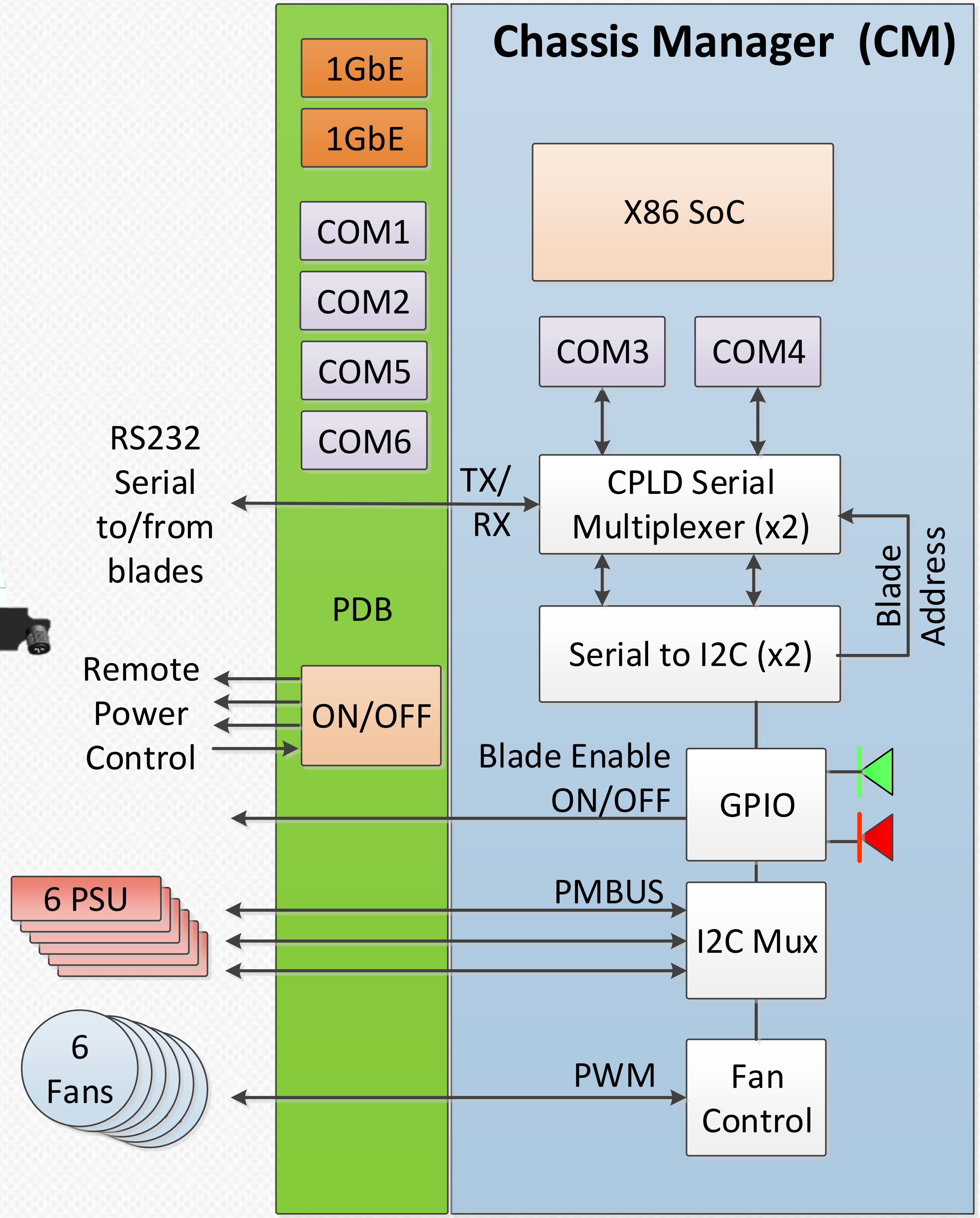
X86 SOC based board that sits in zero U space

- CPU: dual core X86
- Memory: 4GB
- HDD: 64GB MLC SATADOM
- I/O: 6 Serial ports
- Security: TPM1.2
- Dual 1Gb NICs
- 4 AC switches: 1 input, 3 output



Runs Windows Embedded or Windows Server

(minimal configuration)



Chassis Manager (CM)

1GbE

1GbE

COM1

COM2

COM5

COM6

X86 SoC

COM3

COM4

RS232 Serial to/from blades

TX/ RX

CPLD Serial Multiplexer (x2)

Serial to I2C (x2)

