

Integrating the Calix FPGA OLT into R-CORD

Alan DiCicco
8 November 2017





Agenda

- **We've come a long way!**
- **FPGA OLT. What's the deal?**
- **What did Calix accomplish?**
- **Where are you headed?**

Celebrate accomplishments

ONF, the R-CORD Community, and Calix



Residential CORD

- Vision and architecture

Transformational

- Business, Network, Operations
- Agility of a cloud provider
- Efficiencies of a datacenter

OpenCORD Reference

- Template for production deployment



Benefits:

- Vendor Independence
- Reduce TCO
- Accelerate service delivery

Residential CORD – Disaggregated OLT (& ONU)



Orchestration

R-CORD Services

Controller: ONOS

Manager: vOLTHA

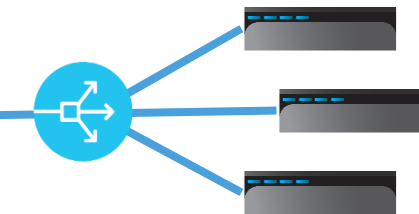
SDN-controlled
Leaf-Spine
Fabric

Shared Servers, Storage, Switches, I/O

- Residential-CORD triple play services
- Mix-and-match 10G PON OLT and ONU
- Virtualized ONOS-based control plane
- Virtualized PON manager (vOLTHA)
- Data-center inspired infrastructure



XGS-PON OLT



XGS-PON ONU

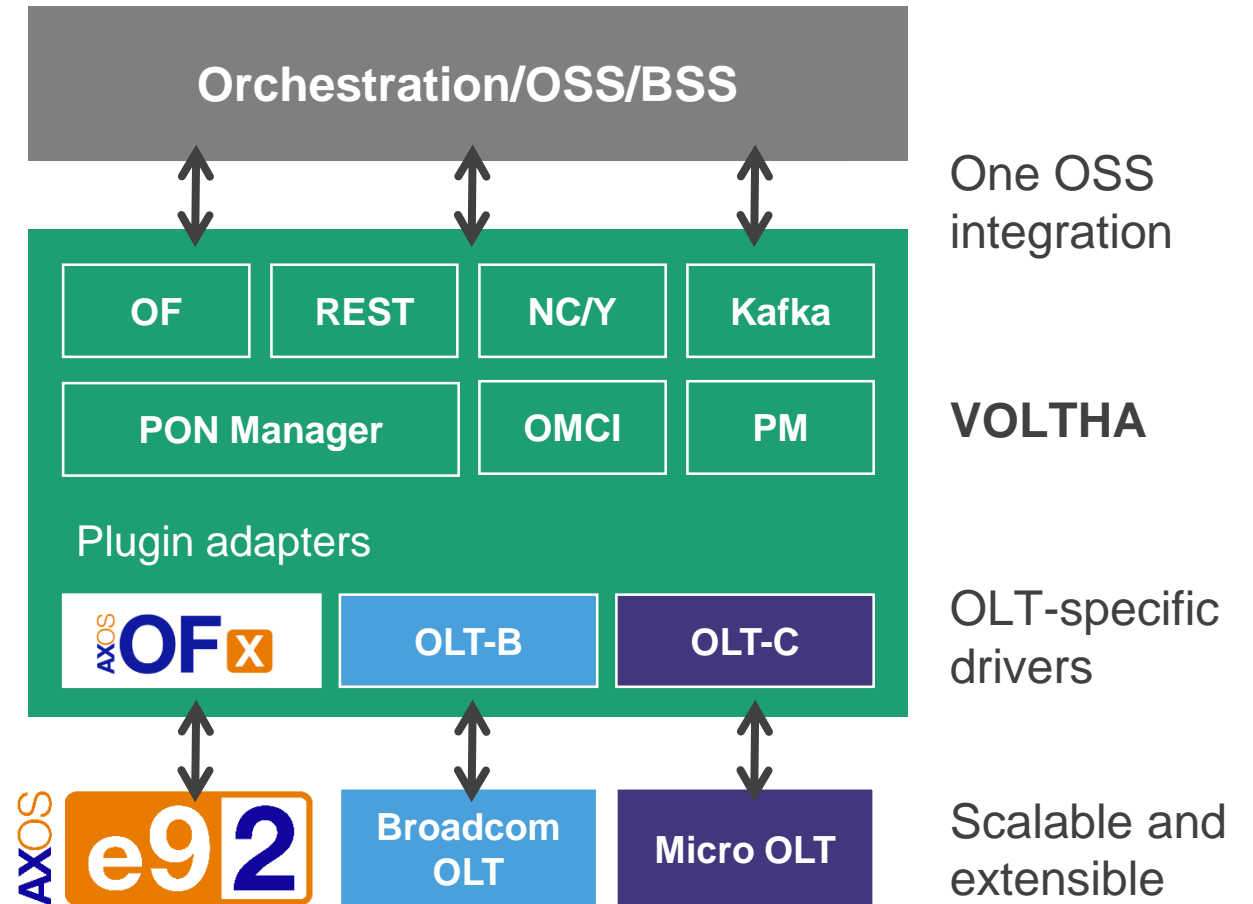
The Heart of CORD: Virtual OLT Hardware Abstraction (VOLTHA)

