



Optics in Open Networks

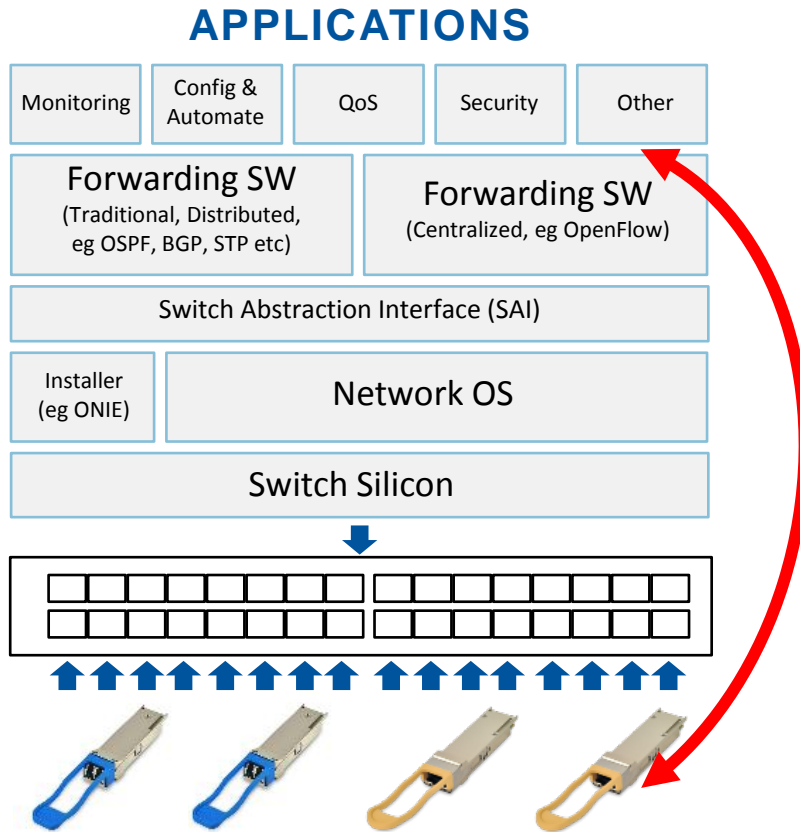
NANOG 72
Atlanta, February 2018

Christian Urricariet
Finisar

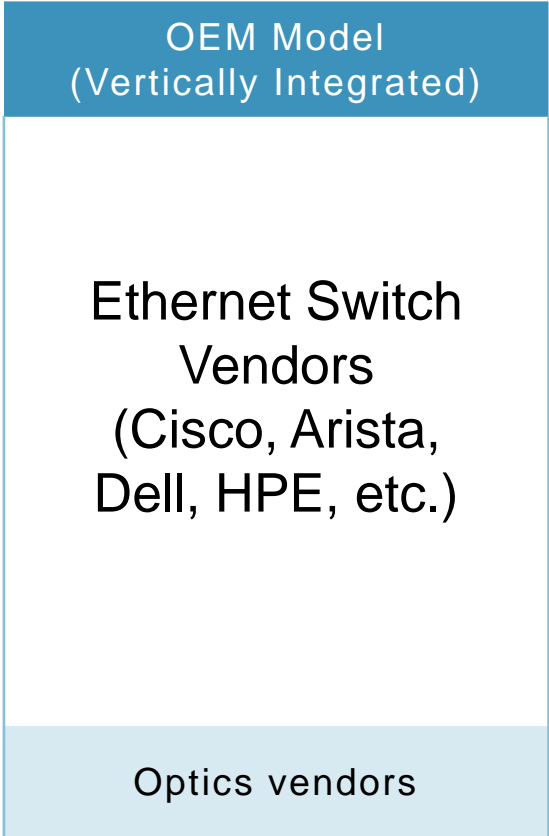
Agenda

- How can optical transceivers be managed in an open networking environment?
- Open standards and interoperability issues...
- Open Optical Monitoring (OOM): What is it and how does it help?
- OOM architecture and some technical details.
- OOM application examples.
- How can you access OOM?
- Latest OOM news.

