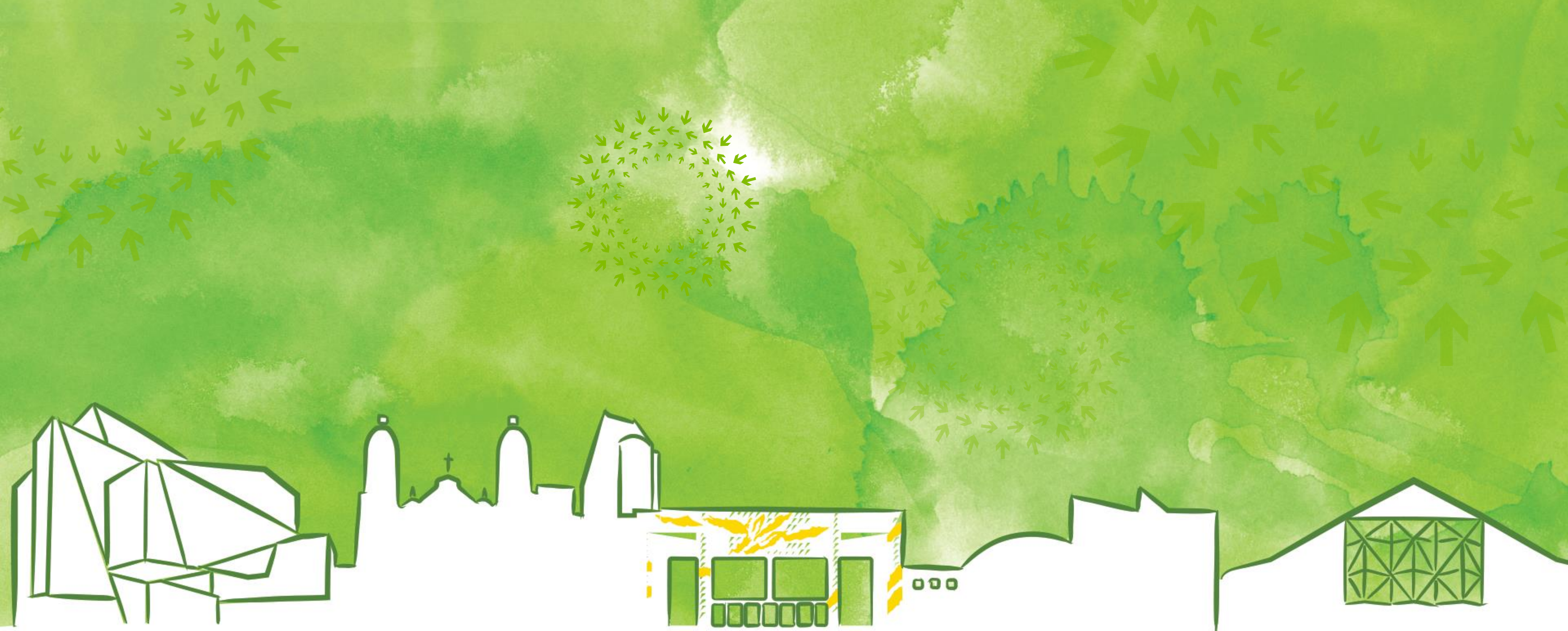


OPEN

Compute Project





OCP U.S. SUMMIT 2016 March 9-10 | San Jose, CA

Platinum Sponsor

OCP U.S. SUMMIT 2016

OpenBMC Updates

Tian Fang, Jubin Mehta, Sai Dasari
FACEBOOK INC.

Agenda

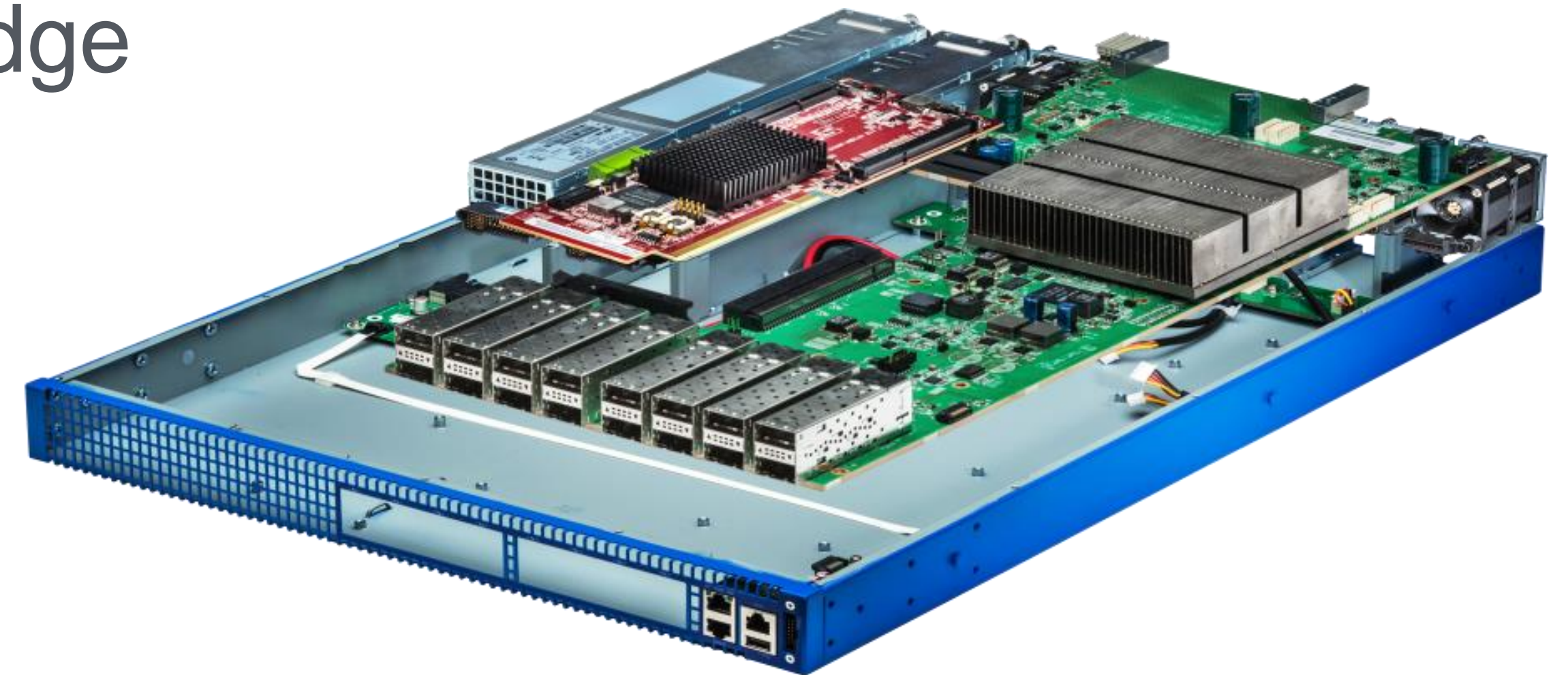
- History
- Current Status
- Deployment
- Storage Enclosure Management
- Multi Node Server Management
- User Interface – SSH Utilities, REST API
- What's next
- Q & A

