

0CP U.S. SUMMIT 2016 March 9-10 | San Jose, CA





## SAI Contributors















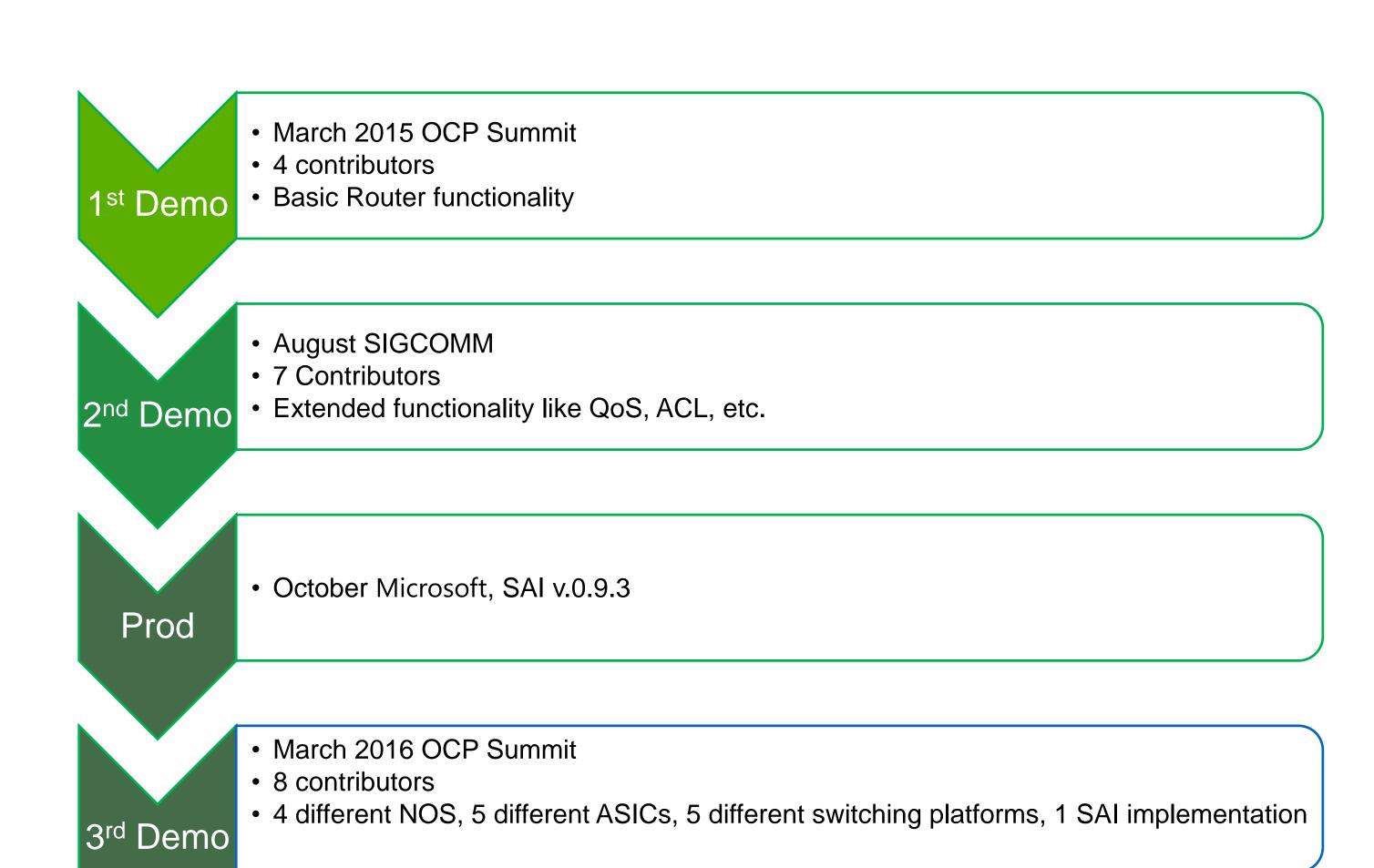


#### Momentum, Timeline

Supporting Companies (Dell, Mellanox, Microsoft, Broadcom, Facebook, Intel) 1st proposal V0.9.0 December 2014 Contributors (Dell, Mellanox, Microsoft) • 7 proposals Versioning, Upstreaming, Pull requests V0.9.1 • First Demo! December 2015 Contributors (Dell, Mellanox, Microsoft, Broadcom) • 15+ proposals V0.9.2 Possible deployment March 2015 Contributors (Dell, Mellanox, Microsoft, Broadcom, Cavium, Barefoot, Metaswitch • 20+ proposals covering a broad range of proprietary ASIC architecture. V0.9.3 Official acceptance into OCP, Official GitHub Contributors (Dell, Mellanox, Microsoft, Broadcom, Cavium, Barefoot, Metaswitch, **Centec**) Proposals with QoS, Warm Reboot, and Unified Tunneling Packet Test Framework (PTF)

SAI logical pipeline

#### Momentum, Demos & Deployment



### Technical Merit, Architectural philosophy

Does it define a pipeline/behavioral model?

