

OCP U.S. SUMMIT 2016

March 9-10 | San Jose, CA

OCP U.S. SUMMIT 2016

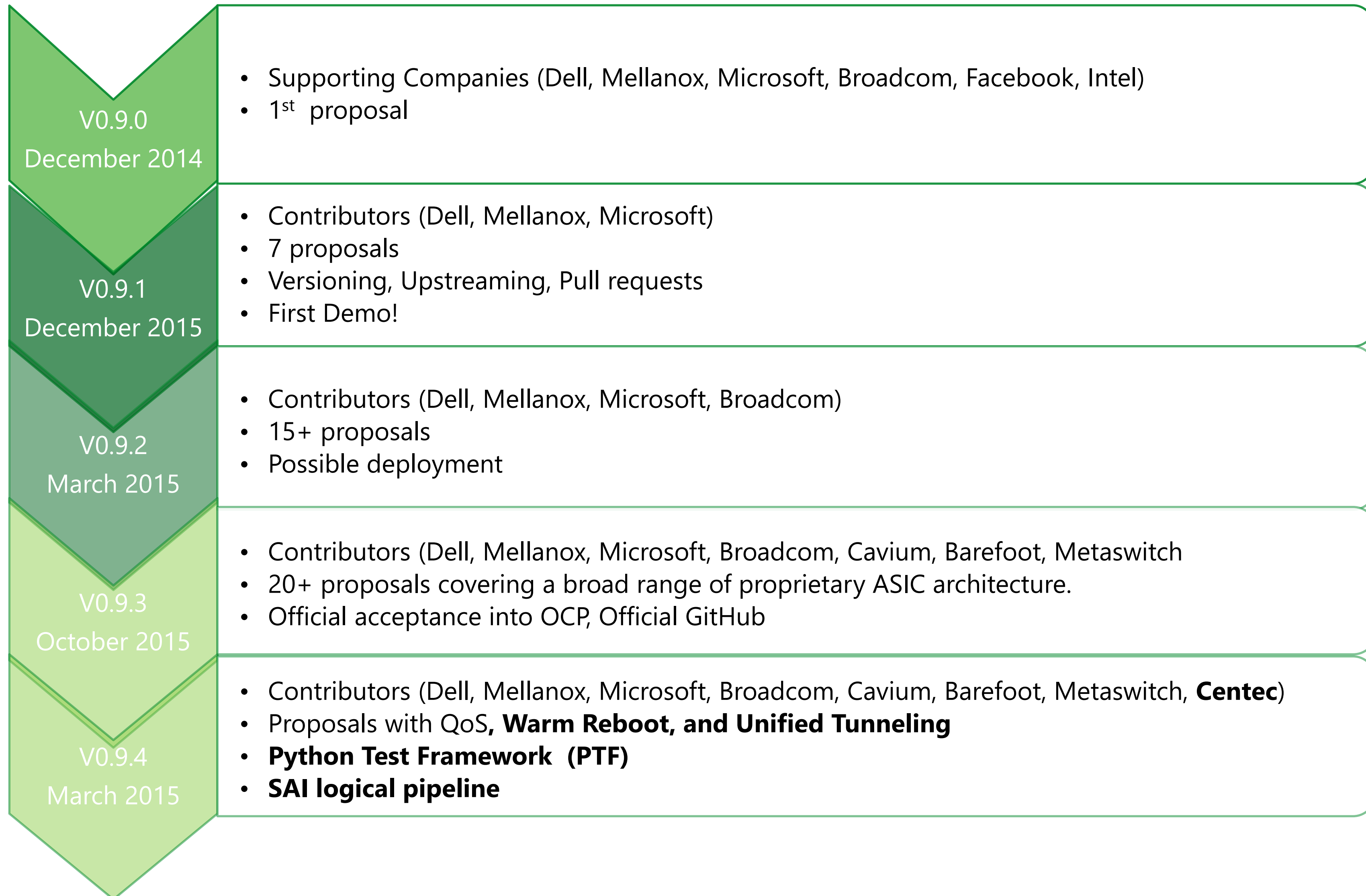
Switch Abstraction Interface (SAI)