Blade Address

PDB

Remote Power Control

ON/OFF

Blade Enable ON/OFF

GPIO

6 PSU

PMBUS

I2C Mux

6 Fans

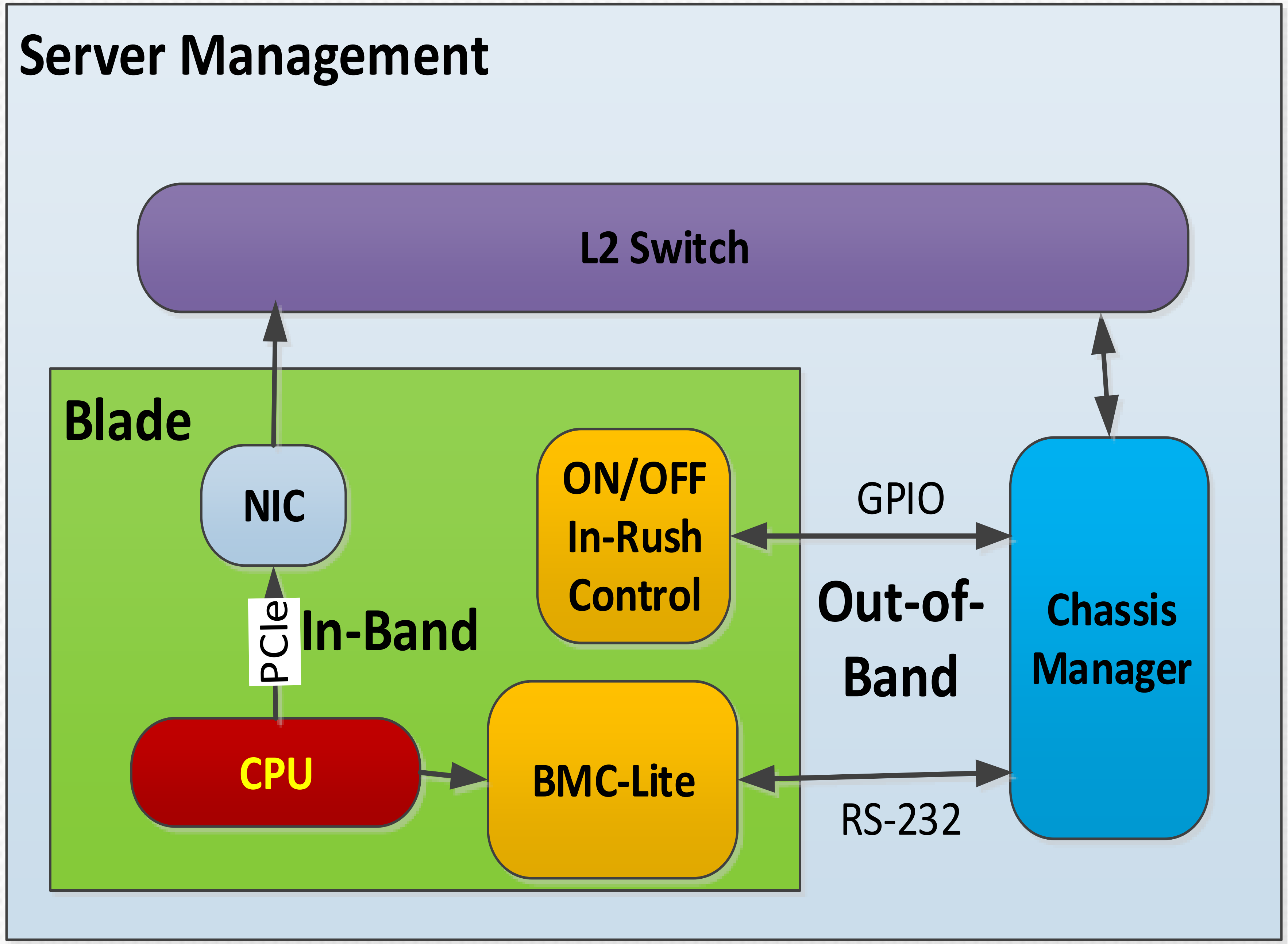
PWM

Fan Control

Manageability for Compute blade

Each Blade has a BMC-Lite:

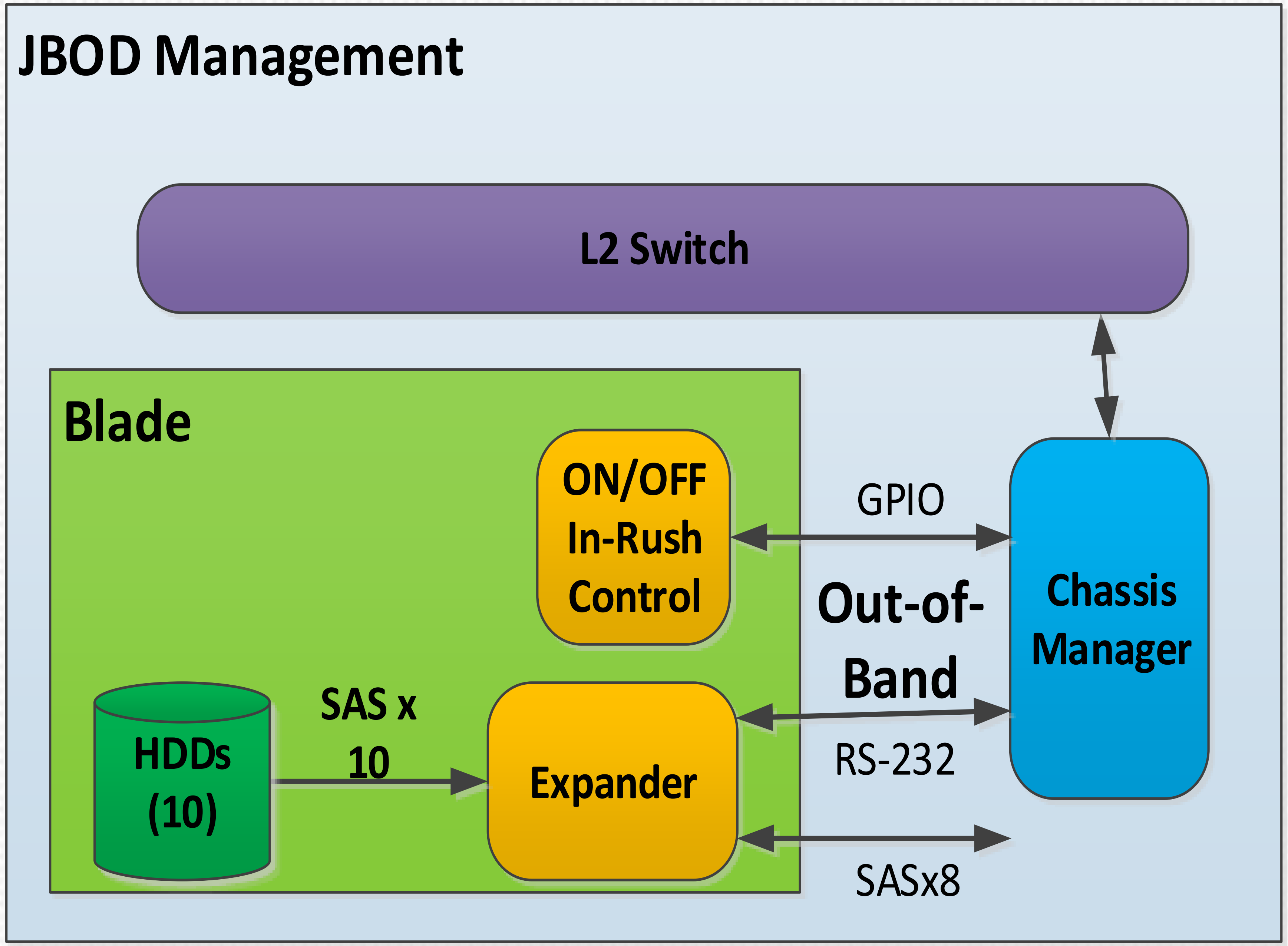
- Chassis manager communicates with each BMC-Lite Via Serial
- Chassis manager controls hard-power to blade through in-rush controller
- Each BMC-Lite implements a small subset of IPMI commands