Can we write a conformance test for it?

Can we run any generic application on it?

Do we need to read an ASIC user manual?

Feature IS SAI CONFORMANT

### SAI keeps momentum

Increasing in number of contributors

Increasing in number of proposals

~ 3 releases a year

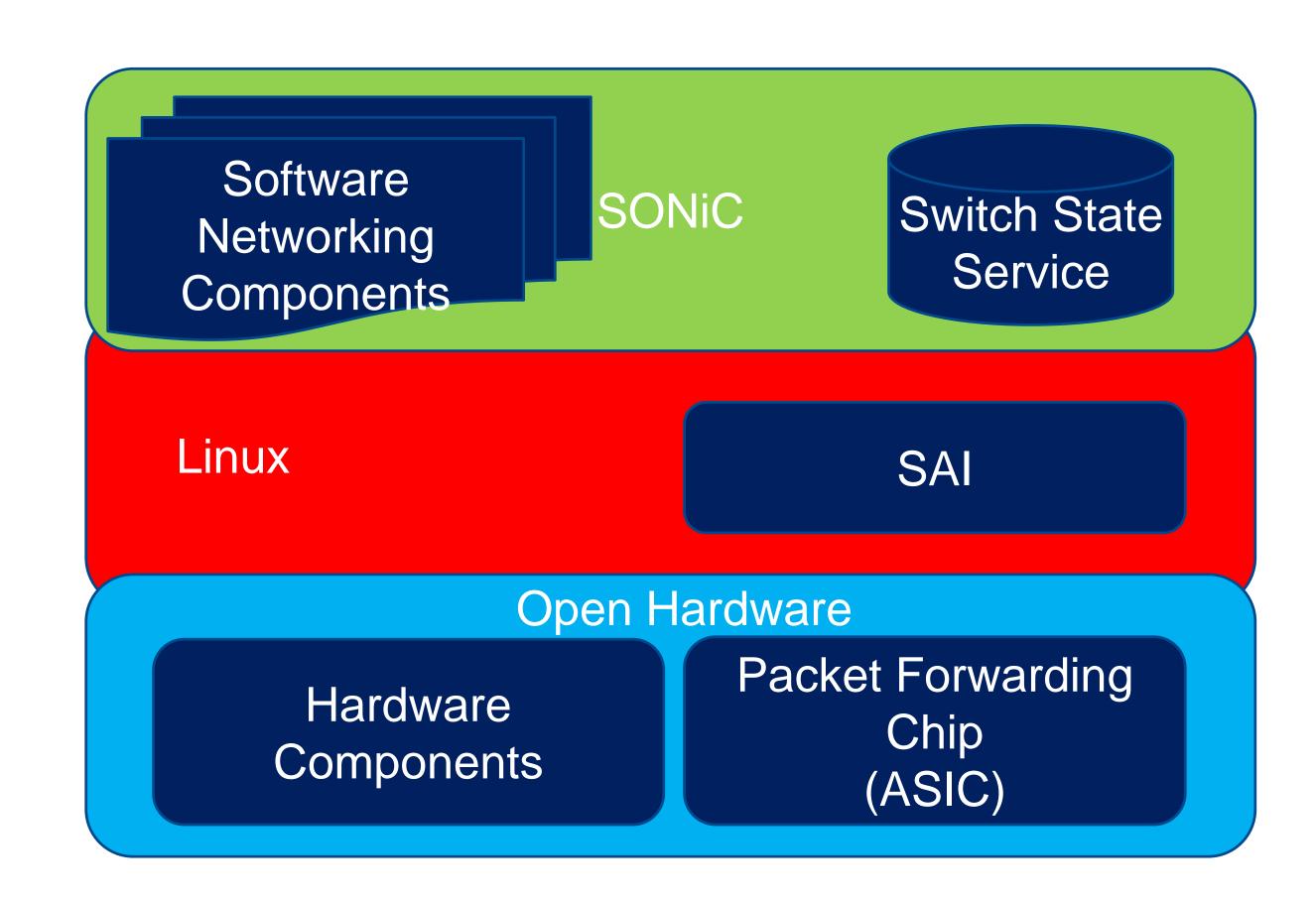