OCP U.S. SUMMIT 2016

SAI Contributors



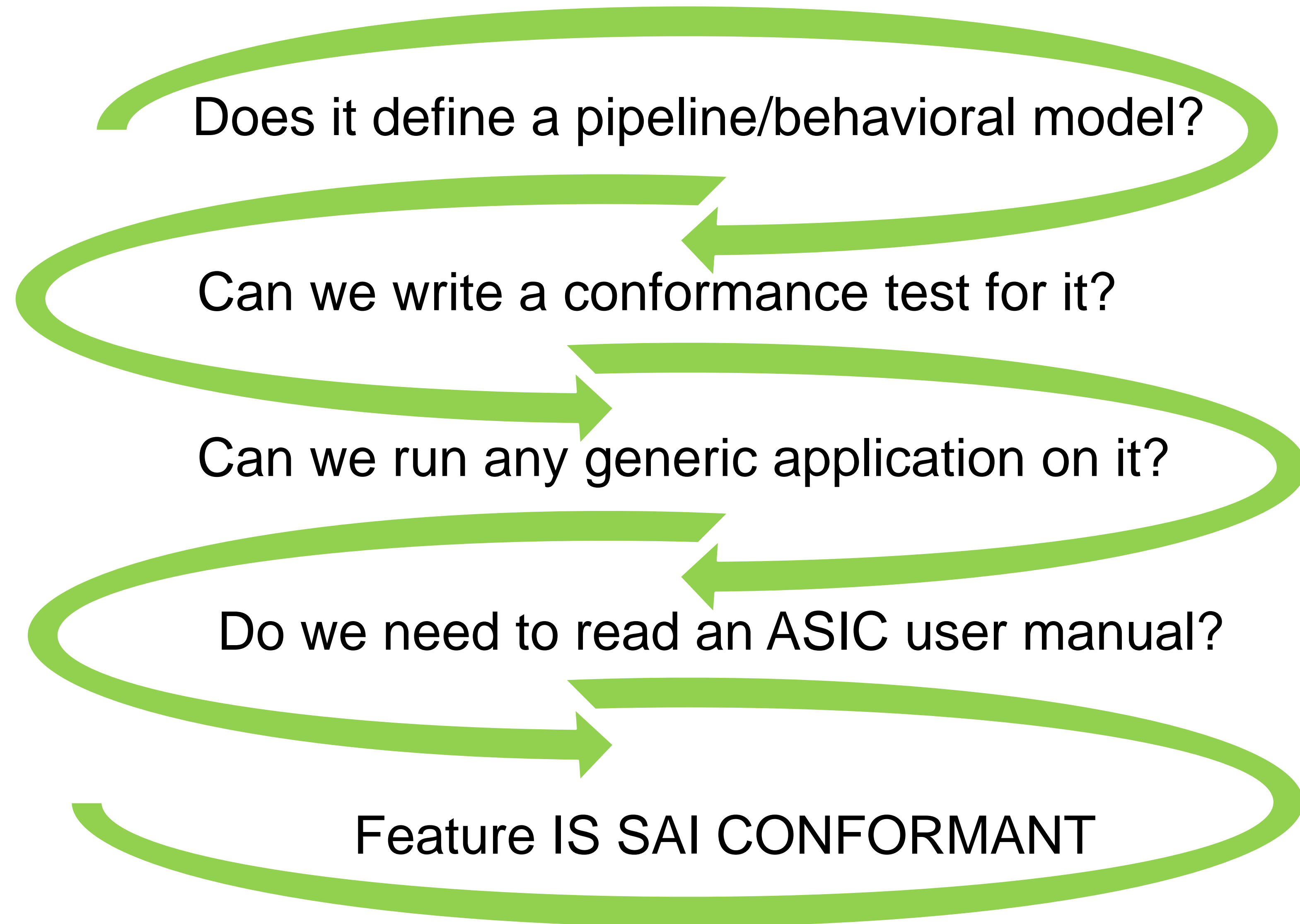
Momentum, Timeline



Momentum, Demos & Deployment



Technical Merit, Architectural philosophy



SAI keeps momentum

- Increasing in number of contributors
- Increasing in number of proposals
- ~ 3 releases a year
- Heading towards testing compliancy
- Working on a logical pipeline

OCP U.S. SUMMIT 2016

Software for Open Networking in the Cloud (SONiC)

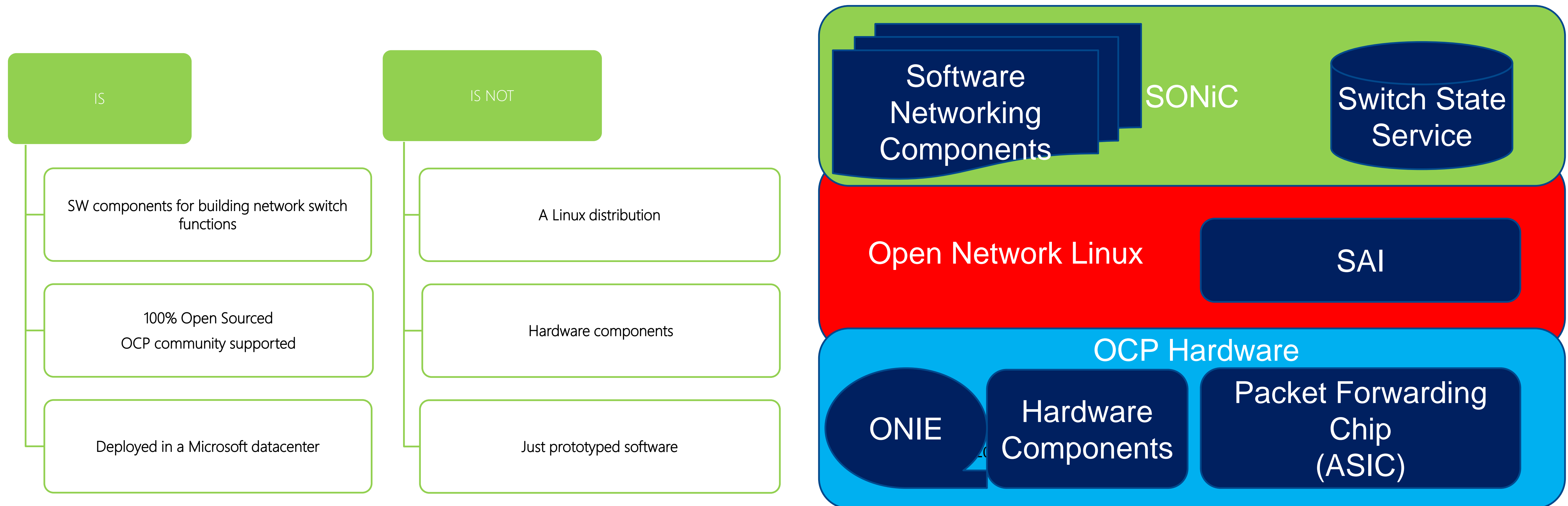
OCP U.S. SUMMIT 2016

SONiC Contributors

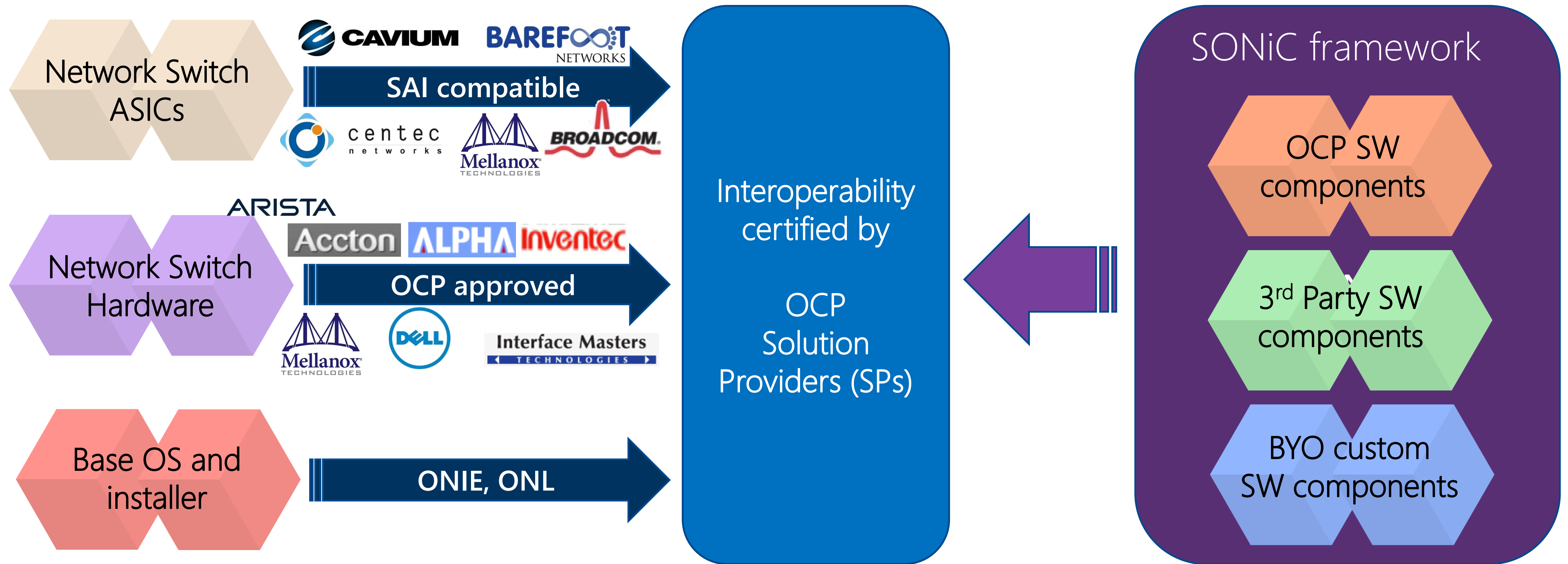
ARISTA



SONiC and the OCP Stack



OCP Ecosystem Enhanced with SONiC



Fully Open Sourced switching platform - Increased choices for OCP end users

What Is SONiC

A collection of software components/tools

- Builds on the foundations of SAI
- Provides L2/L3 functionalities
- Loosely-coupled modular design
- Separation of states and logic

Community driven, open source effort

- Shared on GitHub, Apache License
- Believe in working code + quick iteration

What can SONiC enable?