OpenBMC history

→ Started in Facebook in 2014 Jun

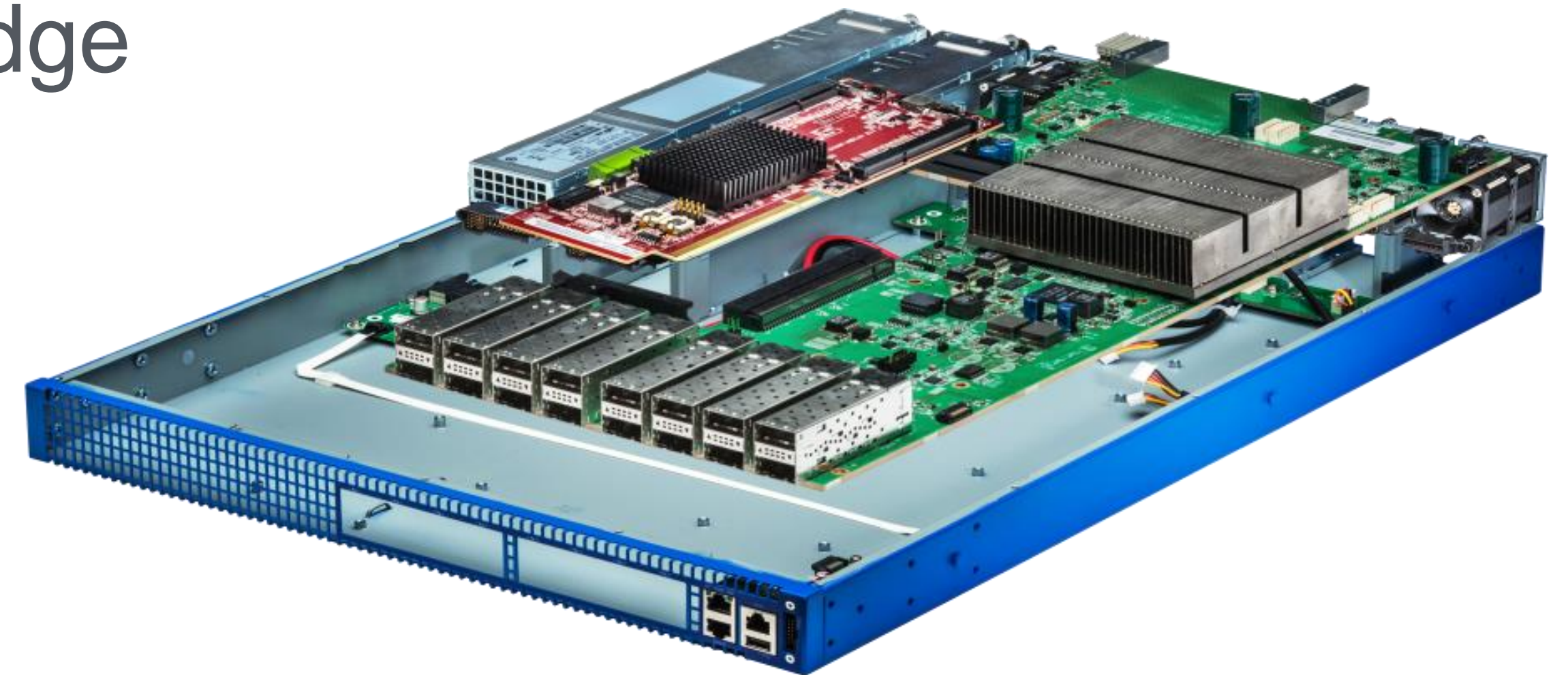
OpenBMC history

- Started in Facebook in 2014 Jun
- Developed for Facebook TOR, Wedge
 - AST1250, 256M DDR3, 32M SPI Flash
 - I2C, PMbus, SMBus
 - UART, GPIO, PWM/TACHO
 - MAC
 - USB Gadget
 - SPI, EEPROM
 - Temperature sensor, voltage sensor
 - ADC, PSU