Traditional Networking Vendors Fully Exploit the Optical Data

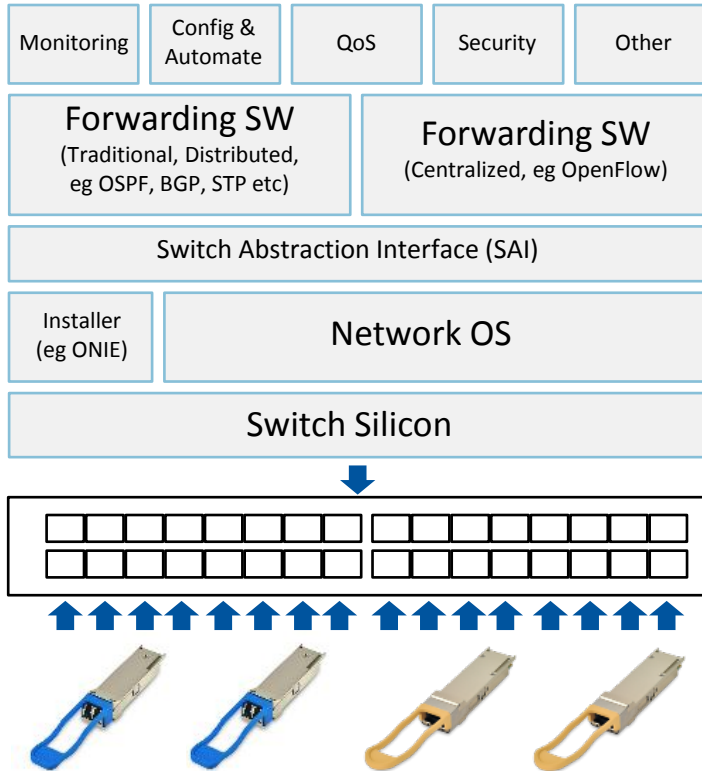


Switch Vendors built all the plumbing to get optical data to the end user



Open Network Ecosystem: Choice, Flexibility and Innovation

APPLICATIONS



Open Switch,
Bare Metal Model

100s of 3rd
Party Apps

Applications

Many 3rd Party
& Open-source

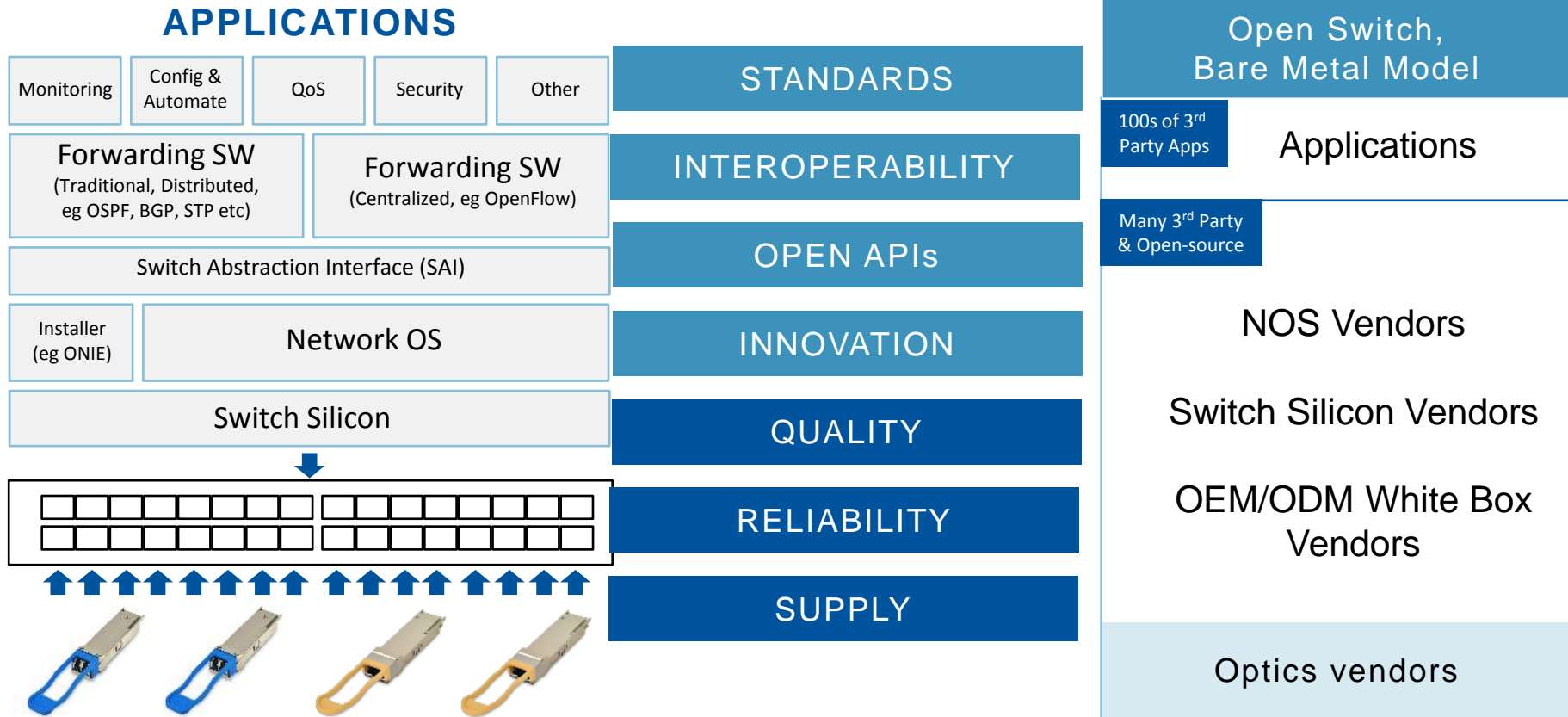
NOS Vendors

Switch Silicon Vendors

OEM/ODM White Box
Vendors

Optics vendors

Optics in Open Networks? Requirements for Success



Data Center Optics Standards are Evolving Rapidly

Industry standards are still very relevant:
IEEE, OIF, T11, SFF, CFP MSA,
CWDM4 MSA, QSFP-DD MSA...