Building complete and production-ready stack

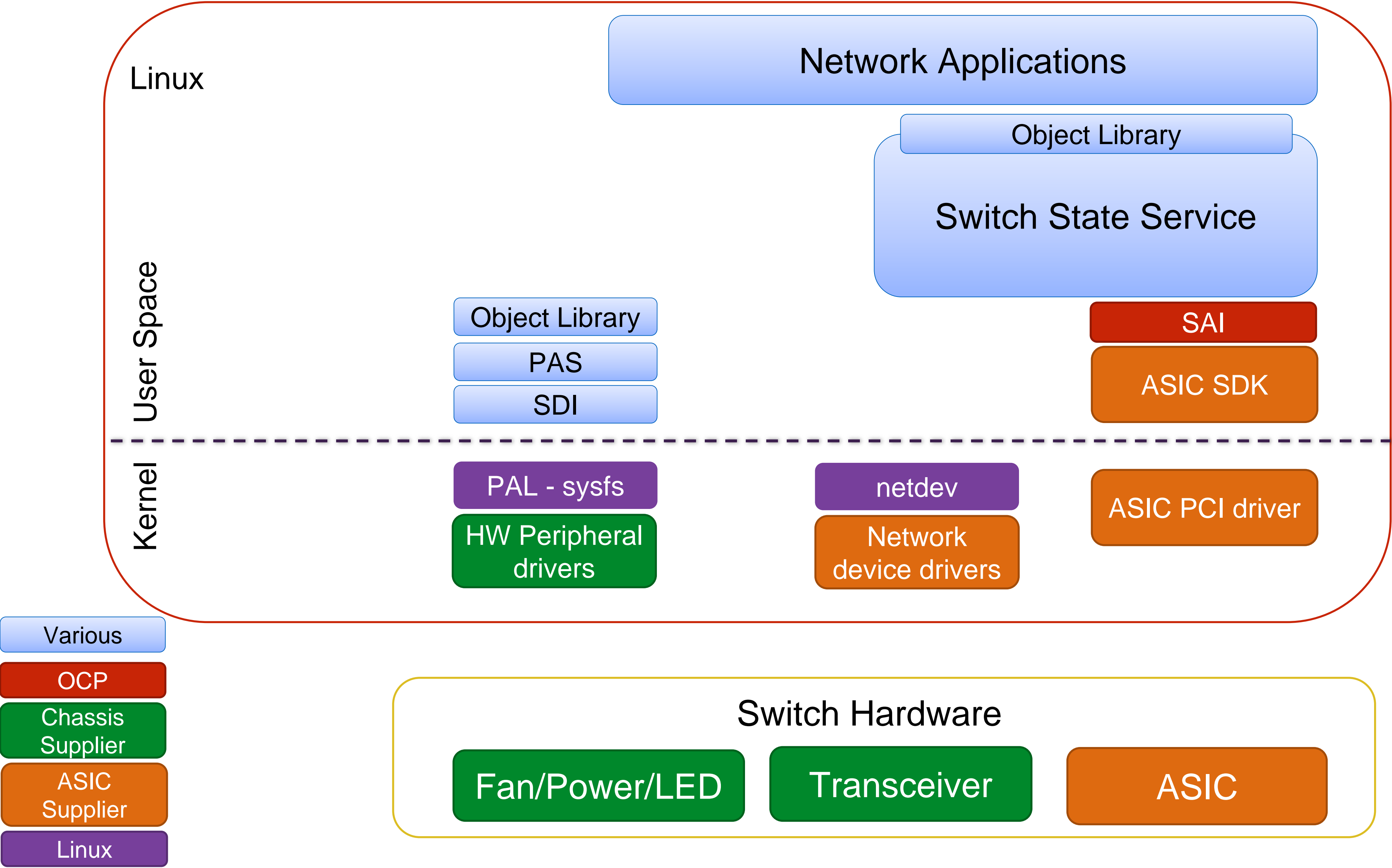
Easy portability

- ASICs (through SAI)
- Platform (Arista 7050QX, Dell S6000, Mellanox Spectrum, ongoing with ONL)
- Base Linux Distribution (Debian/Ubuntu/ONL)

Fast evolution

- for both prototype and production

SONiC High Level Architecture



Switch State Service (SSS)