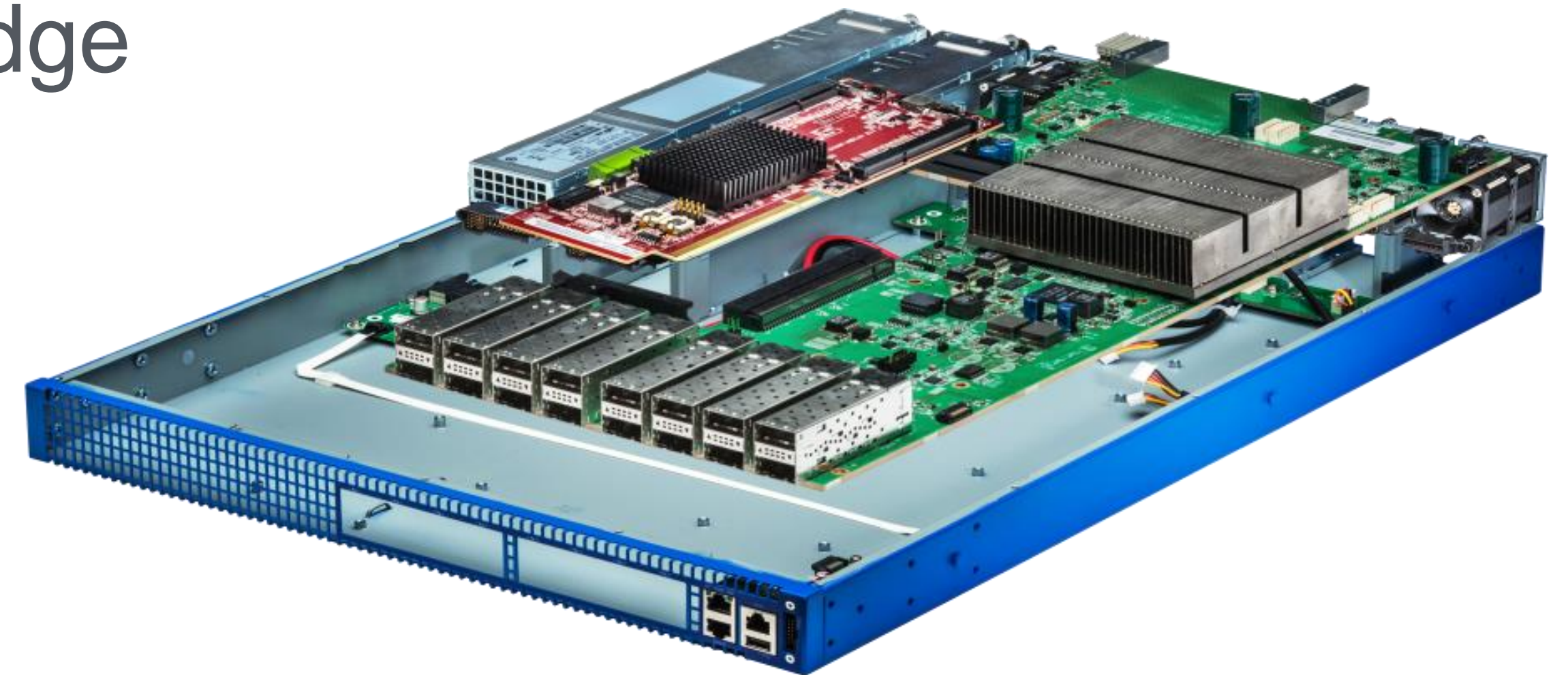
OpenBMC history

- Started in Facebook in 2014 Jun
- Developed for Facebook TOR, Wedge
- Used Yocto 1.6.1
 - Aspeed SDK 0.64
 - U-boot 2013.07
 - Kernel 2.6.28.9
 - Busybox
 - OpenSSH
 - Python 2.7



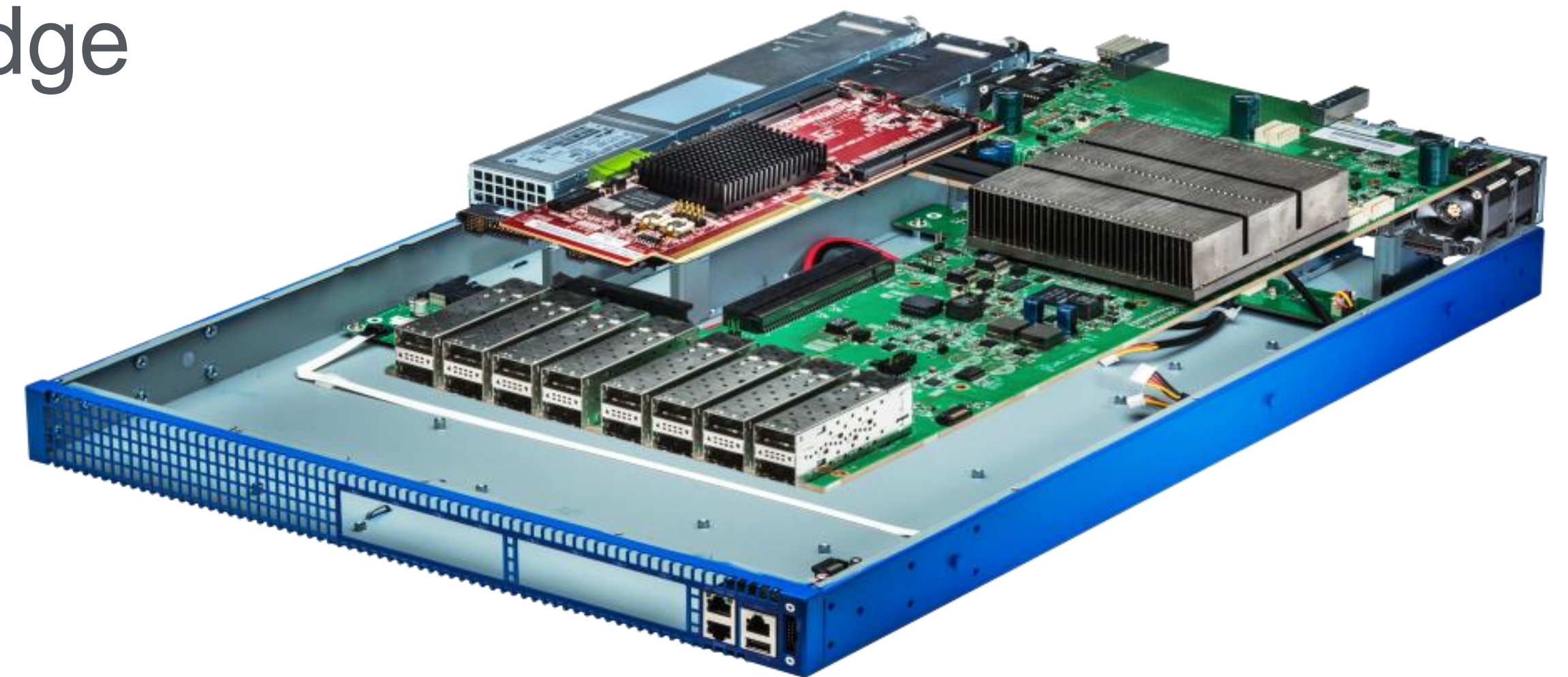
OpenBMC history

- Started in Facebook in 2014 Jun
- Developed for Facebook TOR, Wedge
- Used Yocto 1.6.1
- Open sourced through github
 - <https://github.com//facebook/openbmc>