Isolation, translation, scale

- Harmonized & centralized management

Toward production quality

- Standardized northbound interfaces
- Status and performance monitoring
- OLT / ONT software management ✓
- ONT auto registration ✓
- VOLTHA high availability
- Orchestration integration



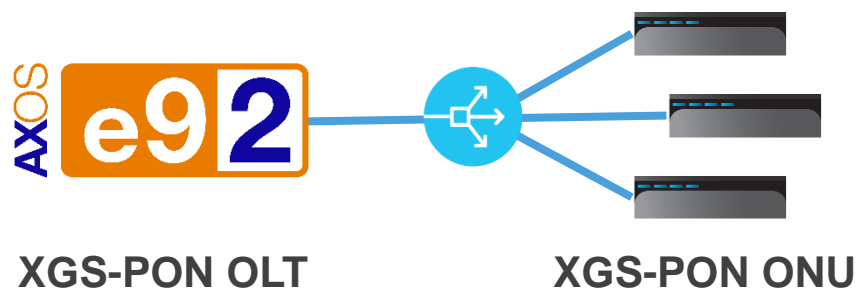
ONOS tenant application interoperability

NB APIs	■	■	■
Control and Data Plane Functions	DHCP	802.1x	IGMP
Service Abstraction	■		
SB APIs	■	■	■



- Operating system for network tenant apps
- Full interoperability with R-CORD applications