Within the Data Center Rack

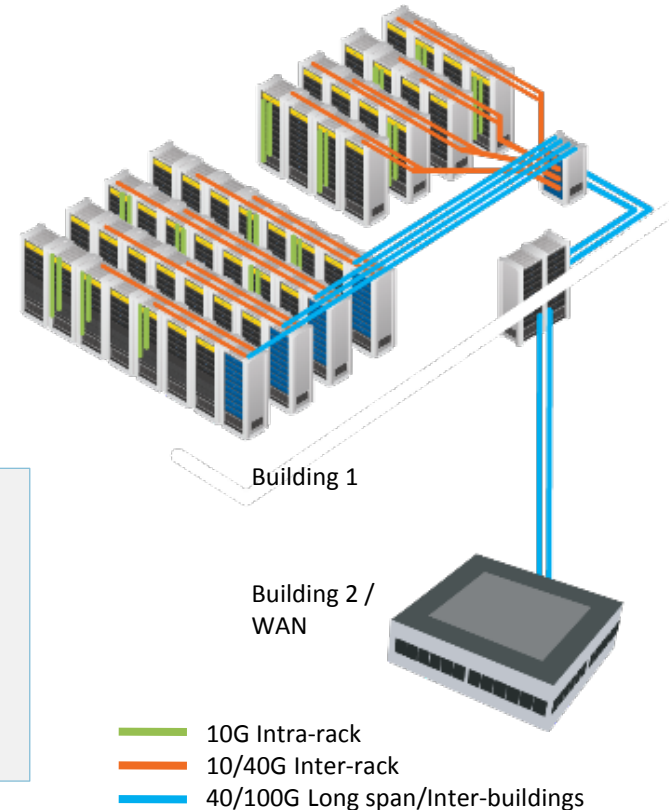
Between Data Center Racks

Long Spans/ DCI & WAN

- **10GE** SFP+ SR/LR now
- **25GE** SFP28 SR/LR deployments starting
- 50GE/100GE (next-gen) being standardized

- **40GE** QSFP+ SR4/LR4/Lite now
- **100GE** QSFP28 SR4/CWDM4/LR4 deployments starting
- 200GE/400GE being standardized

- **100GE** QSFP28 or CFP2/4 now
- LR4, ER4, ER4f, **Coherent ACO/DCO**
- 400G being standardized



Open Networking Interoperability Testing

- OpenCompute.org is leading open switch solution compliance testing.
- Testing dozens of NOS / Switch / Cable / Transceiver Module combinations.
- Proven solutions are added to Open Networking Integrators List.
- Testing occurs weekly at UNH-IOL.
- Latest Plugfest in May 2017 at UNH-IOL focused on 100G CWDM4.