OpenBMC history

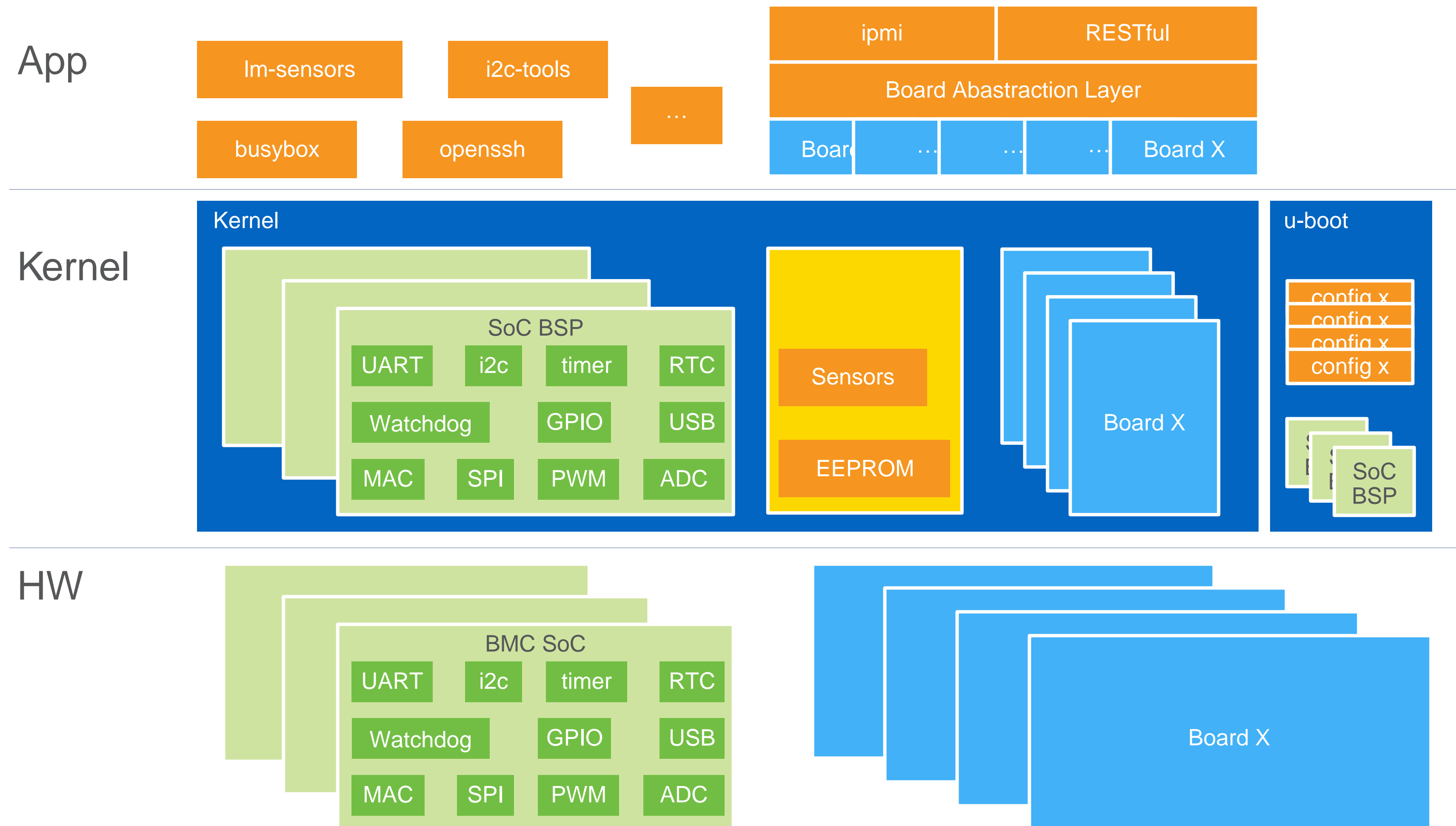
- Started in Facebook in 2014 Jun
- Developed for Facebook TOR, Wedge
- Used Yocto 1.6.1
- Open sourced through github
- Presented in 2015 OCP



3rd-party BMC vs. OpenBMC

	3rd Party BMC	OpenBMC
Source Code	<ul style="list-style-type: none">• Proprietary, Closed	<ul style="list-style-type: none">• Designed by Facebook• Open Source @ Github
User Interface	<ul style="list-style-type: none">• ipmitool - Raw Bytes	<ul style="list-style-type: none">• SSH to BMC Linux Shell• REST API - JSON objects
Security	<ul style="list-style-type: none">• RMCP+ over UDP• Known vulnerabilities	<ul style="list-style-type: none">• Secure Shell (ssh)• http(s)
Authentication/Authorization	<ul style="list-style-type: none">• BMC-centric username/password database	<ul style="list-style-type: none">• Centralized certificate based
Multi-Node Management	<ul style="list-style-type: none">• Need to virtualize BMC with multiple IP addresses	<ul style="list-style-type: none">• Supported Natively with node/slot number

OpenBMC architecture



OpenBMC now

- Migrated to Yocto 1.8.1
- Support 5 different boards
- Thousands of boards running OpenBMC deployed in production

OpenBMC deployment

- Manage OpenBMC as server
- Push OpenBMC as SW
- Spatula framework to sync configuration
- Use ssh certs to manage access permission
- Use RESTful to access FRU and sensors

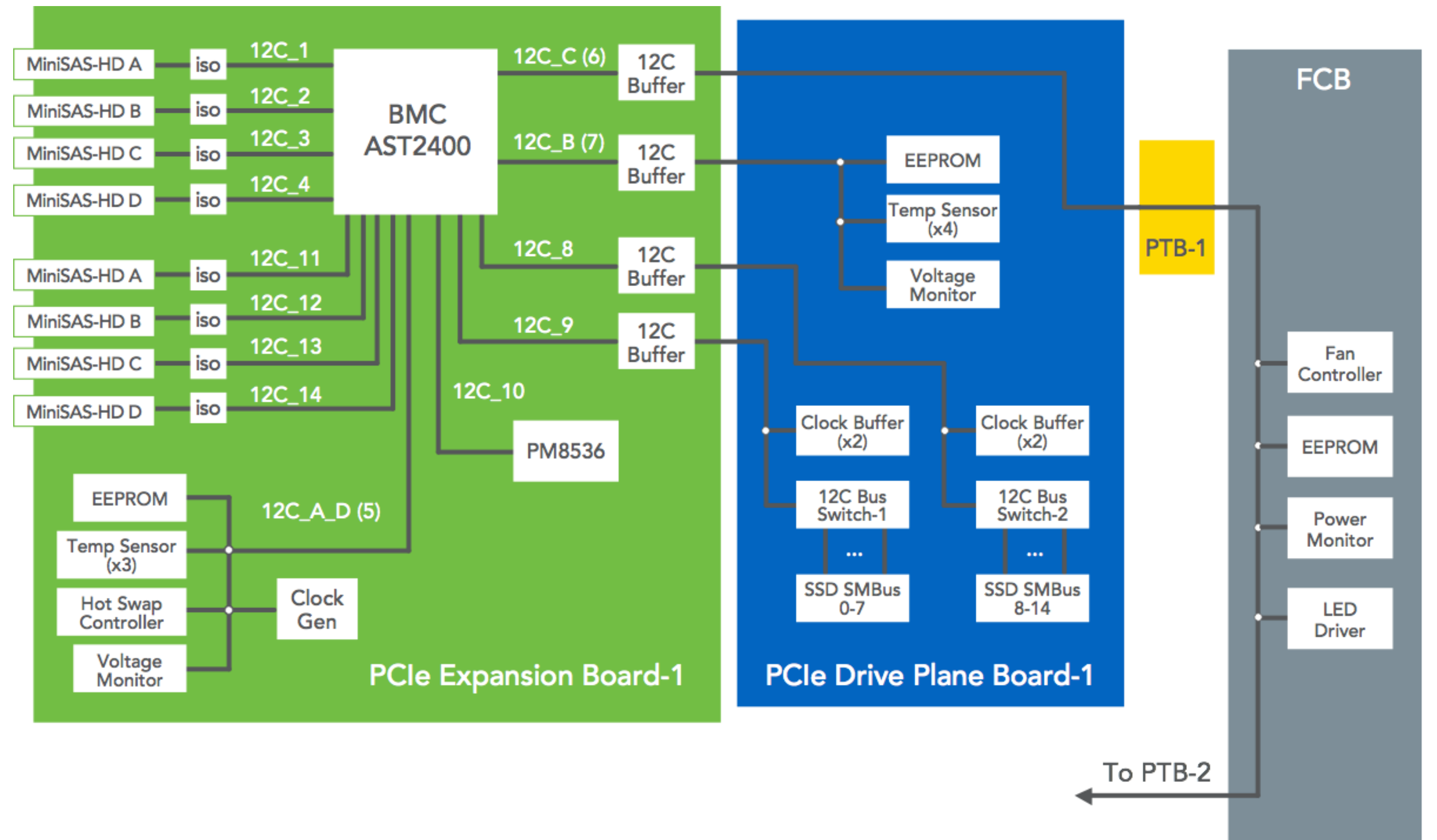
OpenBMC on Lightning

- NVMe-based OCP
Open Vault
Storage “knox”
- Supports upto
 - 30 (x4) SSDs
 - 60 (x2) SSDs