- ✓ IGMP Proxy
- ✓ DHCP Relay
- ✓ 802.1x Network Access Control



Global SDN/NFV ecosystems extend Software Defined Access



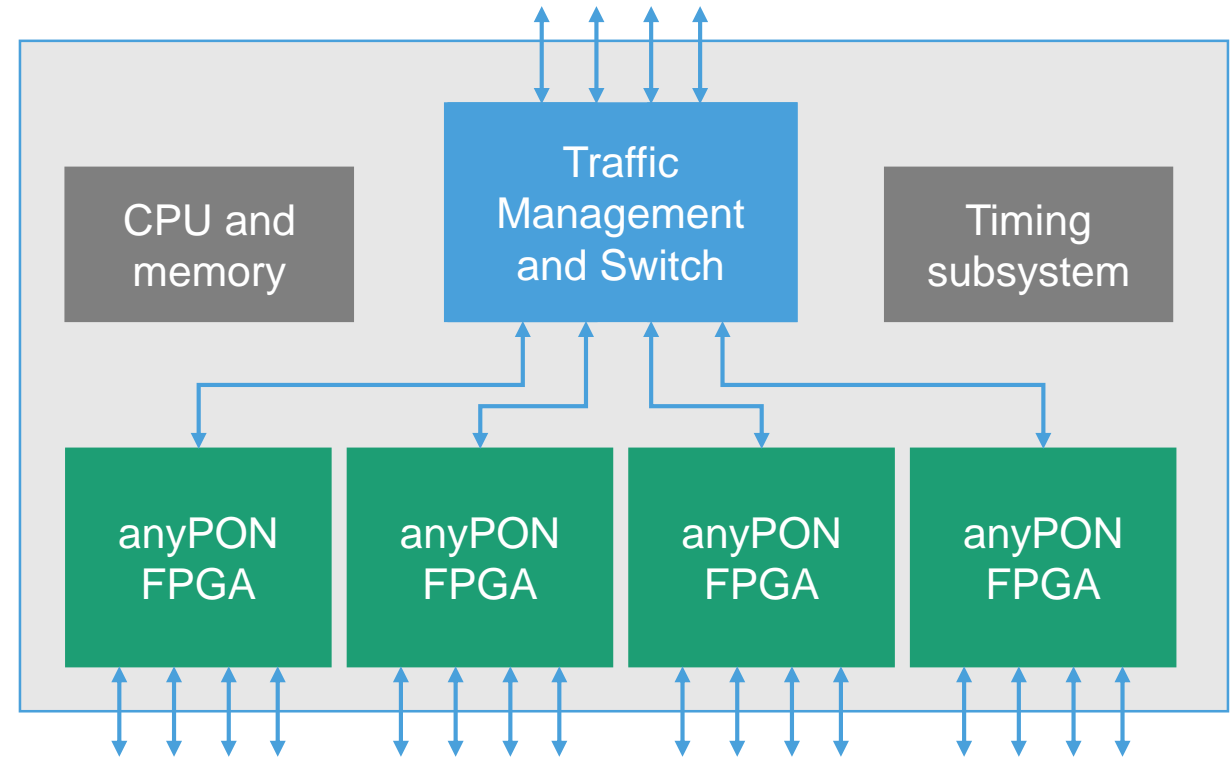
FPGA-based OLTs

What and why?



What is an FPGA-based OLT? What makes it different?

- High performance four-port **PON FPGA** supporting NG-PON2, XGS-PON, GPON, 10G EPON, etc.
- Non-blocking **Traffic Management and Switch**
- **CPU and memory**
- **Timing subsystem**
- All that other stuff... LEDs, baseboard management controllers (BMC), craft port, etc.



Q: What's the difference?

A: Merchant ASIC / SDK vs. merchant FPGA / firmware

More info: http://www.opencompute.org/wiki/Telcos#Specs_and_Designs

Why does Calix use and FPGA instead of merchant silicon?

1. We've been building OLTs for nearly 20 years
2. Flexibility... we literally support "any PON"
3. Agility... new functions independent of silicon vendor
4. Packaging, energy efficient, globally compliant options...



In the
Central Office



In a Cabinet



Strand, Pole,
Vault or MDU



What did Calix accomplish?