SAI DB: persist SAI objects

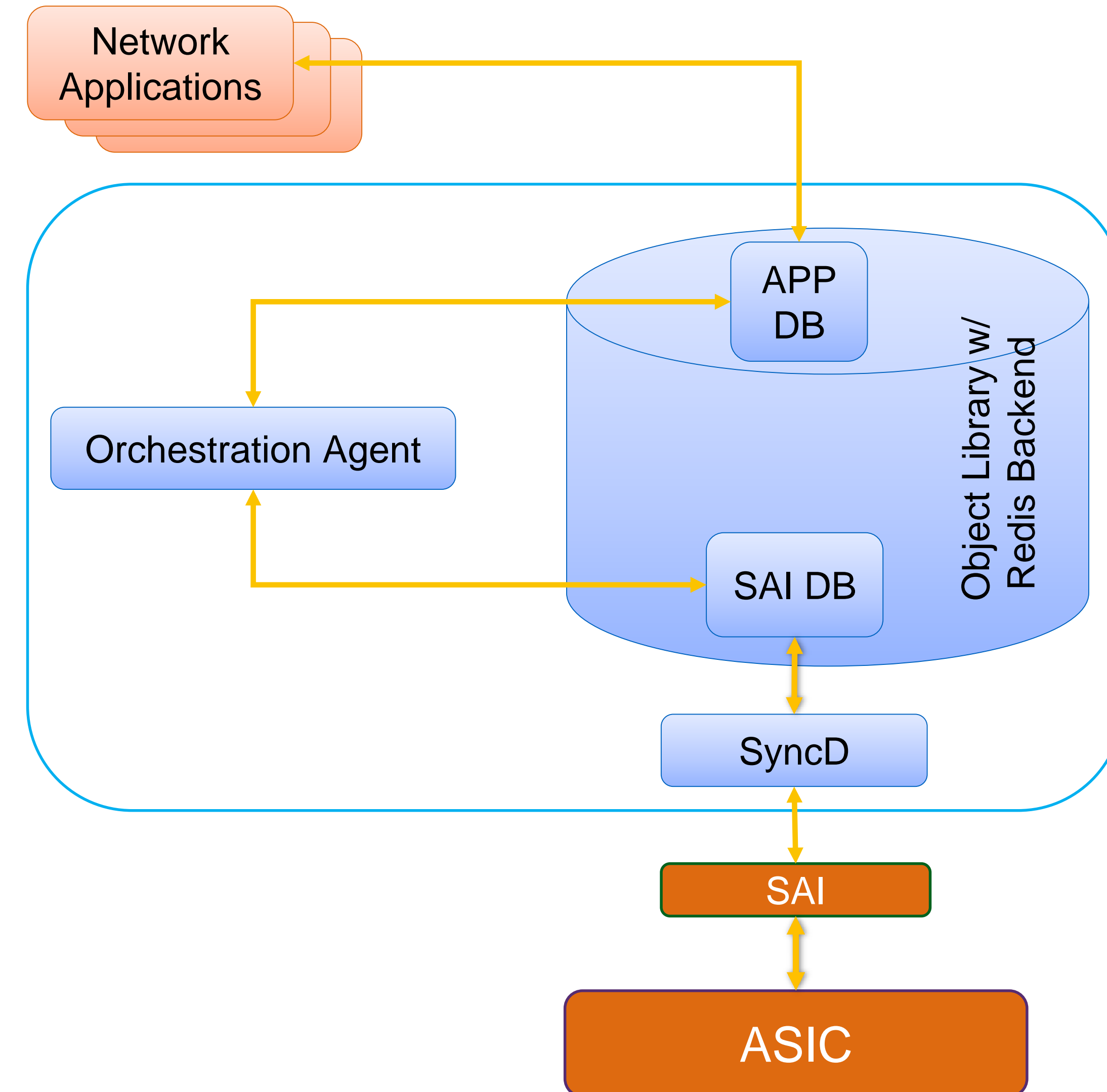
App DB: persist App objects

DB backend: redis with object library

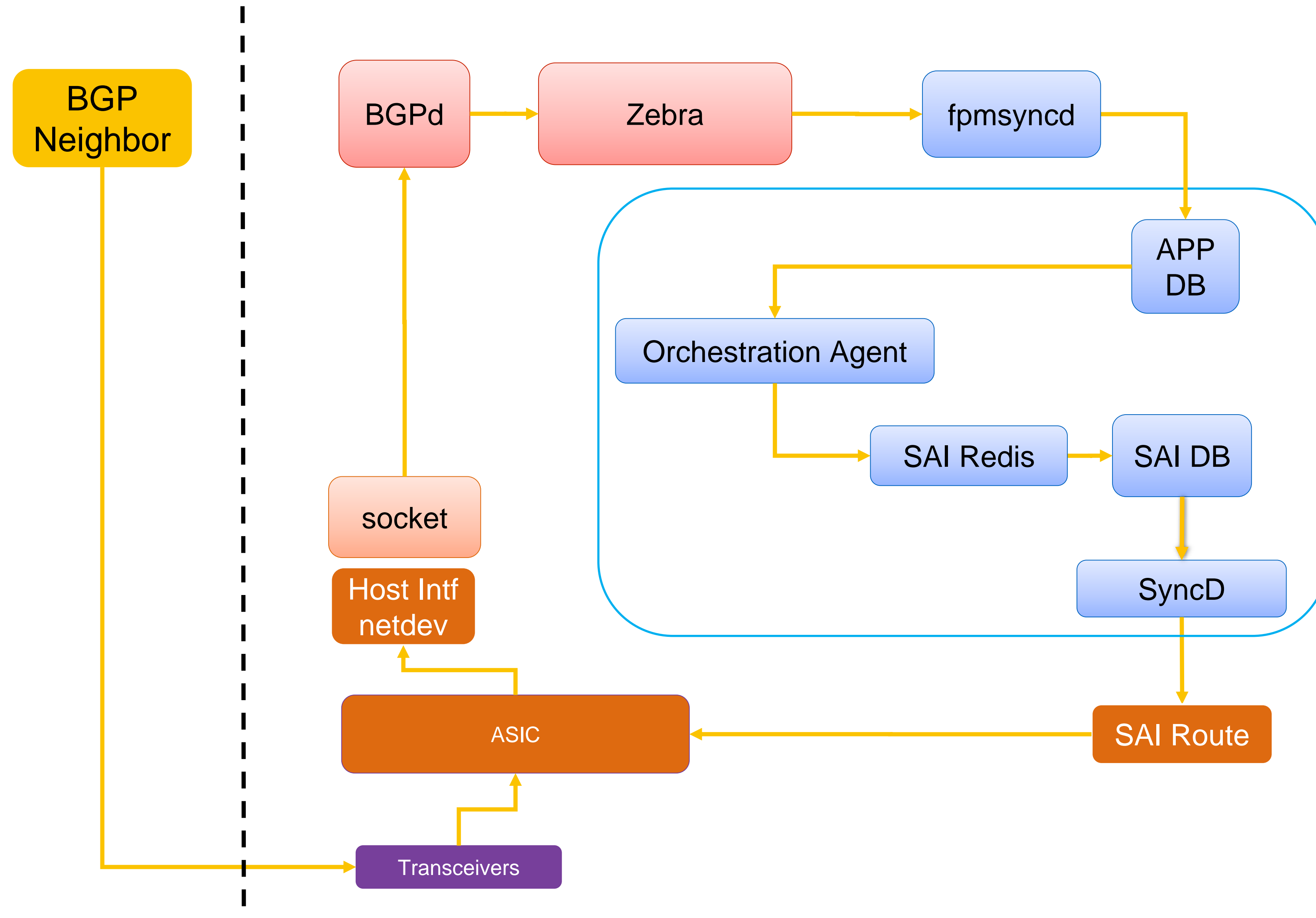
SyncD: sync SAI objects between software and hardware

Orchestration Agent: translation between apps and SAI objects, resolution of dependency and conflict