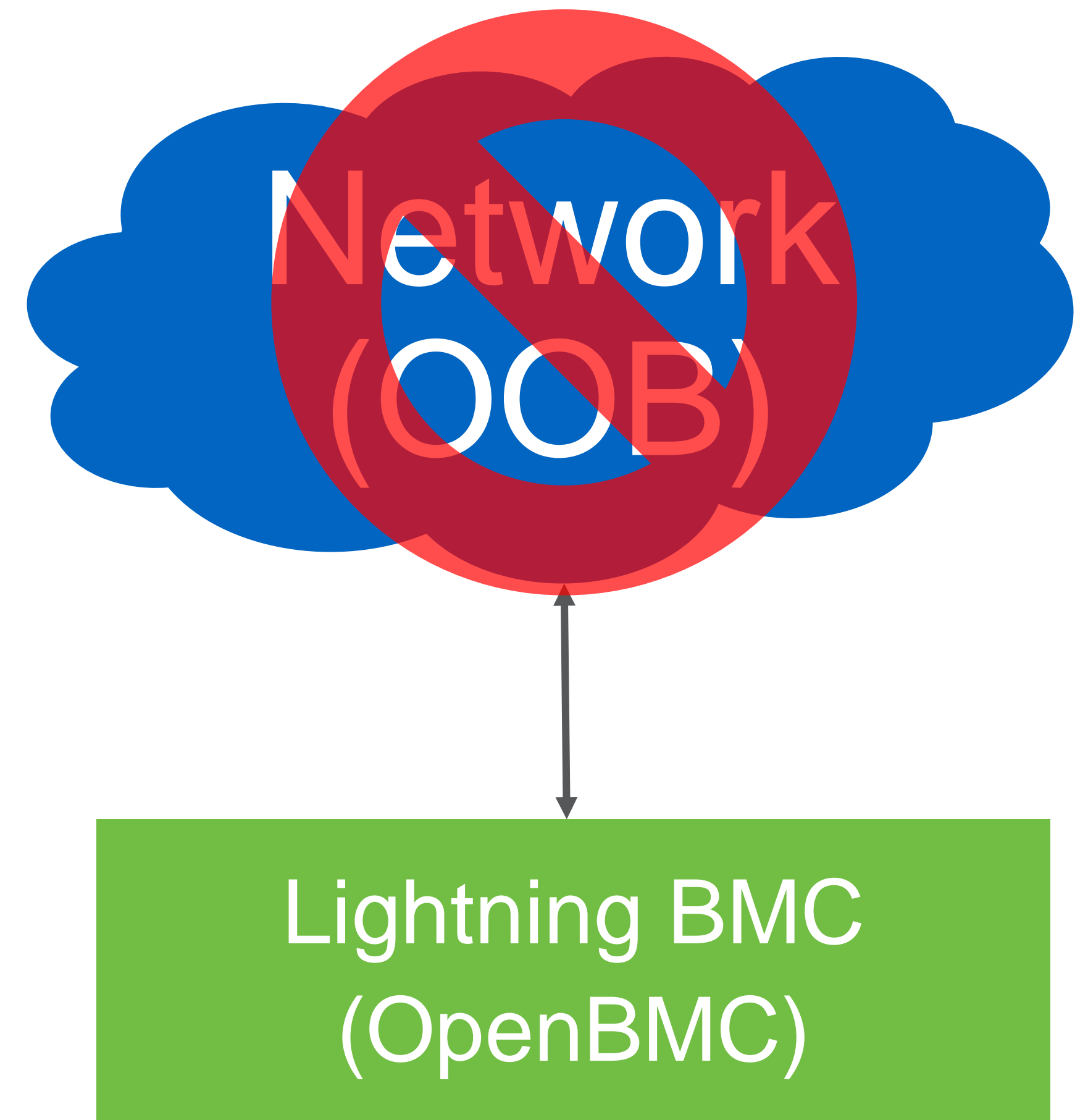
Why OpenBMC on Lightning?