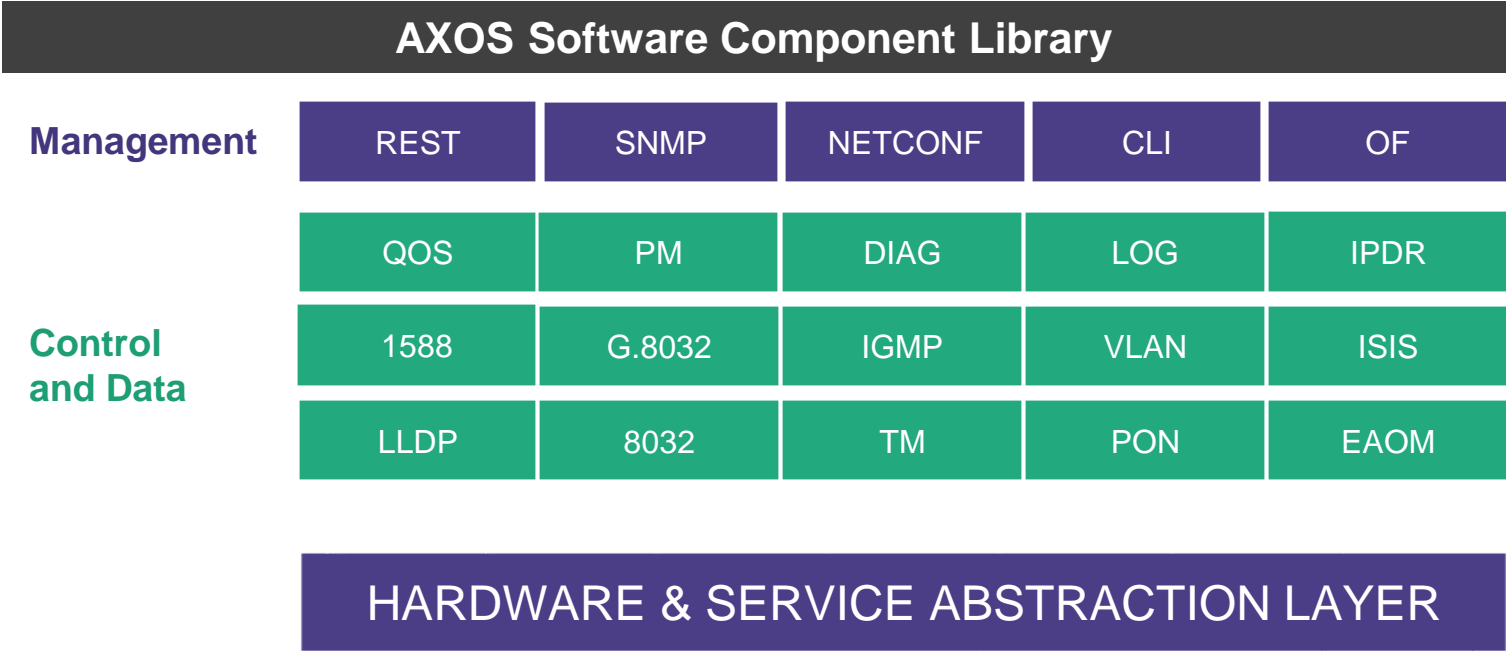
Huh?



Modular software independently developed, tested, and deployed



- Independent modular software components in a layered architecture
- Incremental development without impact to system
- Full YANG FCAPS modelling drives API integration