Manageability for JBOD blade

Chassis Manager communicates with each Expander Via Serial

Each Expander implements a small subset of IPMI commands



Compute blade BMC-Lite

Connected to PCH

- Monitor CPU & DIMMS

Temperature sensors

- Inlet, Exhaust

Voltage and Power Sensors

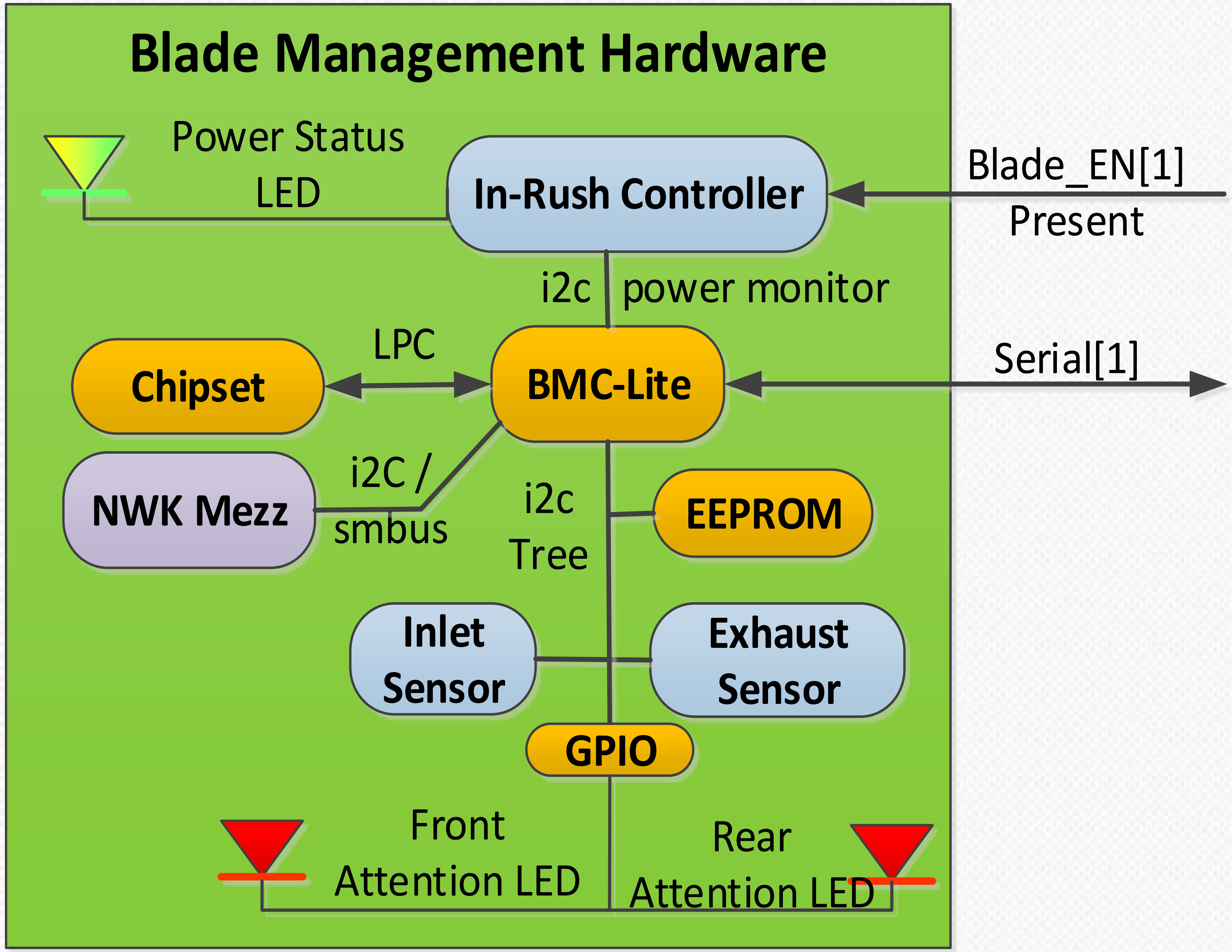
- Voltage regulator

EEPROM to store FRU data

- Serial numbers, model numbers, etc.

Serial out to communicate to chassis manager

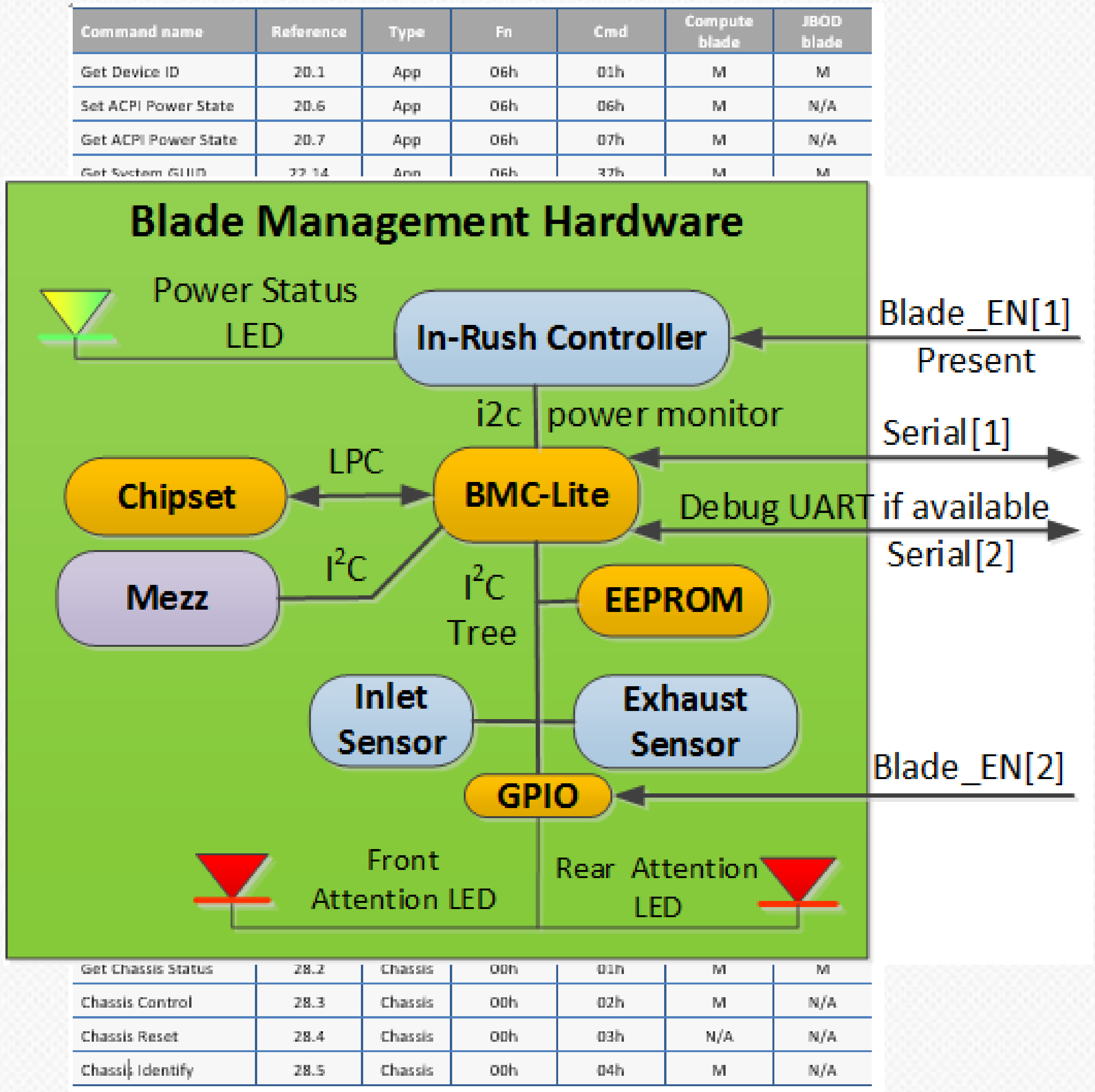
No NIC side-band communication



Compute blade BMC-Lite (cont.)