- Enclosure Management
- Chassis Management
- Multi-host support



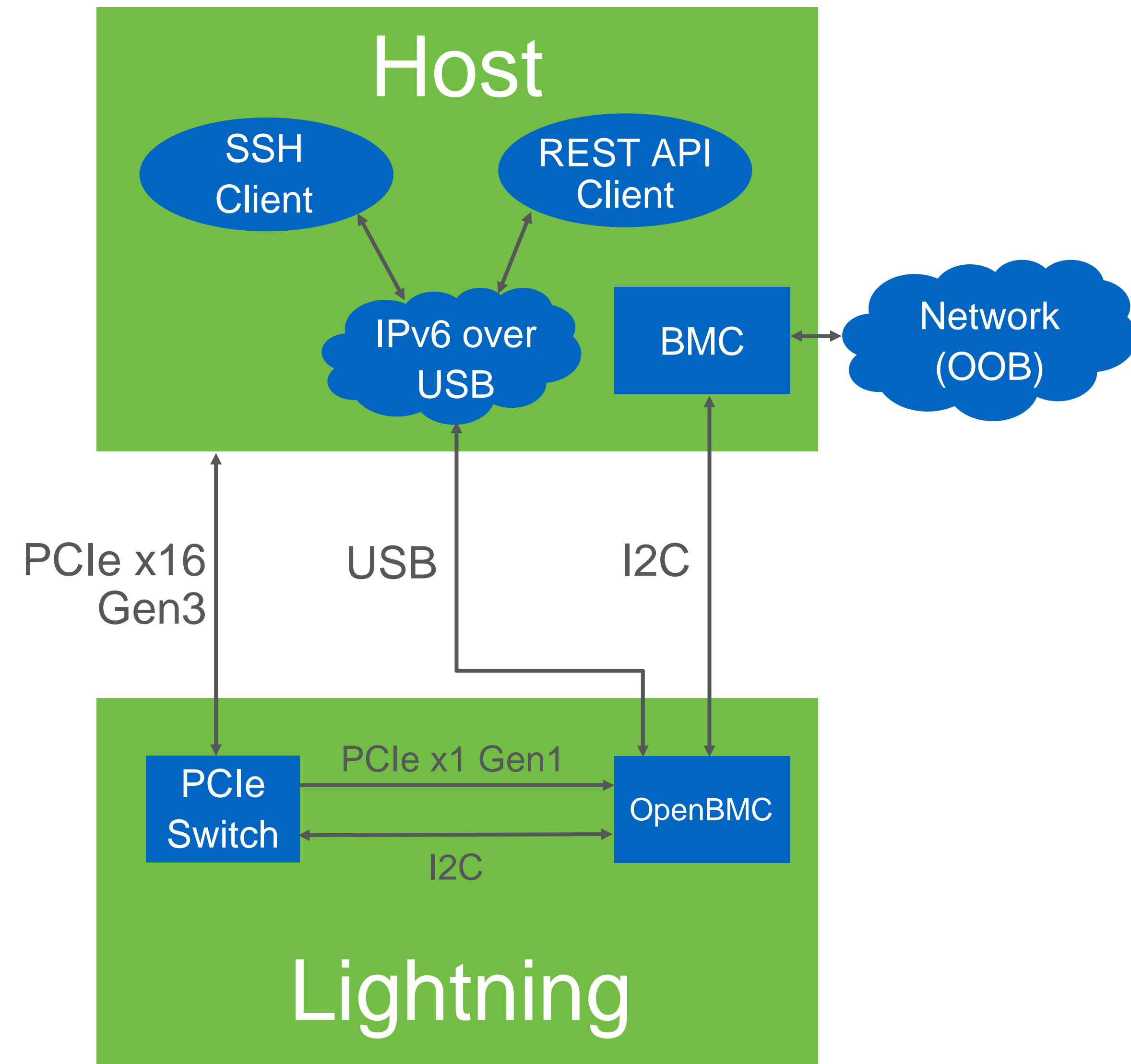
No sideband network

- No NIC on Lightning
- Storage and Hosts mapping issue
- BMC compute core too weak for Infrastructure tools.