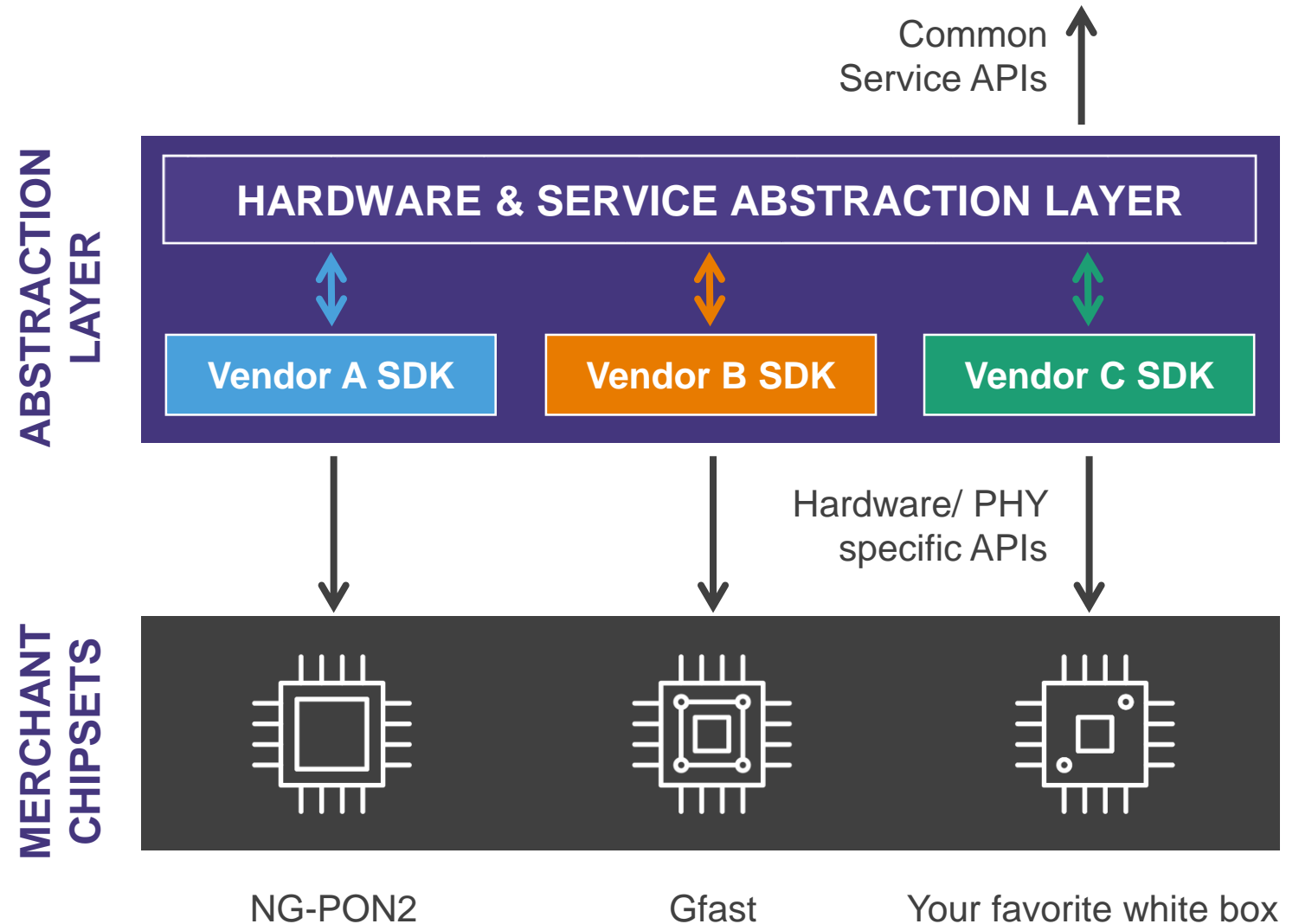
YANG for EVERYTHING

- Services
- Interfaces
- IP functions
- Bridges
- AAA
- PON, ONUs
- User Security
- Upgrade, Backup, Restore, Health
- Alarm Management
- Event Management
- Performance (Export/Poll)
- Troubleshooting & Diagnostics
- Zero Touch and Call Home
- Equipment and Inventory