BMC-Lite

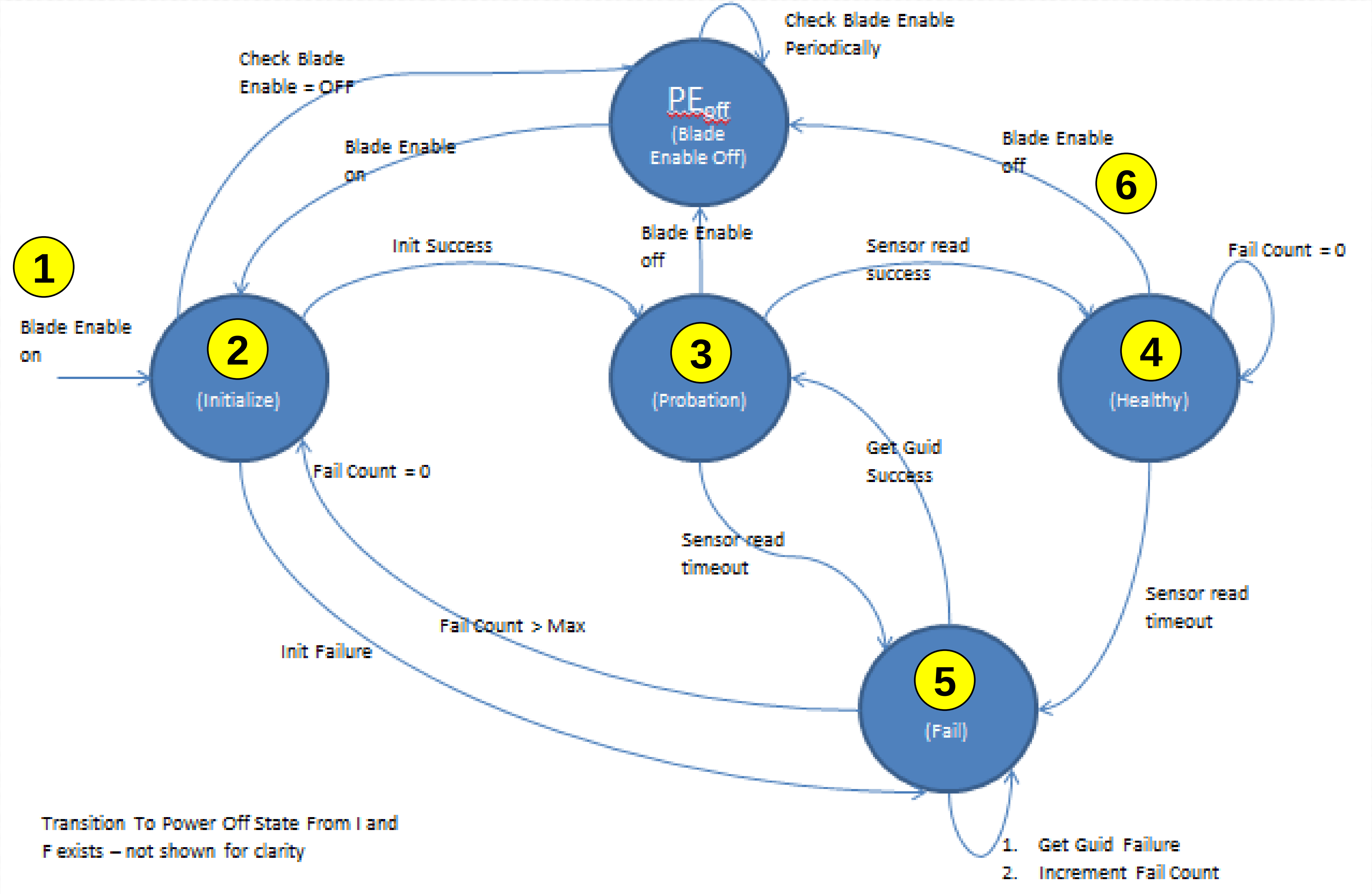
- ✓ IPMI basic mode over Serial
- ✓ I²C Master (SDR)
- ✓ UART I/O
- ✓ System Event Log
- ✓ Power Control
- ✗ ~~KVM, Video drivers~~
- ✗ ~~Ethernet, Network Stack or SOL~~
- ✗ ~~USB~~
- ✗ ~~Full IPMI Command Set~~



Command name	Reference	Type	Fn	Cmd	Compute blade	JBOD blade
Get Device ID	20.1	App	06h	01h	M	M
Set ACPI Power State	20.6	App	06h	06h	M	N/A
Get ACPI Power State	20.7	App	06h	07h	M	N/A
Get System GUID	22.14	App	06h	37h	M	M
Get Chassis Status	28.2	Chassis	00h	01h	M	M
Chassis Control	28.3	Chassis	00h	02h	M	N/A
Chassis Reset	28.4	Chassis	00h	03h	N/A	N/A
Chassis Identify	28.5	Chassis	00h	04h	M	N/A

Chassis manager - blade state management

1. Power On
2. Init()
3. Probation
4. Healthy
5. Fail
6. Hard Power Off



Transition To Power Off State From I and F exists – not shown for clarity

Security: defense in depth

All CMs come with a TPM (1.2)