- Heading towards testing compliancy

- Working on a logical pipeline

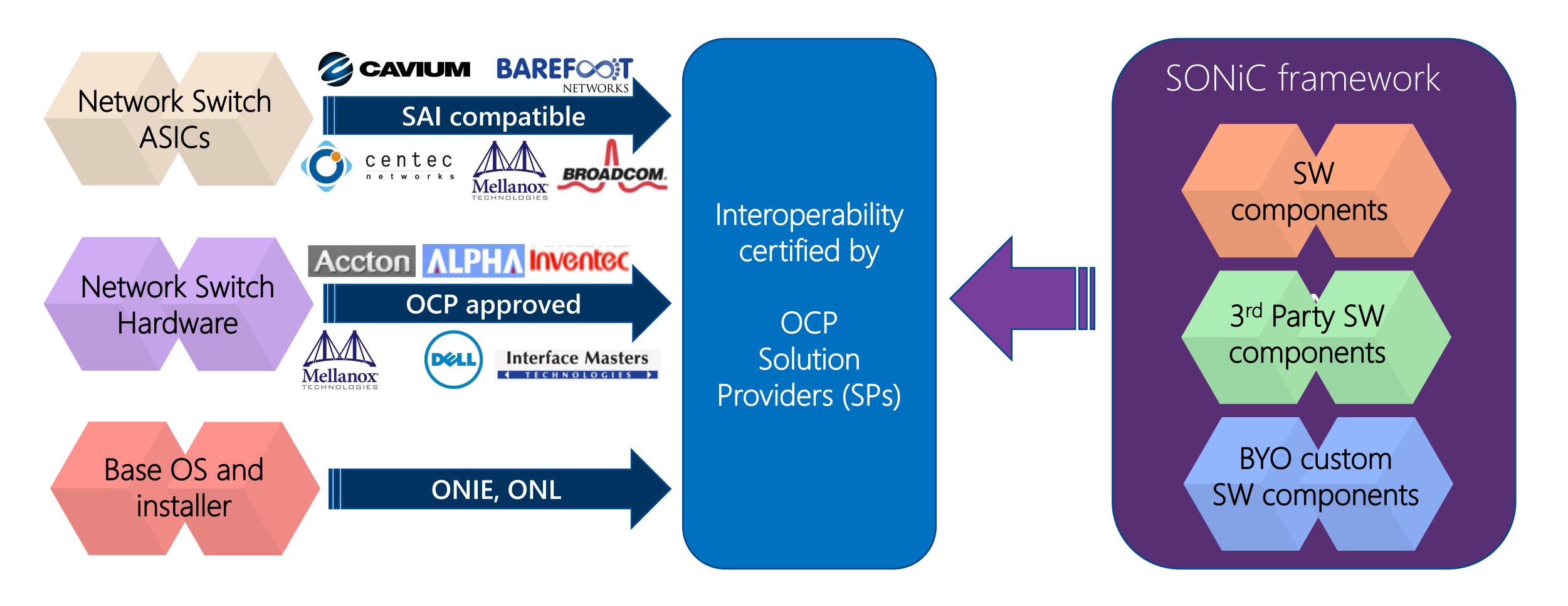




#### SONiC and the OCP Stack



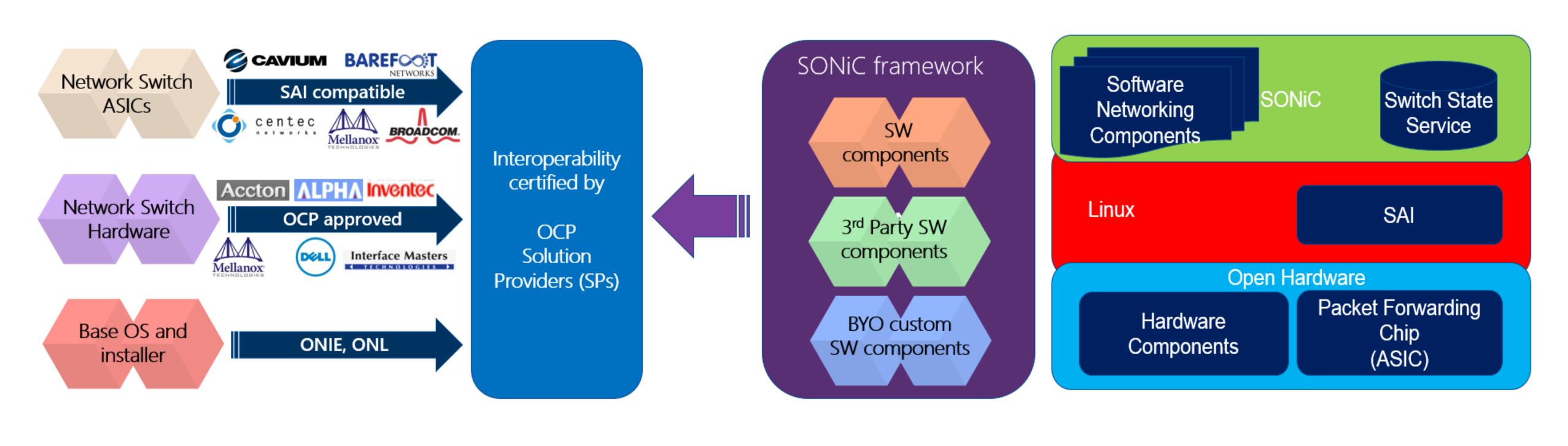
### OCP Ecosystem Enhanced with SONiC



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