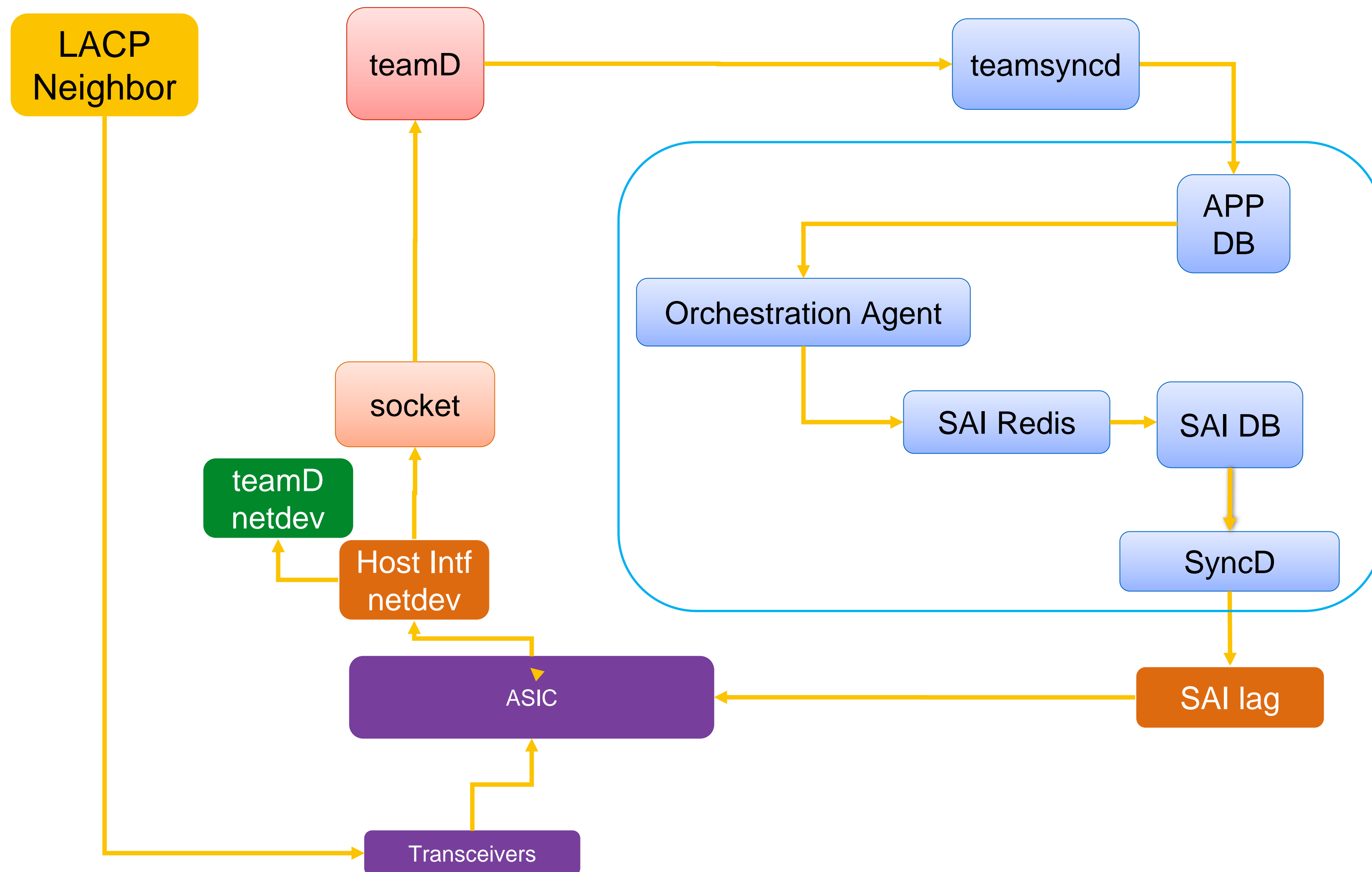
Key Goal: Evolve components independently



How Routing Works in SONiC

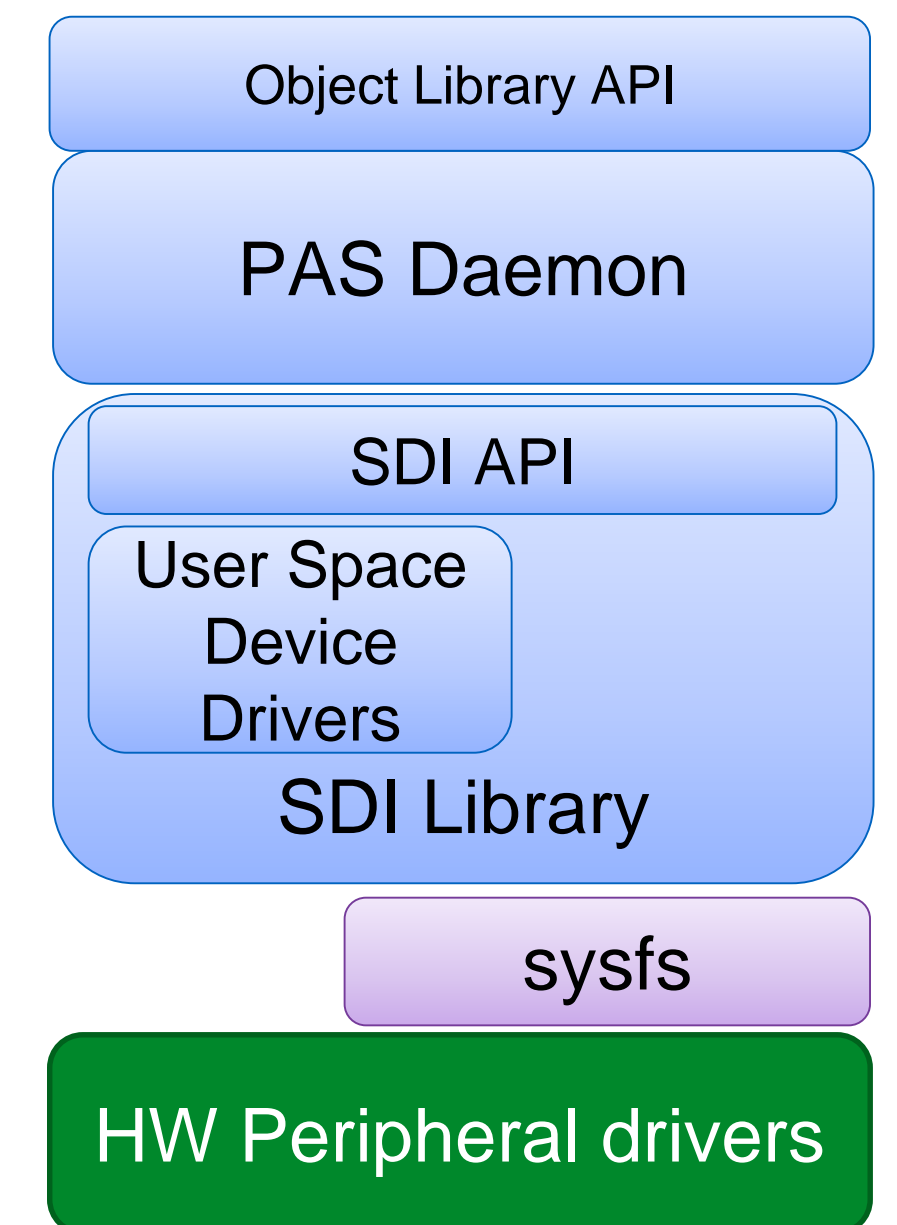


How LAG Works in SONiC



SONiC Platform Support

- PAS is implemented as daemon (PAS daemon)
 - monitor system device health and raises failure events
 - detects insertion/removal events (fan, PSU's, transceivers)
 - detects over-temperature events
 - allows applications to control system devices (get/set)
- SDI encapsulates and aggregates system devices
 - For instance, the SDI API defines a “fan device entity” with attributes such as:
 - ***fan unit presence***: implemented in CPLD and accessed through “sysfs”
 - ***fan speed***: fan controller driver used to set/get fan speed (through I2C ioctl calls or sysfs calls, depending on fan controller device implementation)
 - Implementation details of how “fan device entity” attributes are accessed is encapsulated by SDI API
 - SDI library implementation can use either sysfs/kernel drivers or user space drivers