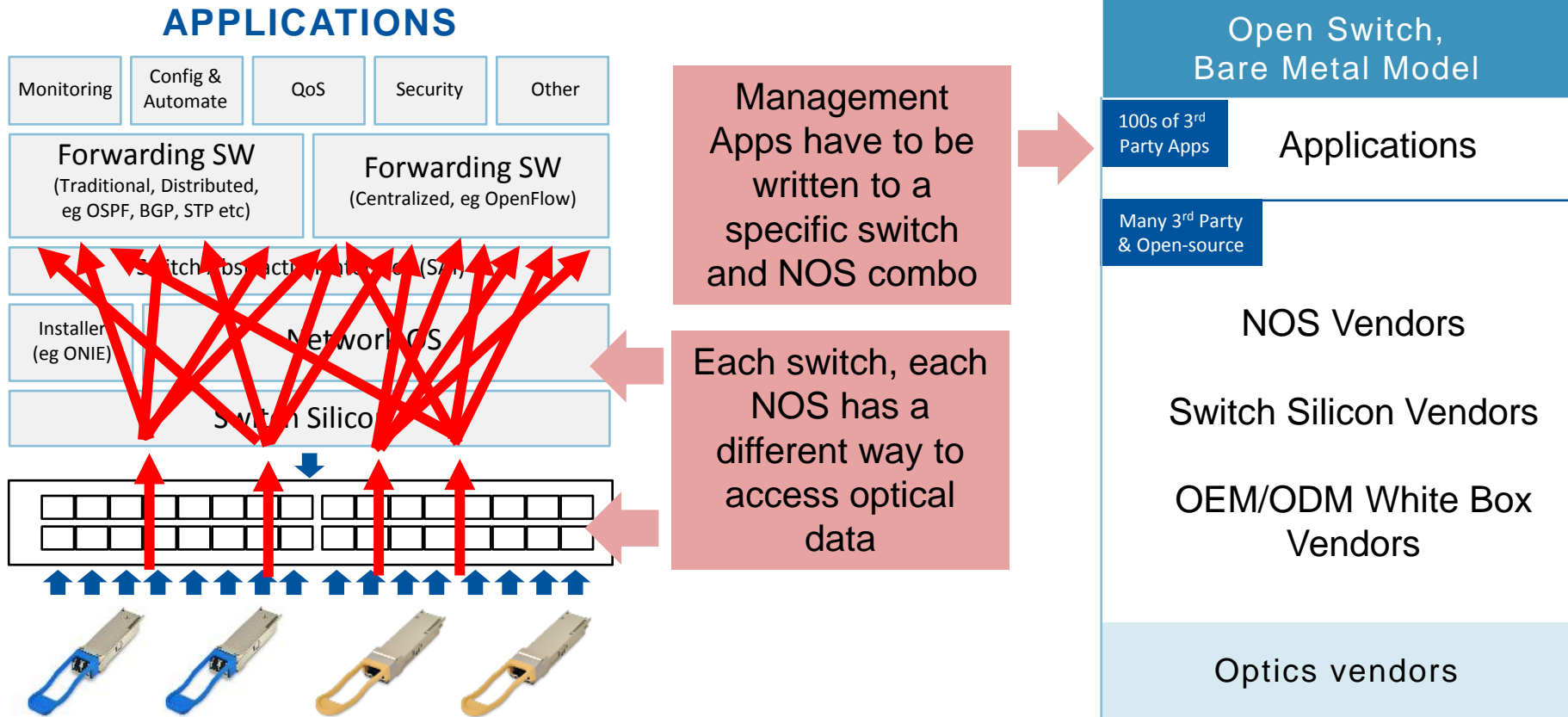
White Box Switch Vendors

NOS Vendors

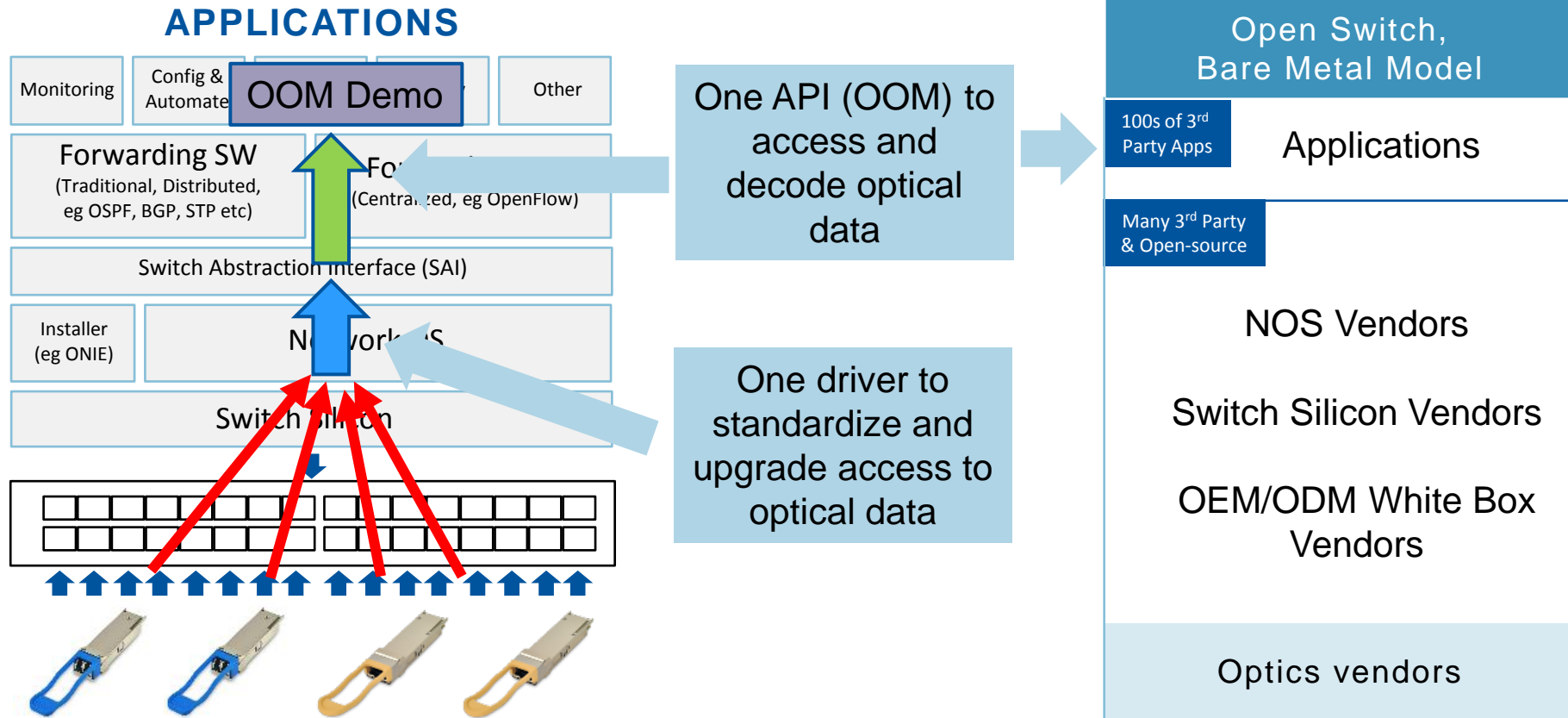
Optics Vendors

Copper Cable Vendors

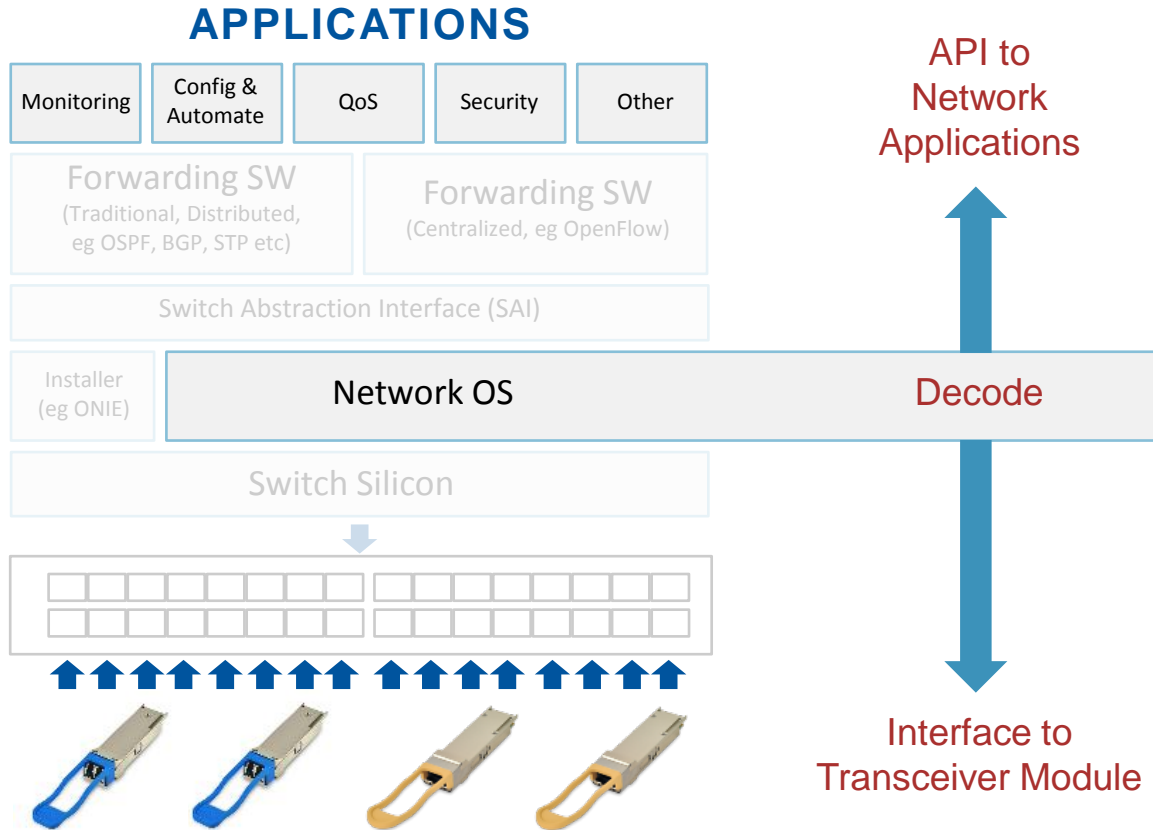
Open Networking: The Data is getting lost in the “Flexibility”