Separate software from hardware – applying SDN principles at the silicon level



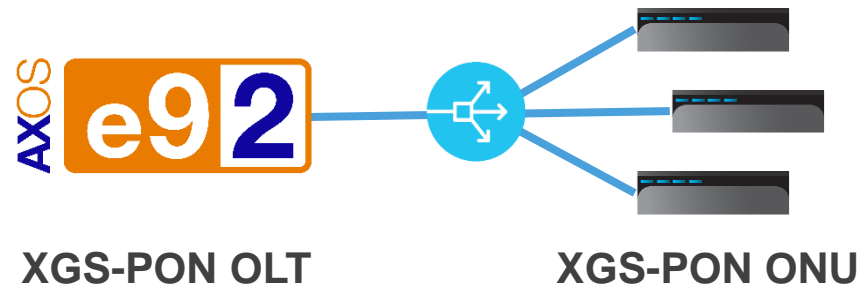
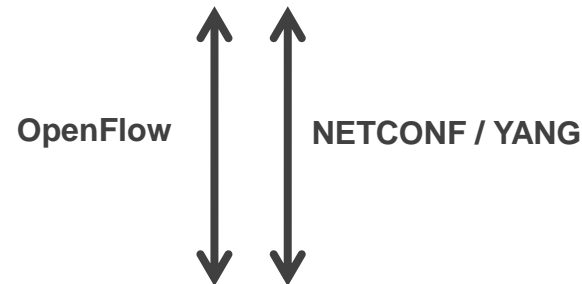
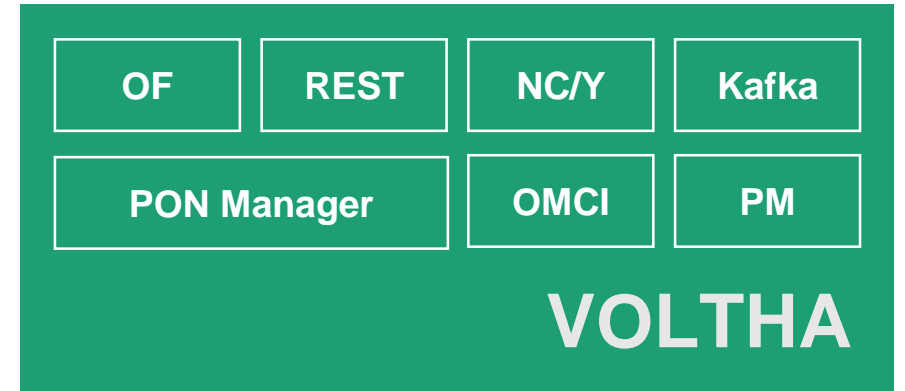
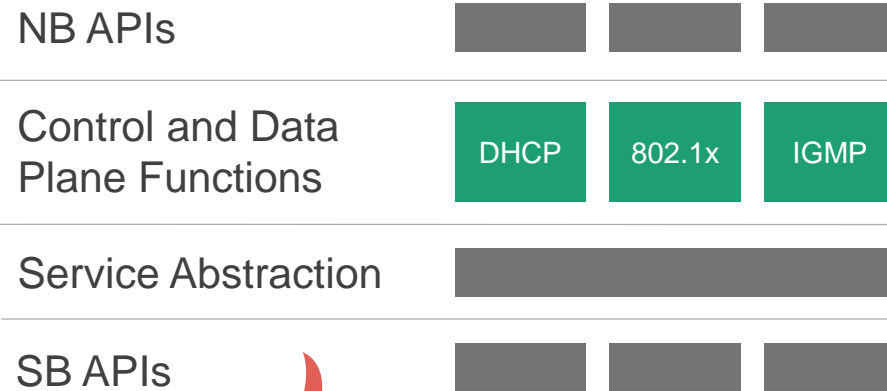
- Decouple software from hardware
- Achieve consistent service models and workflows across all technologies
- Deliver tomorrow's new technology faster