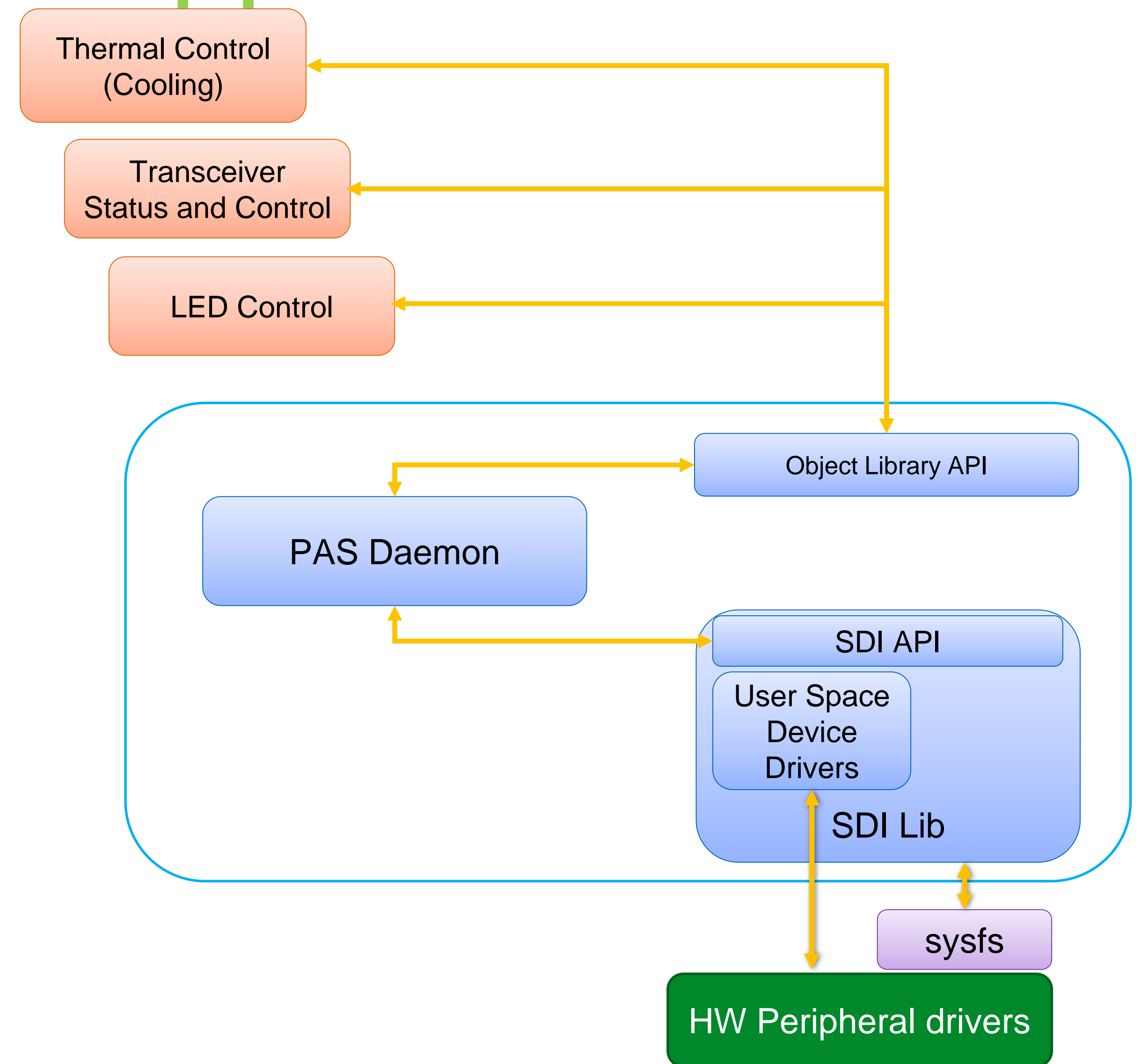


SONiC Platform Applications

■ Platform Applications

- use the PAS Object Library API to control (set/get) system devices
- register for events published by PAS daemon
 - Device Insertion / removal
 - Other events (e.g. over-temperature events, device failures)

For instance, a thermal control (cooling) application can register for “over-temperature events and increase (set) fan speed values accordingly.



Demo: SONiC Walk Through



The diagram illustrates the components of a SONiC system. It features a central vertical column of five green rounded rectangular boxes, each containing a component name. To the left of each box is a short vertical line, and to the right is a long horizontal white box with a green border. These horizontal boxes are connected to the central green boxes by short vertical lines, suggesting a flow or relationship between the components and the data they manage.