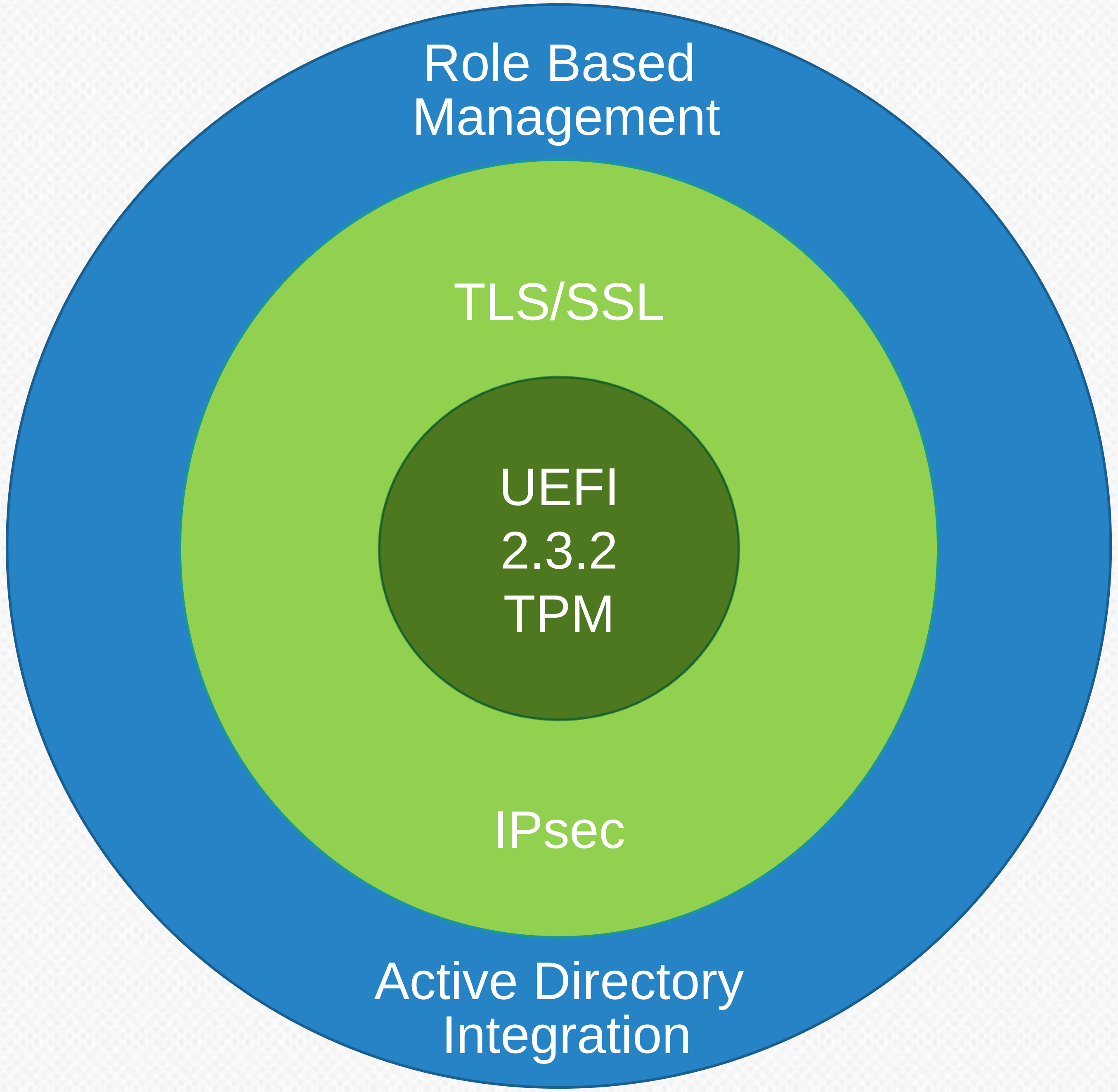
All CMs implement UEFI 2.3.2

- Secure BIOS
- Secure Boot

All CMs support TLS (SSL) and IPsec for communication encryption

Support for local or security-domain based groups/users

- Active Directory integration
- Three user groups: Admins, Operators, Users



Chassis manager software architecture

Provides all OOB Management functionality

Chassis Management (CM) Service

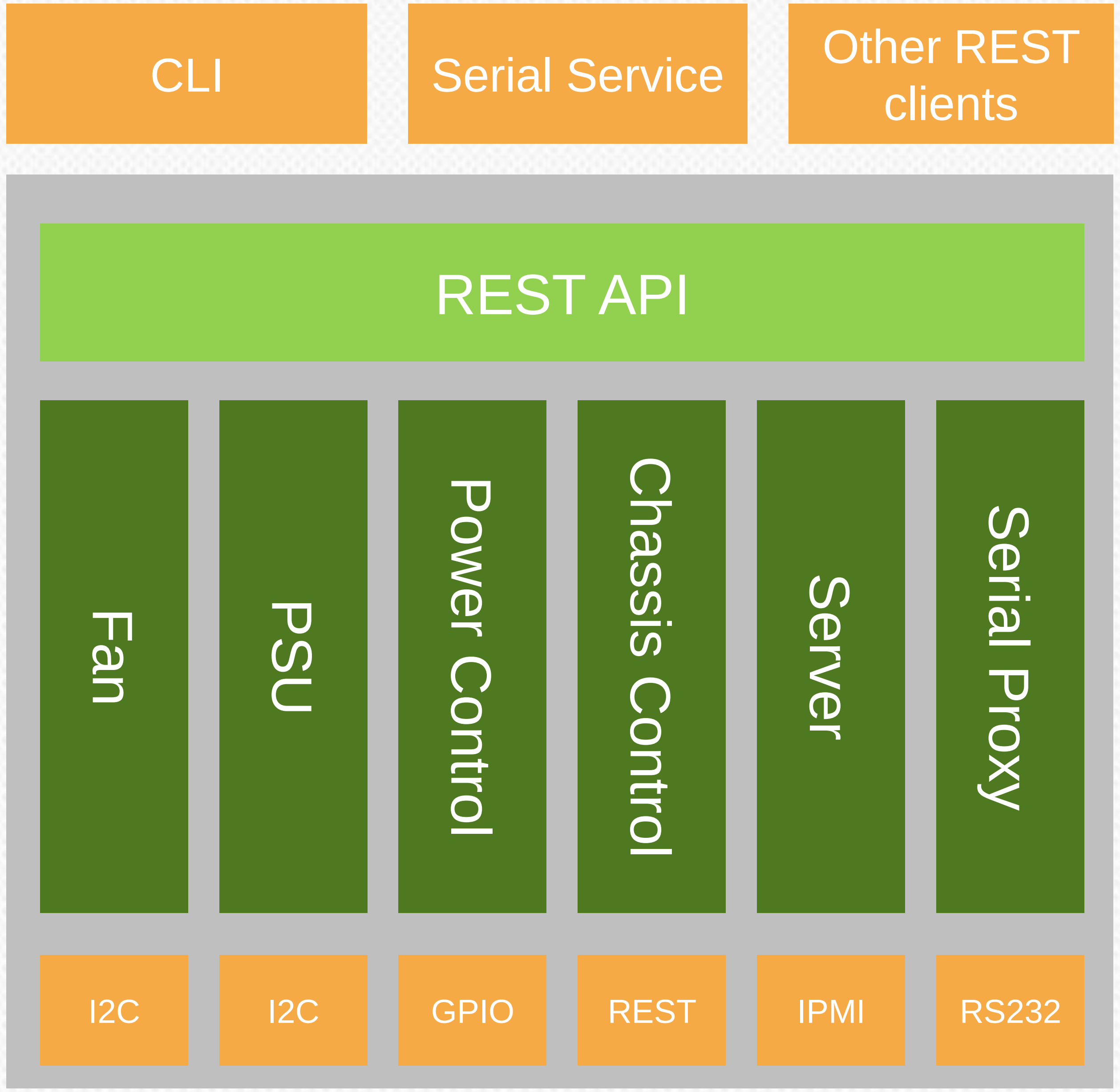
- Managed Devices: Fans, PSU, Power, Blades (IPMI), TOR, Auxiliary power
- RESTful API

Command Line Service

- Provide a command line interface to the CM Service
- Runs anywhere in the network where CM service is reachable

Serial Service

- CLI over RS-232
- Used for bootstrap and bare-metal provisioning



Chassis manager: REST API

GetBladePowerReading Sample Response:

https://localhost:8000/GetBladePowerReading?bladeId=1

Success:

```
<BladePowerResponse
          xmlns="http://schemas.datacontract.org/2004/07/Microsoft.GFS.WCS.Contracts"
          xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <HttpStatusCode>200</HttpStatusCode>
 <bladeResponse>
        <bladeNumber>1</bladeNumber>
        <powerReading>308</powerReading>
  </bladeResponse>
  <error> </error>
  </BladePowerResponse>
```

Compute Blade management

Soft Power Management: On/Off, Default Power setting

Power Capping: Set/Enable/Disable

LEDs: LED ID on/off, Attention LED on/off

Blade Info/Health: FRU, Blade status, Blade health (DIMM, CPU, HDDs, PCIe), SEL

Blade Serial Console: For debugging purposes

Boot Order

Sensors: Temperature, PWM

FW updates: BIOS, BMC, Option ROMs (NIC, HBA) are all updated in-band

⋮

SCSI enclosure management

🟦 OB/IB

🟩 IB

Chassis management

- Chassis Information: Blade, Fans, PSUs
- Chassis Health: Blade, Fans, PSUs
- Chassis Log: get/clear
- LEDs: LED ID on/off, Attention LED on/off
- Chassis User Management: Add/Remove/Update
- NIC: Get/Set for bootstrapping