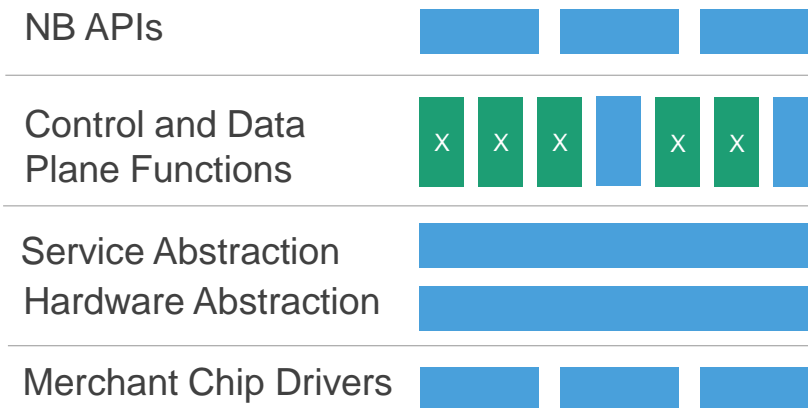
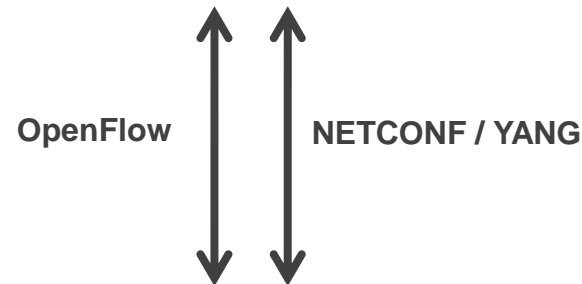
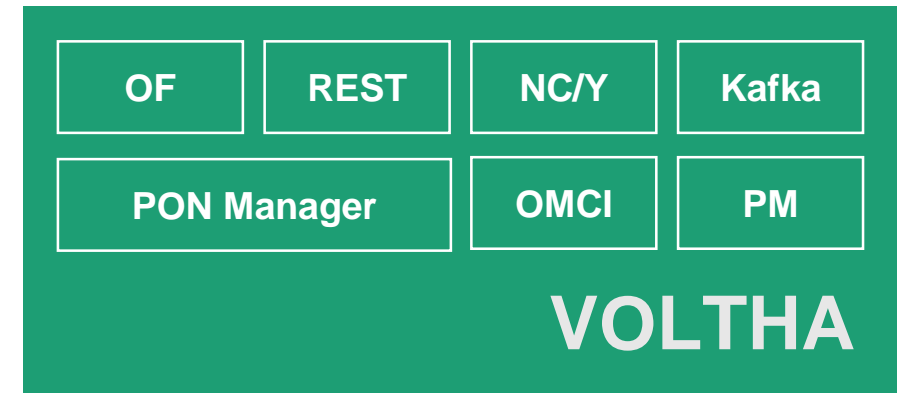
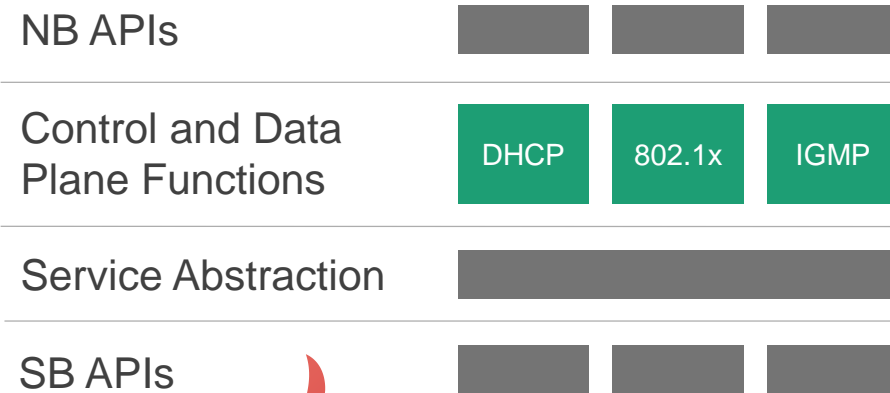
OpenCORD Integration – a software and hardware packaging exercise



OpenCORD Integration – a software and hardware packaging exercise



OpenCORD Integration – a software and hardware packaging exercise



Moving forward

Choose the ROI-maximizing path



SDA Takes Many Forms... Maximize your investment

AXOS. Why are we different?

- Integrated into OpenCORD
- Migrate you to OpenCORD
- Ready to extend OpenCORD

Software Defined Access

- The path you take is driven by your business
- Deploy, transform, extend



Infinite Possibilities

Thank you