Driving a Common Solution for Open Networking



Open Optical Monitoring and Control (OOM)



During OCP Interop testing at UNH-IOL, there were problems with consistently accessing EEPROM information in optical transceivers.

OOM was kicked off by the OCP Networking group in October 2015 to address these problems.

Project Sponsors:

Accton/Edgecore

Big Switch Networks

Broadcom

Cumulus Networks

Finisar

What is Open Optical Monitoring (OOM)?

OOM is a Python package, providing a standard API to read/write optical transceiver modules.

- EEPROM data encoded/decoded in key/value pairs.

Same API: Any Linux-based NOS, any switch, any transceiver vendor, any transceiver type.

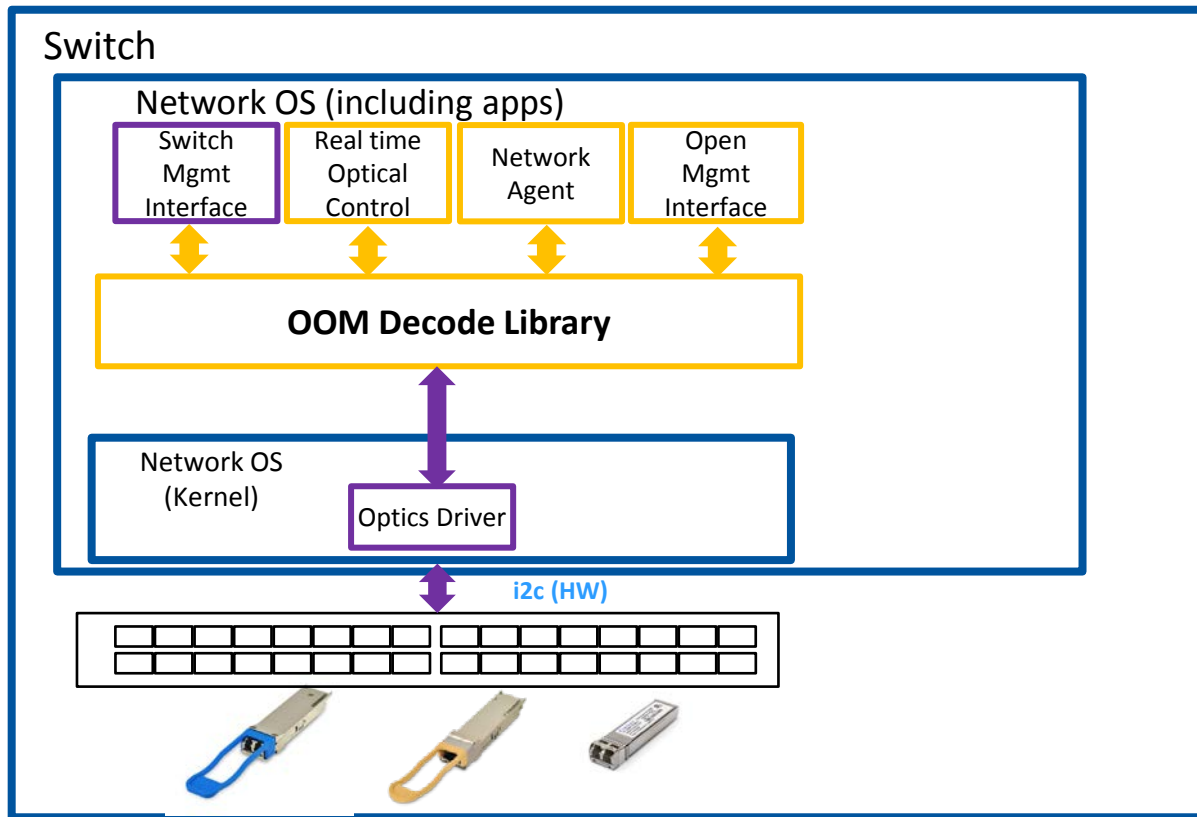
Open Source, easy to maintain, easy to extend.

```
from oom import *
for port in oom_get_portlist():           # enumerate the ports on the switch
    status = oom_get_memory(port, 'DOM') # DOM = voltage, temp, {TX, Rx}Power, bias
    print port.port_name + str(status)
```

```
port0{'VCC': 3.30, 'TEMP': 23.55, 'TX_POWER': 0.57, 'RX_POWER': 0.56, 'TX_BIAS': 7.4}
port1{'VCC': 3.31, 'TEMP': 24.02, 'TX_POWER': 0.57, 'RX_POWER': 0.53, 'TX_BIAS': 7.3}
```