JBOD management

Hard Power Management: On/Off, Default Power setting

LEDs: LED ID On/Off, Attention LED On/Off

Blade Info/Health: FRU, Blade status, Blade health (HDDs status, link speed), SEL

JBOD Serial Console: for Debugging purposes

Command line interface

REST client application to provide quick access to Chassis management functionality

One to One mapping between CLI commands and REST API

Provides VT100 emulation over REST for serial console redirection

Launching WCSCLI

- wcscli –h <hostname> -p <port> -s<SSL encryption option> [[-u] <username> [-x] <password>] [-b <batch_file_name>]

Command line interface - example

```
wcscli –getchassisinfo [-s] [-p] [-c] [-h]
```

-s – Show information about blades

-p – Show information about power supplies

-c – Show chassis manager information

-h – Help, display the correct syntax

Command line interface – example cont.

wcscli# wcscli getchassisinfo –s –p –c

Sample output:

```
== Compute Nodes ==
#      | Name  | GUID     | State | BMC MAC       | Completion Code
1      | BLADE1        | 71cd4e40-a900-11e1-9856-089e013a37e8  | On    | DeviceID: 0MAC        Address:
08:9E:01:22:FB:42     | Success
….
== Power Supplies ==
#      | Serial Num    | State | Pout (W)      | Completion Code
1      | 46-49-51-44-31-32-33-37-30-30-31-31-32-33     | On    | 194   | Success
…
== Chassis Controller ==
Firmware Version        : 02.02
Hardware Version        : 1
Serial Number           : 33333333
Asset Tag               :
IP Address              : 192.168.100.23
IP Address Source       : 192.168.100.8
System Uptime           : 00:21:32.5127429
```

Wiring diagram for 48 blades: half rack

Power

- CM 1 and 2 are cross wired
- TOR Power is controlled by adjacent rack CM

Network Bootstrap

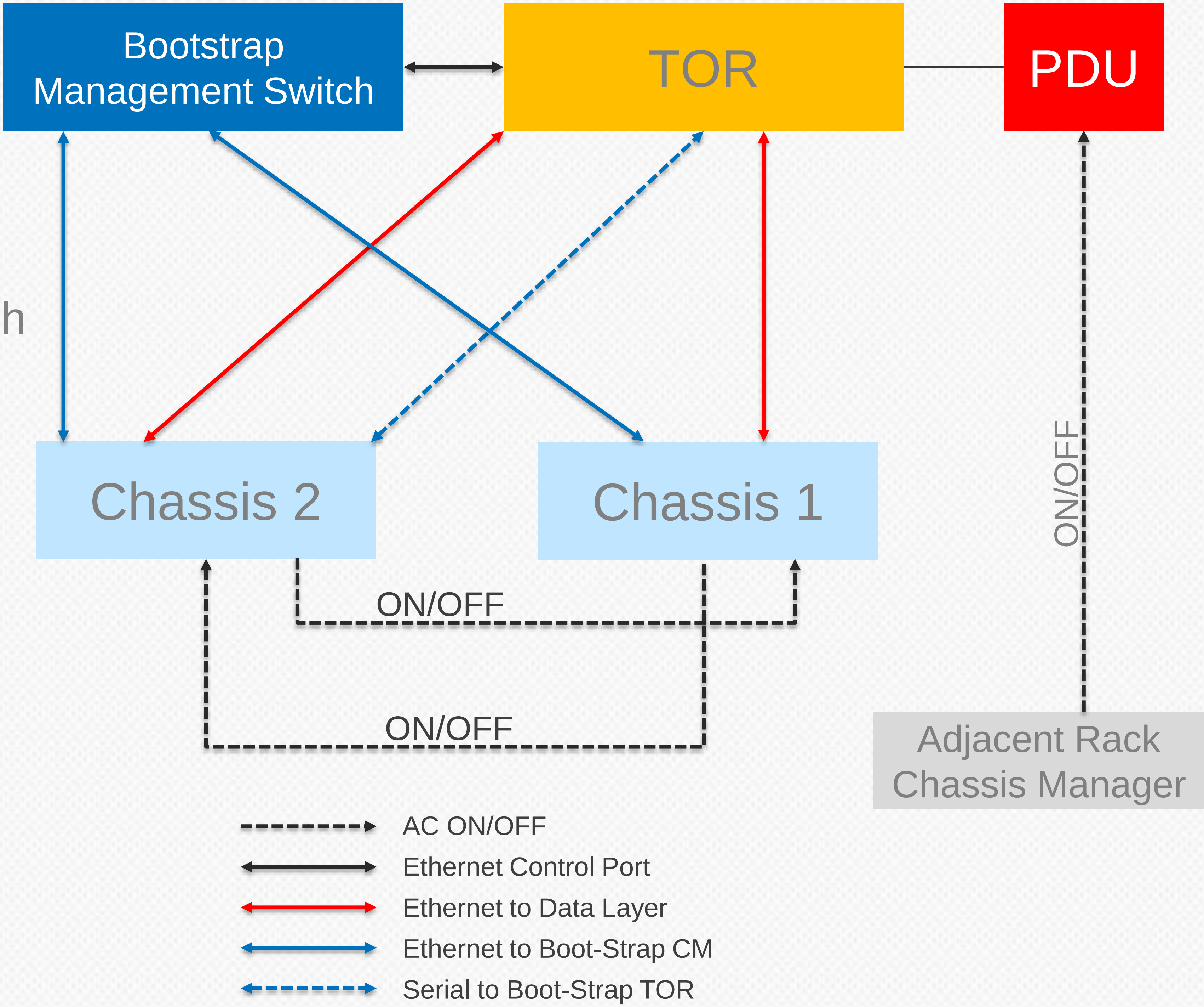
- Nic1 of CM 1 and 2 are attached to Bootstrap switch
- Serial port 1 of CM 2 is attached to TOR for Bootstrapping TOR

Data Plane

- Nic2 of CM1 and 2 are attached to TOR

Pre-Deployment state

- TOR is not configured
- CM are pre-imaged, but image not trusted
- Only connectivity is through bootstrap switch