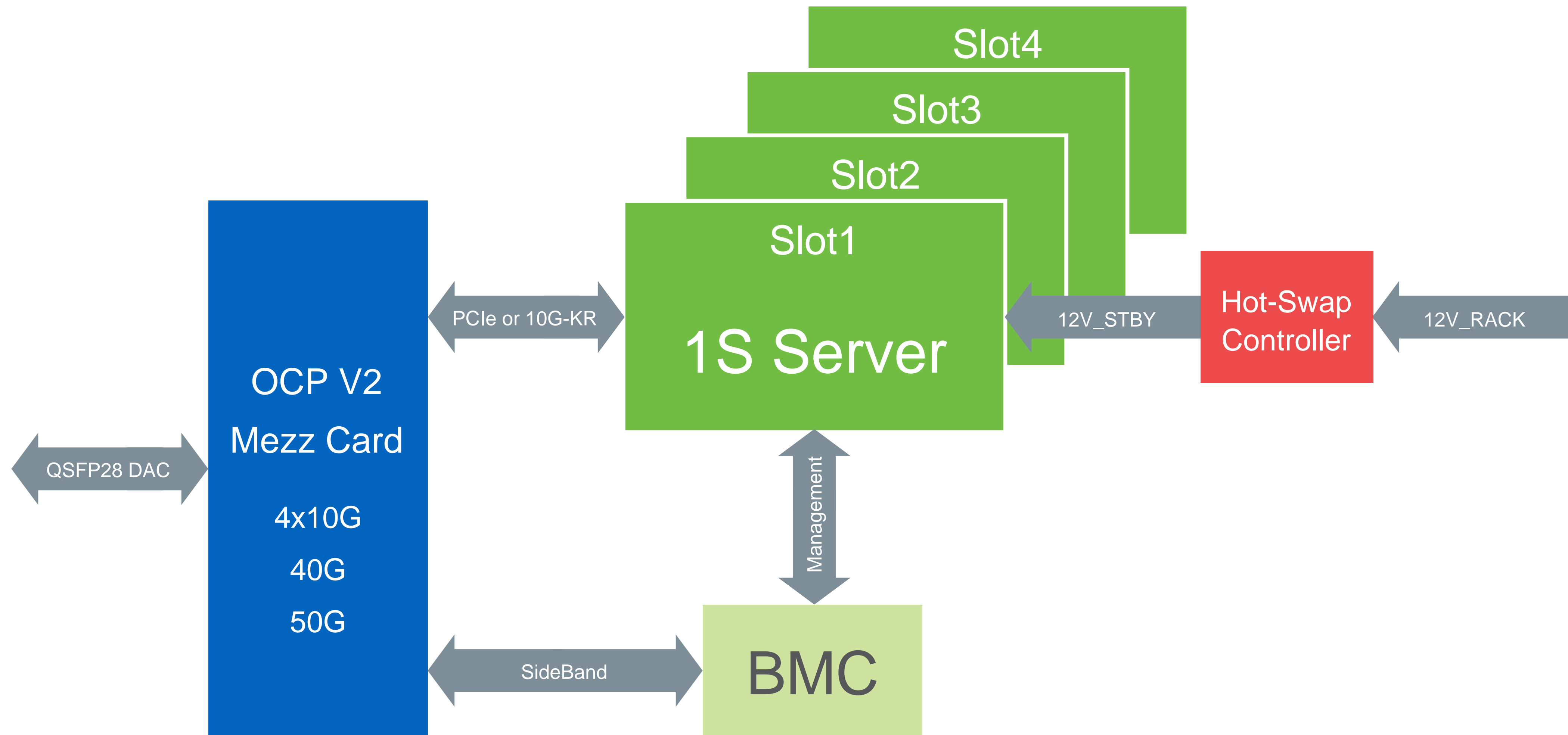


OpenBMC Interfaces

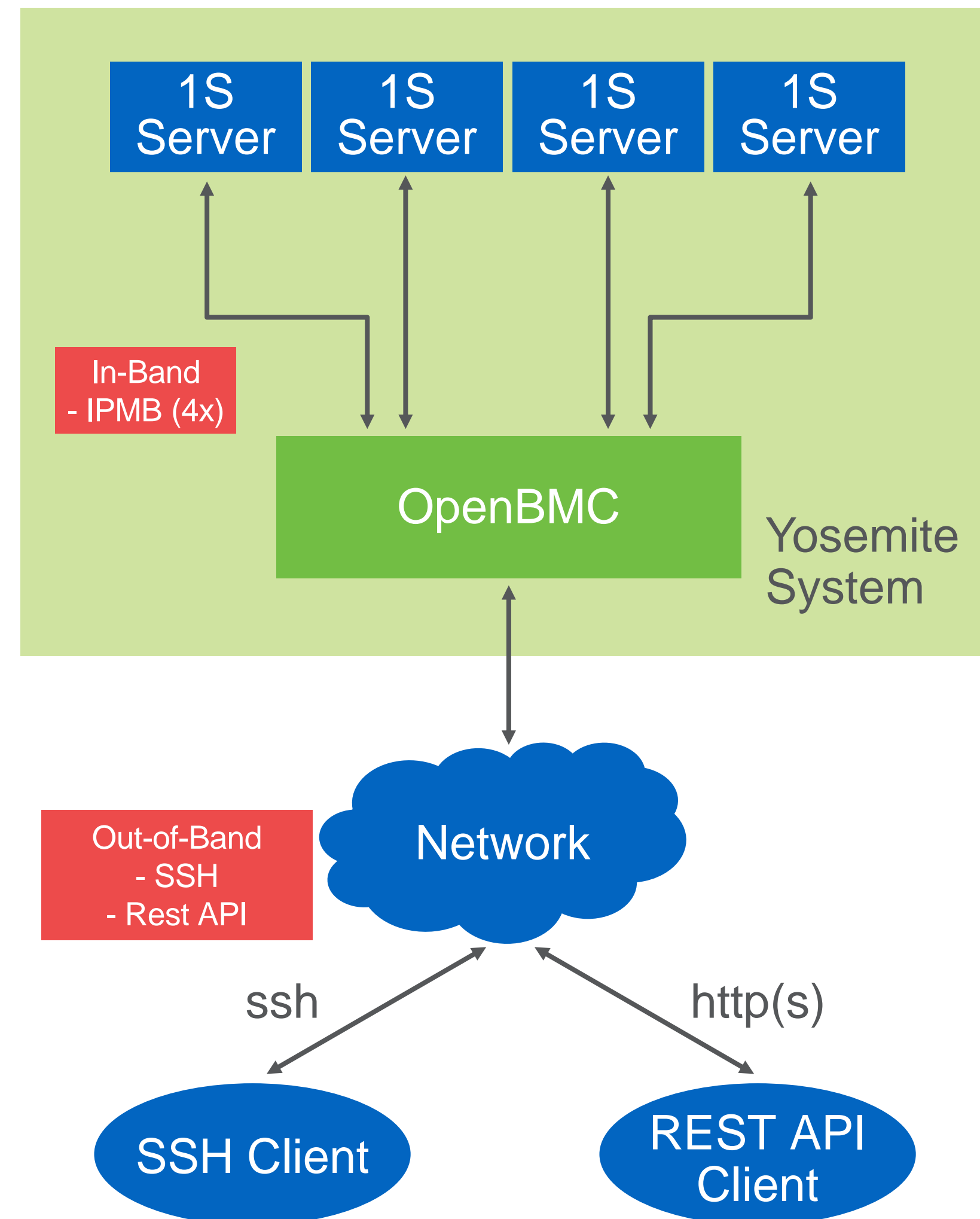
- USB bus from Host to Lightning BMC (In-band only)
- I2C bus from Host BMC to Lightning BMC
- PCIe x1 (Gen1) from Host to Lightning BMC (In-band only)



Yosemite platform architecture



OpenBMC user interface



User Interface

→ SSH

- Linux Shell Access
- Utilities

→ REST API

- http/https
- RESTful Resource End Points

Utilities

→ FRU Information

- fruid-util [all, slot#, spb, nic] [--dump | --write] <file>

→ Power Control/Status

- power-util [slot#] [status, graceful-shutdown, off, on, cycle, 12V-off, 12V-on, 12V-cycle]
- power-util sled-cycle

→ Serial-Over-Lan

- sol-util <slot#> <--force | --history>
- Ctrl-x to close the session

→ Sensor Readings

- sensor-util [all, slot#, spb, nic] <--threshold> <sensor num>

Utilities (contd.)

→ Error Logs

- `log-util [all | slot# | spb | nic] <--clear|--print>`

→ Firmware Information/Update

- `fw-util <slot#> --version`
- `fw-util <slot#> --update <--cpld|bios|bicbl>`

→ Front Panel Control

- `fpc-util <slot#> --usb`
- `fpc-util <slot#|sled> --identify <on|off>`

→ Configuration

- `cfg-util <dump-all|key> <value>`

Utilities (contd.)

→ Fan Control

- `fan-util <--get|--set> <fan#> <pwm%>`

→ Communication with Intel ME

- `me-util <slot#> <[0..n]data_bytes_to_send>`

→ BMC firmware update

- `flaschcp -v <kernel> /dev/mtd2`
- `flaschcp -v <rfs> /dev/mtd3`
- Reboot

REST API design

→ Resource Model

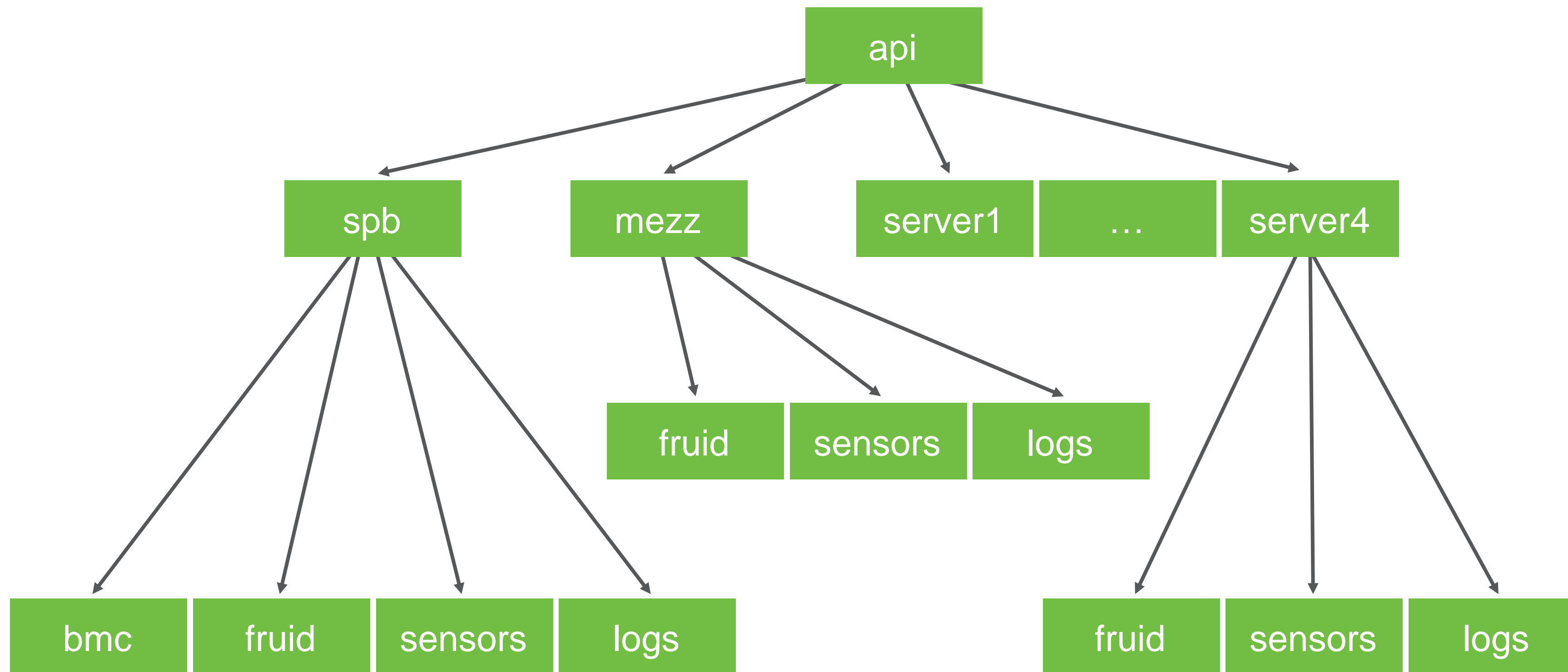
- Simple, Consistent, Discoverable
- Reflect hardware topology