EEPROM

Port Status

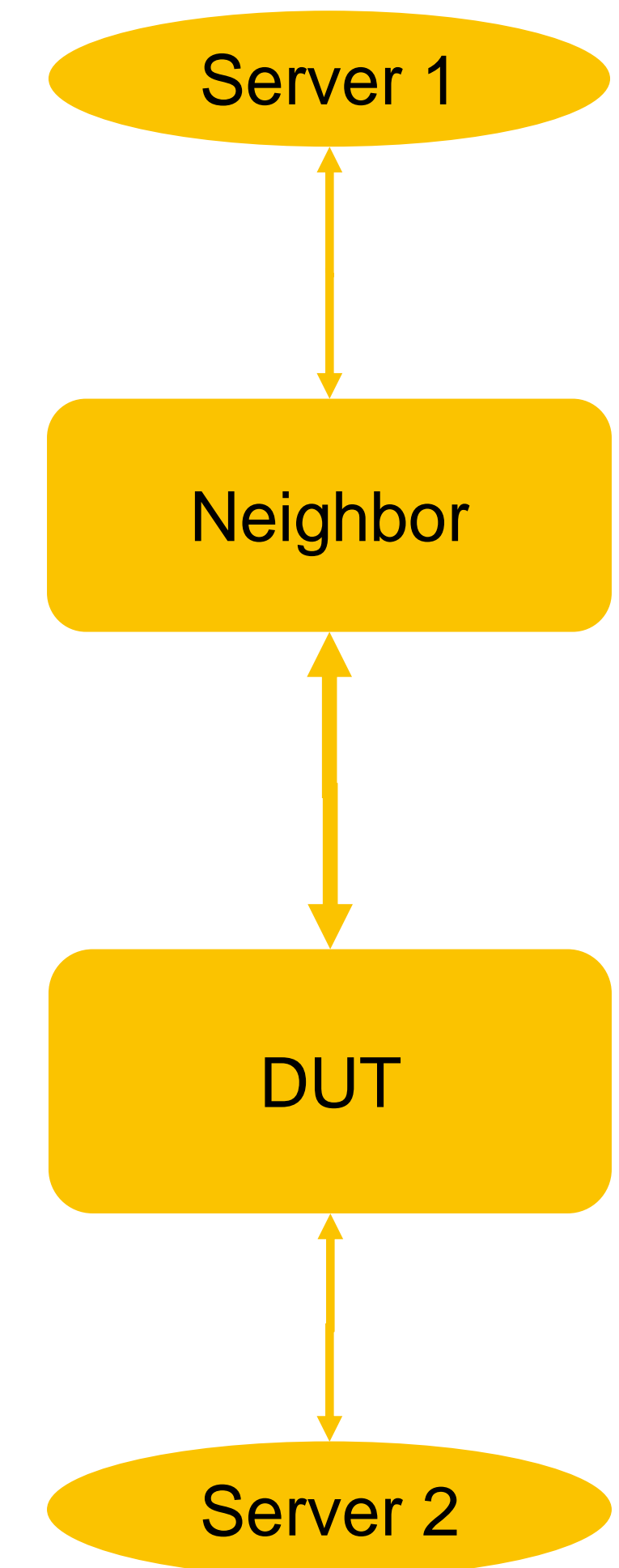
TCPDump

Redis

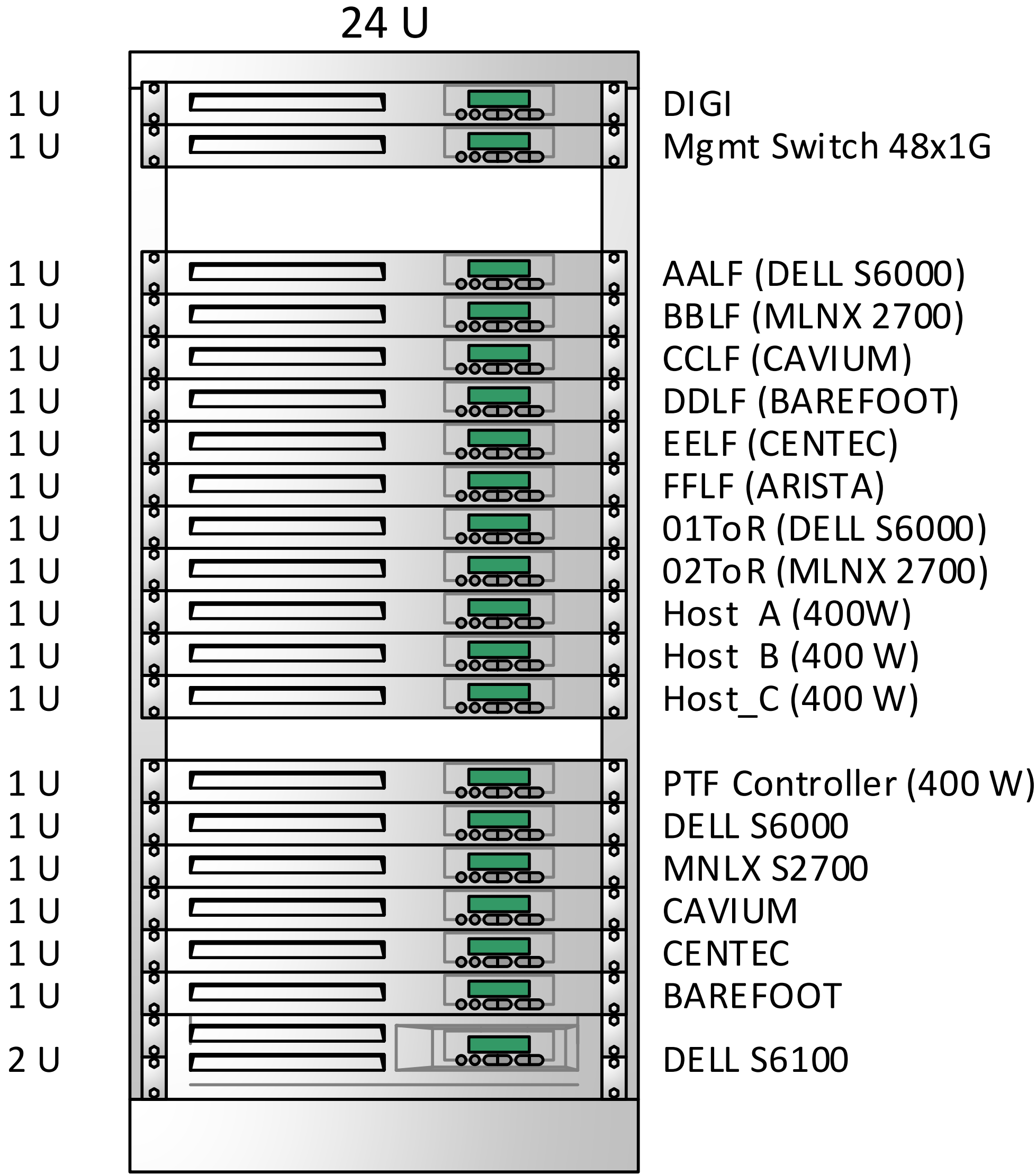
Quagga and FIB

Demo: Hitless Quagga to GoBGP Migration

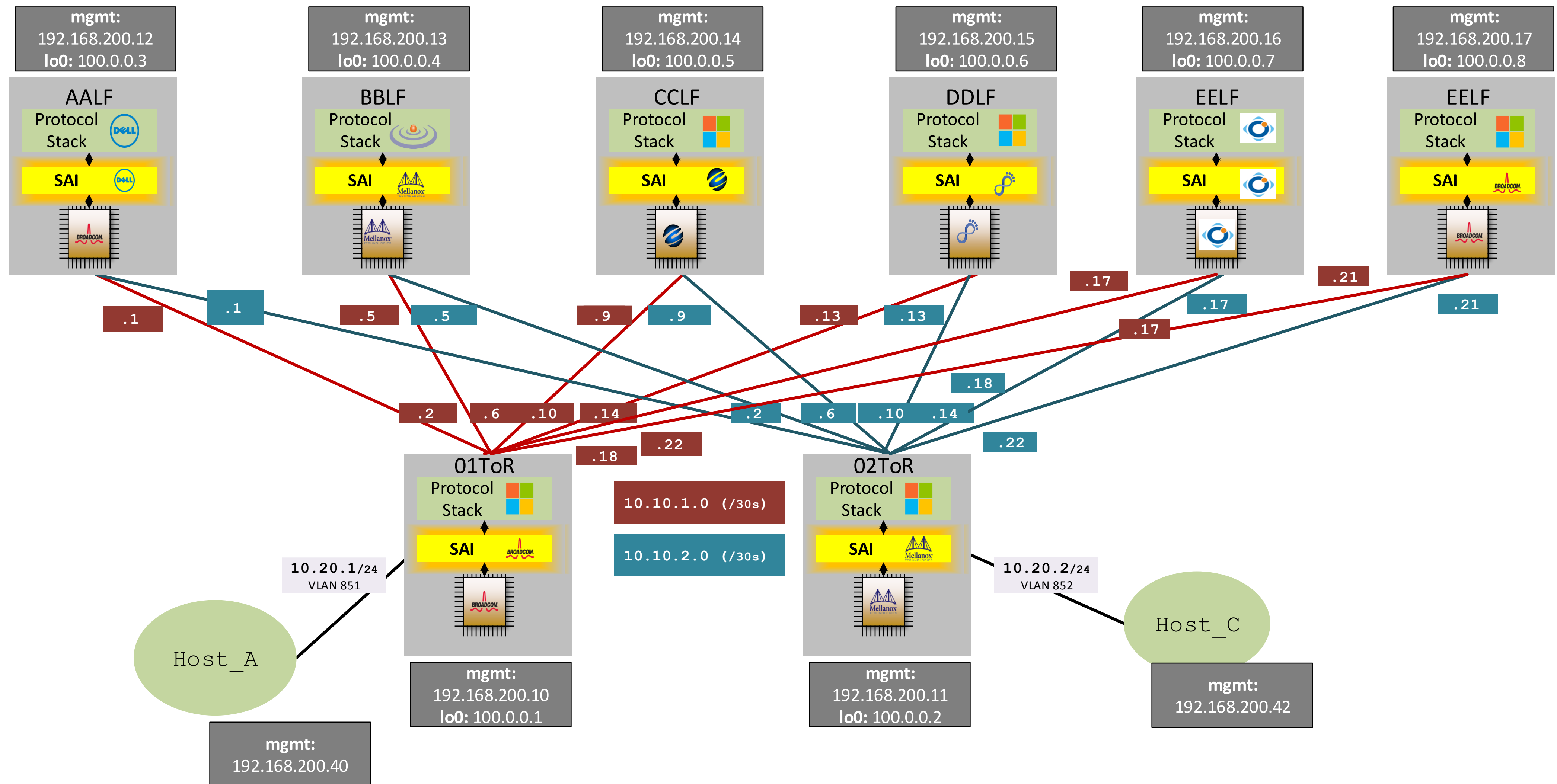
- What is the real scenario?
 - Hot patching
 - Rollout new features, e.g. BMP
- How is this achieved?
 1. Neighbor supports graceful restart
 2. DUT request OA to freeze FIB
 3. DUT uninstalls Quagga
 4. DUT installs GoBGP
 5. DUT wait for route convergence
 6. DUT request OA to unfreeze FIB



Demo at the Microsoft Booth

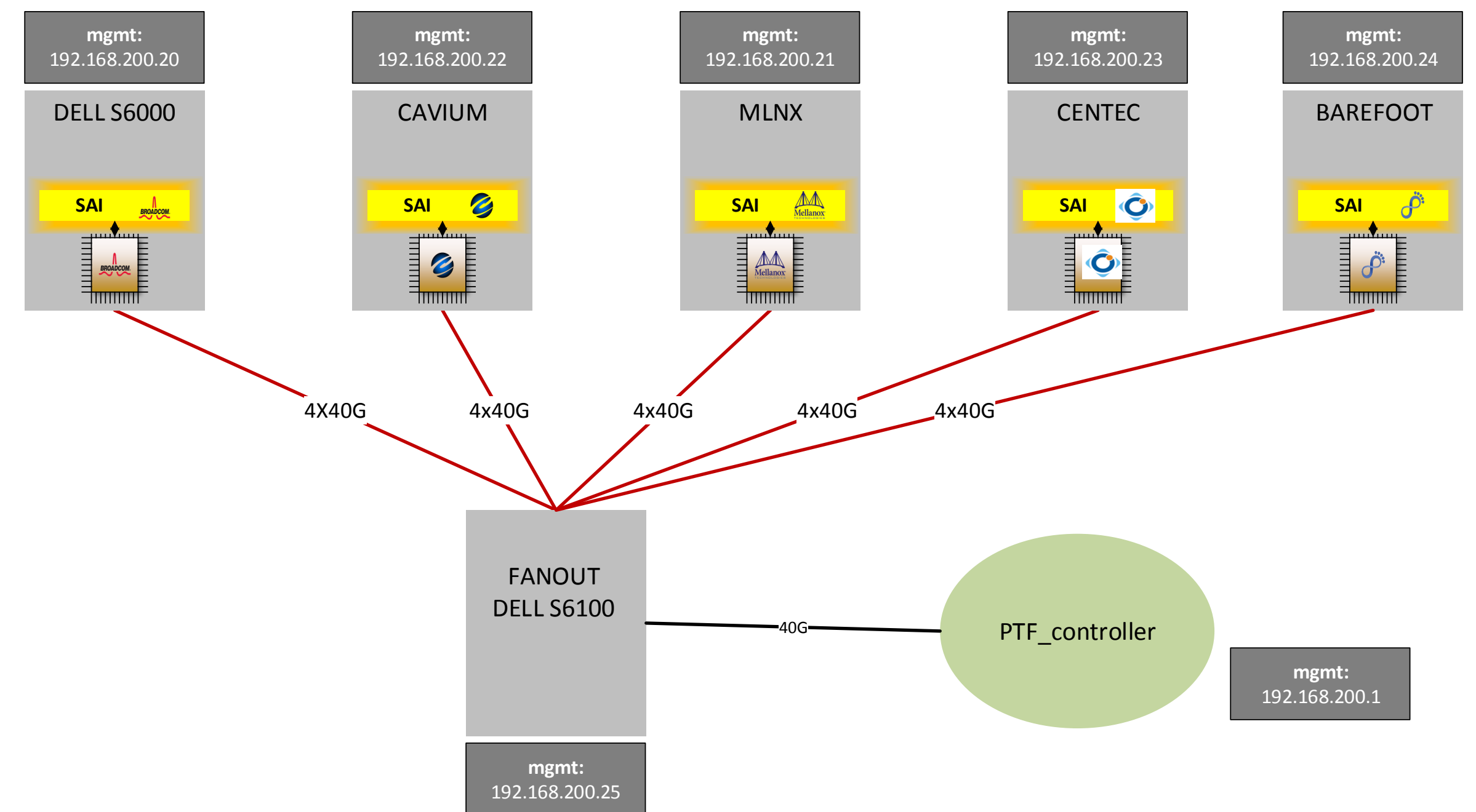


Feature Functionality Demo



Python Test Framework Demo

- Python based test framework
- 20+ test cases
- Working towards compliancy
- Looking for community participation
- Future proposals accepted only with PTF test cases



Conclusion

We are proposing SONiC as a contribution to the OCP

- SONiC

- Fully open sourced
- Rich feature functionality cloud switch
- Community effort
- Currently deployed

- SONiC at OCP

- Runs on top of SAI
- Will integrate with ONL
- Will run on OCP switches

SONiC Governance Model

- Very permissive read, restricted write
- Roles
 - Contributors, many to a repo
 - Maintainers, one for a repo
 - Project Leader, one for all repos
- Maintainer choice based on contributing company
- Technical Meritocracy Model