Simplified OOM Architecture

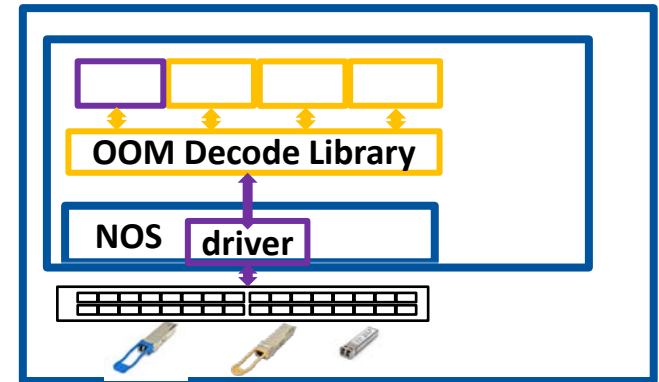


Open Source

Switch Vendor Specific

OOM Design – A Few Technical Details

- OOM Decode Library
 - Table driven, each key has a location and a decode function
 - Custom decode functions for special data formats
 - E.g., temperature is encoded in a 16-bit two's complement value, in units of 1/256°C
 - Also includes key collections, such as 'Serial ID' and 'DOM'
 - Table can be easily extended to enable access to additional capabilities
- 'optoe' Driver
 - Standard Linux kernel driver
 - Supports SFP family and QSFP family devices
 - Accesses transceiver EEPROM with Linux i2c APIs
 - Presents EEPROM as a file in /sys/bus/i2c/devices
 - Linux standard open(), read(), write() calls
 - Read/Write access to all 256 architected pages



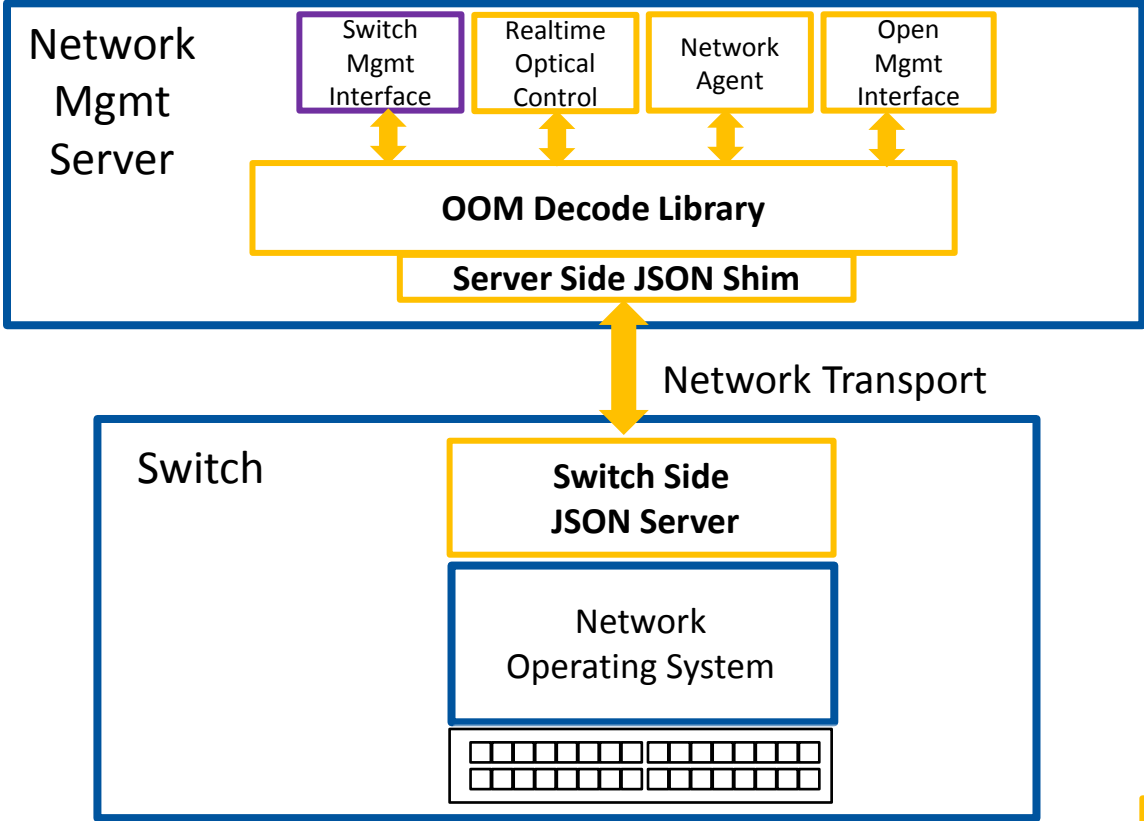
OOM Design – Logic Flow

- `oom_get_portlist()` – Get a list of all the ports
 - Scans `/sys/bus/i2c/devices` looking for SFP/QSFP ports
 - Returns a list of ports, for use by other OOM functions
- `oom_get_keyvalue(port, key)`
 - Lookup key in table, get location and decoder
 - 'port' provides access to the EEPROM file...
 - Read from the EEPROM, at the location specified in the table
 - Call the decoder with that data, return the result to the caller
- `oom_get_memory(port, collection)` – Get a named collection of key values
 - For each key in collection: `answer[key] = oom_get_keyvalue(port, key)`
 - Return answer to caller

```
from oom import *  
for port in oom_get_portlist():  
    status = oom_get_memory(port, 'DOM')  
    print port.port_name + str(status)
```

```
port0{'VCC': 3.30, 'TEMP': 23.55, 'TX_POWER':  
0.57, 'RX_POWER': 0.56, 'TX_BIAS': 7.4}  
port1{'VCC': 3.31, 'TEMP': 24.02, 'TX_POWER':  
0.57, 'RX_POWER': 0.53, 'TX_BIAS': 7.3}
```