→ Resource Node Attributes

- Information: { Key1: Value1, key2: value2, ... }
- Actions: [action1, action2, action3, ...]
- Resources: [node1, node2, node3, ...]

System resource model

(Yosemite)



RESTful resource endpoints

→ Root Node:

- /api

→ Side Plane Board:

- /api/spb, /api/spb/bmc, /api/spb/fruid, /api/spb/sensors, /api/spb/logs

→ Server Board:

- /api/server3, /api/server3/fruid, /api/server3/sensors, /api/server3/logs

→ NIC Mezzanine Card:

- /api/mezz, /api/mezz/fruid, /api/mezz/sensors, /api/mezz/logs

REST API usage examples

- `curl http://<bmc-ip-addr>:8080/api`
- `curl http://<bmc-ip-addr>:8080/api/spb/bmc`
- `curl http://<bmc-ip-addr>:8080/api/server2/fruid`
- `curl http://<bmc-ip-addr>:8080/api/mezz/sensors`
- `curl -H "Content-Type: application/json" -X POST -d '{"action": "power-on"}' http://<bmc-ip-addr>:8080/api/server3`

Summary

→ What did we learn

- OpenBMC architecture is flexible to support different HW
- Easy to bringup a new board
- Most work focus on board specific changes

→ On-going

- Porting Drivers to Linux Kernel 4.1.x
- New Aspeed BMC chip (AST2500) support

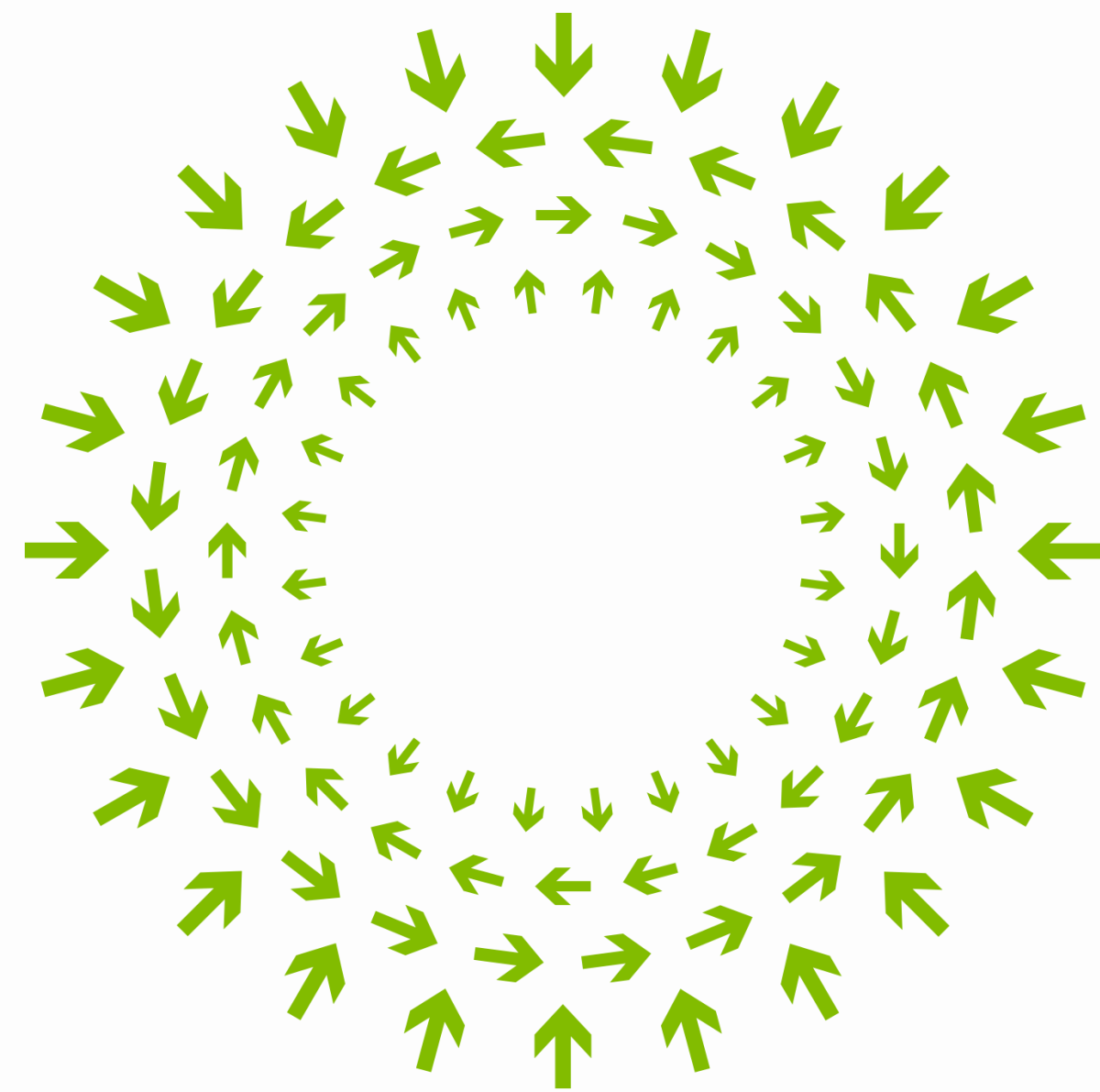
Next steps

→ Challenges

- Driver stability issue
- Better Tooling: Development, provisioning, maintenance
- BMC Vendor's EVK: Limited to evaluation or bring-up
- Full Server h/w: Limited to few firms/engineers

→ OpenBMC Developer Platform

- Base board with AST2500 BMC: similar to EVK
- COM Express connectivity: Choose and plugin Host CPU
- Portwell COMe OpenBMC Carrier Board (Model: PCOM-C660-OCP, PN: 21-F91411-000)
- Enable and Accelerate Open Source innovation



OPEN

Compute Project