Bootstrap Management Switch

TOR

PDU

Chassis 2

Chassis 1

ON/OFF

ON/OFF

ON/OFF

Adjacent Rack Chassis Manager

AC ON/OFF

Ethernet Control Port

Ethernet to Data Layer

Ethernet to Boot-Strap CM

Serial to Boot-Strap TOR

Example walkthrough: rack bootstrap

1. Connect to CM2 through bootstrap switch

2. Turn off/on CM1, PXE boot CM1, update BIOS, FW, OS image, CM service image, Configure CM1

3. Connect to CM1 through bootstrap switch

4. Turn off/on CM2, PXE boot CM1, update BIOS, FW, OS image, CM service image, Configure CM2

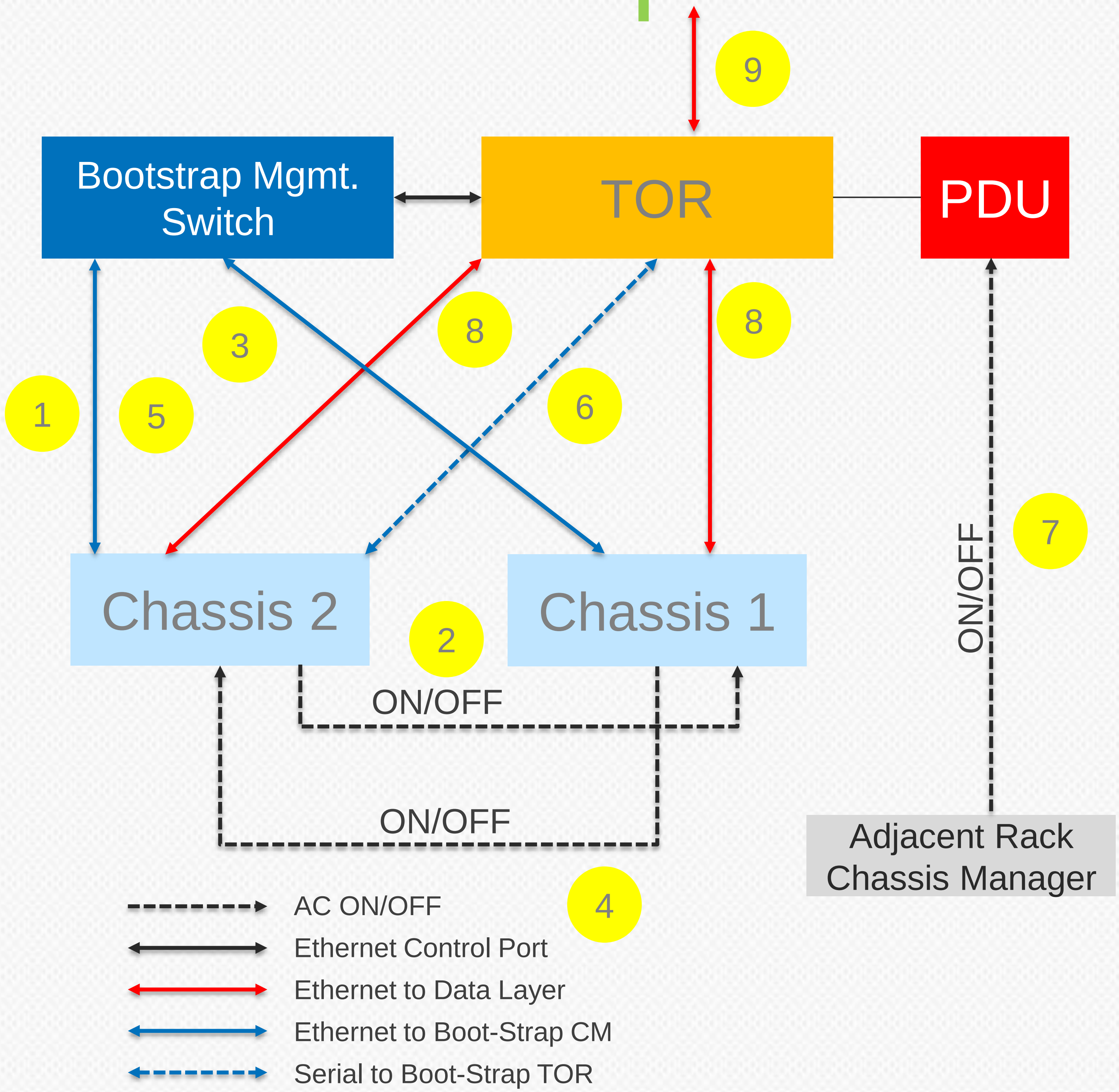
5. Connect to CM2 through bootstrap switch

6. Through CM2 serial proxy connect to TOR, configure TOR

7. Reboot TOR from adjacent Rack CM

8. Data connectivity through TOR to CM1 and CM2 is established

9. Server bootstrapping starts



Scenarios: cable check

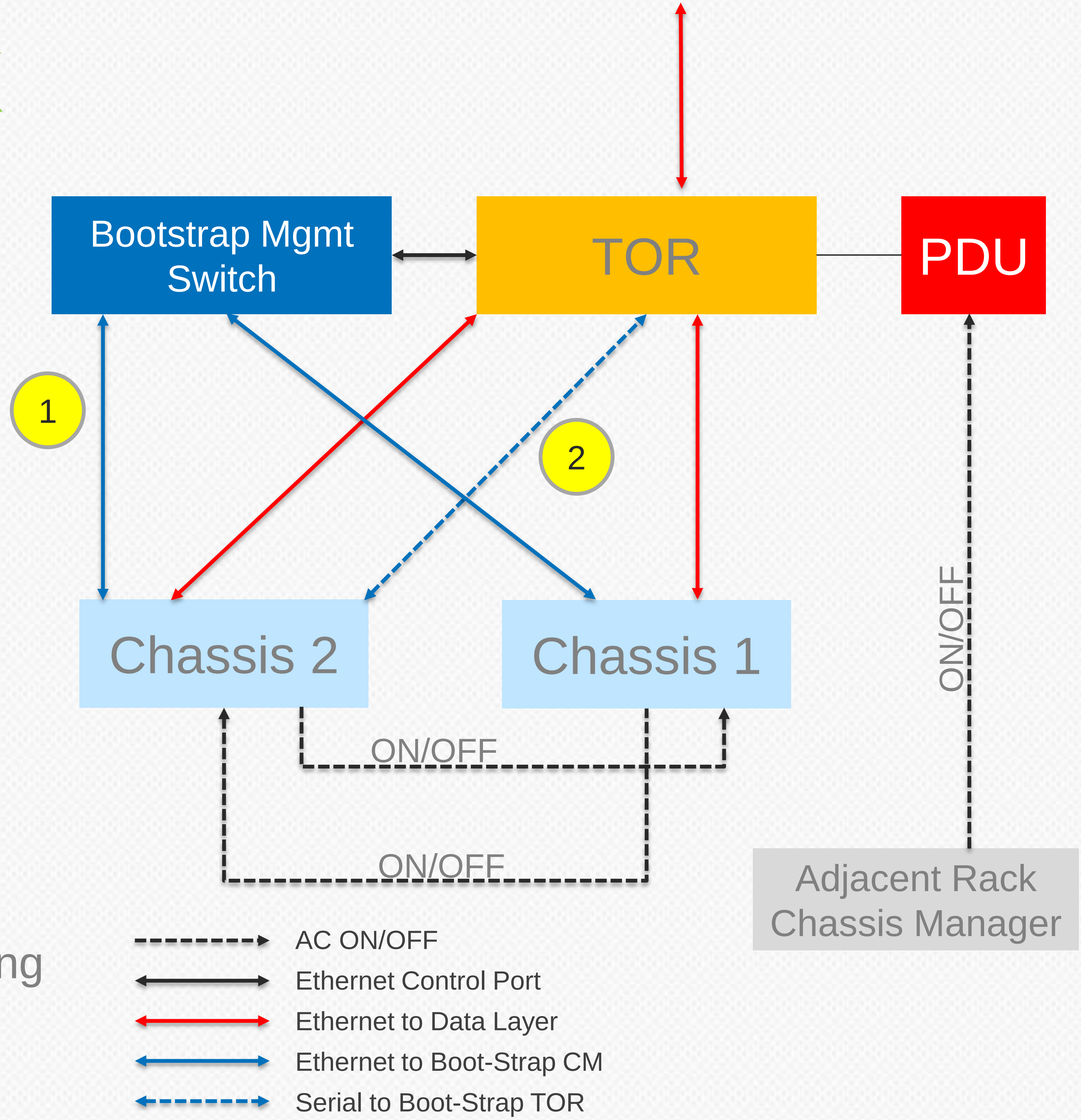
Cable check will require 4 steps:

1. Run “getchassisinfo” command
   - Build the node#/MAC map1

2. GET ARP data from TOR through Serial
   - Build the Port#/MAC map2

3. Join both tables
   - Use map1 and map2 to build Node#/Port# map3

4. Validate
   - Compare map3 to a provided map of expected cabling



Chassis manager development

Code is in Github:

- Project name: https://github.com/MSOpenTech/ChassisManager
- All tools to enlist and build are free of charge (.NET 4.0, VS Express)
- Test libraries will be added later to facilitate functional and conformance testing

Chassis manager development

License: Apache 2.0

Development Model:

- We encourage community contributions
- Will define a process to review and accept contributions and published on manageability website

More information: Technical breakouts

Technical Workshop	Presenter
Management Software Overview	Badriddine Khessib, Director
Hardware Overview	Mark Shaw, Director
Blade Overview – Compute & Storage	Martin Goldstein, Principal Systems Architect
Chassis Manager Hardware Overview	Bryan Kelly, Sr. Platform SW Engineer

Visit the Microsoft booth for live demos by the subject matter experts

Microsoft cloud server spec: OCP contribution

Source Code

Chassis management source code through Open Source

```
/// <summary>
///  Gets Fan speed in RPM
/// </summary>
/// <param name="fanId">target fan Id</param>
/// <returns>Fan speed in RPM</returns>
internal FanSpeedResponse GetFanSpeed(byte fanId)
{
```

Specifications

Chassis, Blade,
Chassis Manager, Mezzanines,
Management APIs

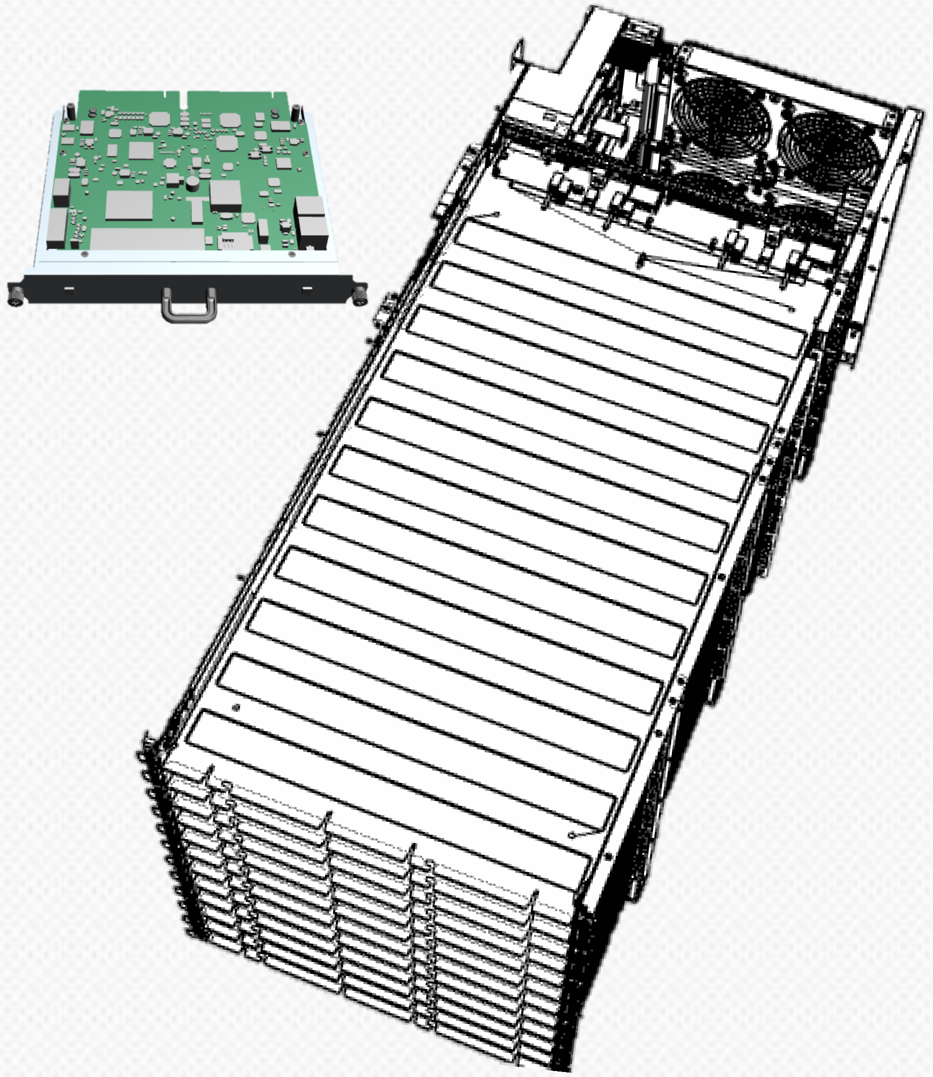
OPEN Compute Project

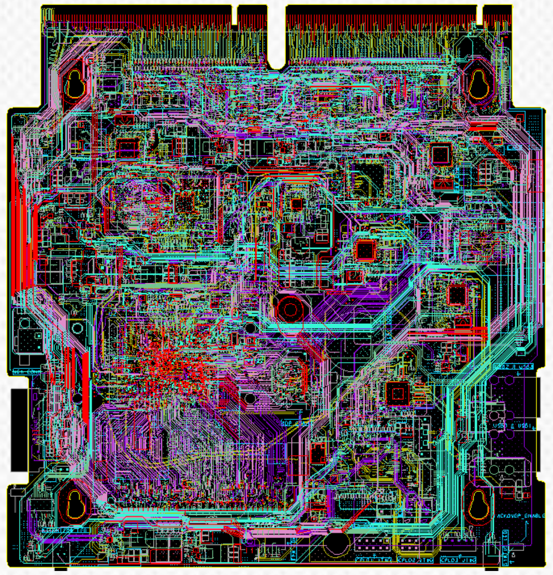
Microsoft Cloud Server System Specifications Hardware: Chassis

MICROSOFT CONFIDENTIAL

Mechanical CAD Models

Chassis, Blade, Chassis Manager,
Mezzanines





Board Files & Gerbers

Chassis Manager, Tray Backplane,
Power Distribution Backplane

Microsoft datacenter resources

Microsoft Datacenters Web Site & Team Blogs

- www.microsoft.com/datacenters

Windows Azure

- http://www.windowsazure.com

Office 365

- http://www.office365.com

Q & A

Microsoft