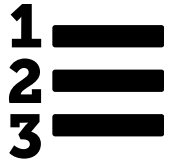
OOM Over The Network



Open Source

Switch Vendor Specific

Open Optical Monitoring (OOM) Applications



INVENTORY

track and verify module part number, revision, vendor, features, options...



HEALTH MONITORING

continuous monitoring of key properties (TX/RX power, laser bias, temperature, thresholds, alarms, warnings...)

Via DOM (Digital Optical Monitoring): a Finisar technology widely licensed to industry



DIAGNOSTICS

isolate connectivity issues to transmit module, receive module or fiber

Operator diagnostics using connection database

Vendor diagnostics while modules are online



CUSTOM USES

innovative new features

Customer or vendor-defined, vendor-specific keys

Example 1: BIST for network commissioning

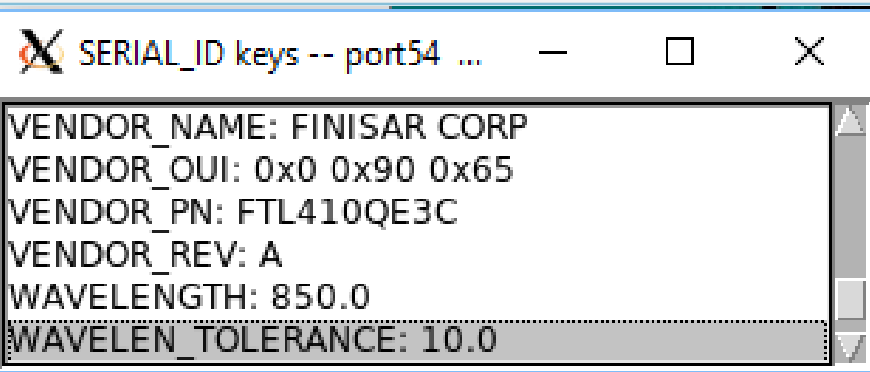
Example 2: Finisar Connectivity Diagnostics

Inventory Tracking with OOM

Record identifying information from all modules on the switch.

Confirm intended vs. actual type of modules present.

```
from oom import *  
for port in oom_get_portlist():  
    inventory = oom_get_memory(port, 'SERIAL_ID') # SERIAL_ID: 23 identifying keys  
    add_record(port, inventory) # add this module to the database  
    audit_record(port, inventory) # check for compliance
```



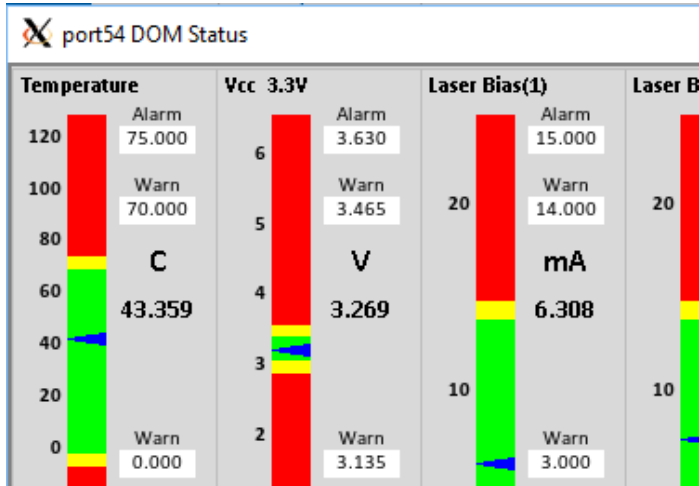
```
SERIAL_ID keys -- port54 ...  
VENDOR_NAME: FINISAR CORP  
VENDOR_OUI: 0x0 0x90 0x65  
VENDOR_PN: FTL410QE3C  
VENDOR_REV: A  
WAVELENGTH: 850.0  
WAVELEN_TOLERANCE: 10.0
```



Health Monitoring with OOM

Monitor and display module health metrics and alarms/warnings.

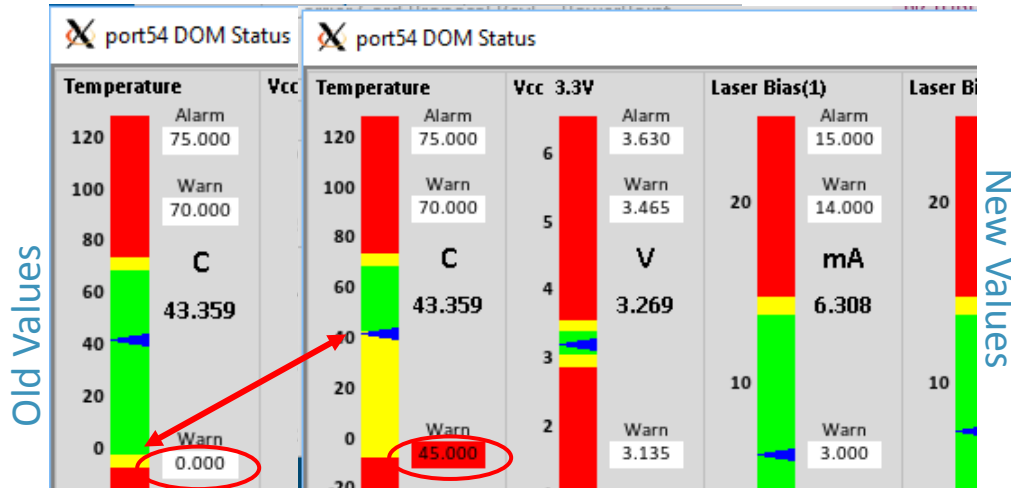
```
from oom import *  
list = oom_get_portlist():  
    health = oom_get_memory(list[53], 'DOM') # DOM: Digital Optical Monitoring  
    show_port(list[53], health) # Display temp, voltage, laser, Rx/Tx power
```



Diagnostics and Support with OOM

Vendor tech support adjusts e.g., low_temp warning threshold to test alert handling.

```
from oom import *  
list = oom_get_portlist(): # enumerate the ports on the switch  
    oom_set_keyvalue(list[53], 'PASSWORD_ENTRY', secret) # Vendor support password  
    oom_set_keyvalue(list[53], 'TEMP_HIGH_ALARM', 45.0) # Change threshold
```



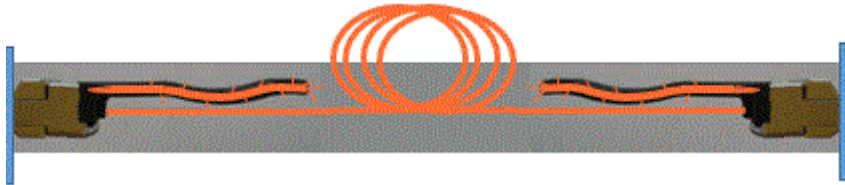
Custom Uses: Vendor Value-Added Content with OOM

Trigger flashing lights on module pull-tab with OOM.

Live in the switch, ***while running normal production workloads.***

Easy documented process to add additional keys to OOM.

```
from oom import *  
list = oom_get_portlist(): # enumerate the ports on the switch  
    oom_set_keyvalue(list[53], 'TAB_LIGHTS', flash) # Make the lights flash
```



OCP SUMMIT: Interoperability and Open APIs



X 2016 DataSheet 3 Module overview

Port Name	Module Type	Part Number	Serial Number	Manufacturer
2231-74004-5FP	PLX85110381	A4401C		FINISAR CORP.

Highlight one or more ports, then right click to select action

OOM – sponsored by: FINISAR

X 2016 DataSheet 3 Module overview

Temperature	Vcc 3.2V	Laser Bias	Ta Power	Ra Power
33.754 °C	2.958 V	8.072 mA	-2.38 dbm	-21.43 dbm

RX USE: 10760
 RX TAIL: 10880
 BK NOMINAL: 10200
 CABLE OPTIC: not det
 CONNECTIONS:
 DATE CODE: 100201
 DIAGNOSTIC MONITORING TYPE:
 S: SCHEDULE 6
 FINISARCF_3PTMWS: 743
 L31: L30N I31LN: 4
 I31PTMWS: 3
 LENGTH1 SIGNAL: 0D
 LENGTH1 HAZ SIGNAL: 3D
 LENGTH1 CNT: 100
 LENGTH1 ORN: 0N
 LENGTH1 SWP: 0
 LENGTH1 SWP CNT: 0
 OPTICS: 003 D0L0
 DATE DEFENSE: 0
 S11: 0472 COMPLIANCE: 3
 TRANSCIEVER: 0x10 not det not det not det not det
 TRANSCIEVER CNT: 0
 VENDOR NAME: FINISAR CORP.
 VENDOR OUI: 04b 8940 5065
 VENDOR PRT: 1-1020-1-1000A
 VENDOR REV: A
 VENDOR SN: 01061C
 WAVELENGTH: 830

OOM demonstrated at OCP Summits 2016 and 2017

How can you Access & Participate in OOM?

- OOM is now an **OCP Accepted™** Project
- Download, use and improve!
- <https://github.com/opencomputeproject/oom>
- <https://youtu.be/kkL2dk7zMOc>
- Share your use-cases with us.
- Used in Interoperability testing at UNH IOL Plugfests.
- Demonstrated in numerous Linux-based NOSs, white box switches, evaluation boards and a module simulator.
- 200+ keys decoded for QSFP+, QSFP28, SFP+...
some CFPx support available.



Latest OOM News

- February 2017 – ‘Universal Python Shim’.
 - No longer need to compile C code to install OOM.
 - Extensible to support any (every) Linux-based NOS.
- July 2017 – Introduced CFP family support in OOM.
- October 2017 – ‘optoe’ driver released for transceiver EEPROM.
 - For any Linux-based NOS.
 - Accesses more transceiver EEPROM capabilities than existing drivers.
- February 2018 – ‘optoe’ driver is in Open Network Linux (ONL), being used by 8 Accton/Edgecore switches and one Quanta switch. It is also in SONiC, being used by 2 Accton/Edgecore switches and one Inventec switch.

Thank You / Q&A

Acknowledgement: Don Bollinger, Finisar

Christian Urricariet
christian.urricariet@finisar.com