# Software for Open Networking in the Cloud (SONiC)

Kamala Subramaniam Prinicipal Architect, Microsoft





#### 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

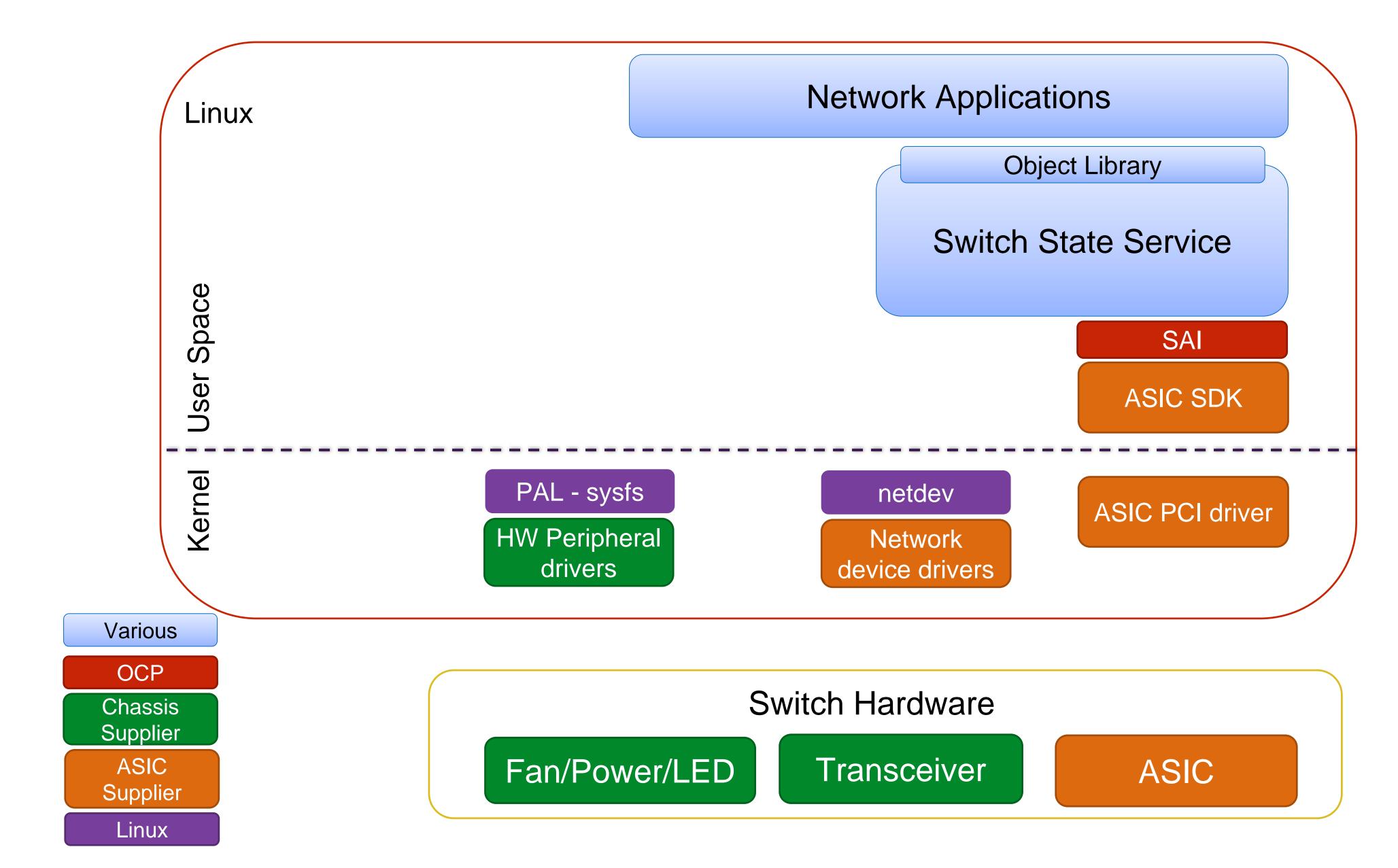
#### Easy portability

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

#### 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

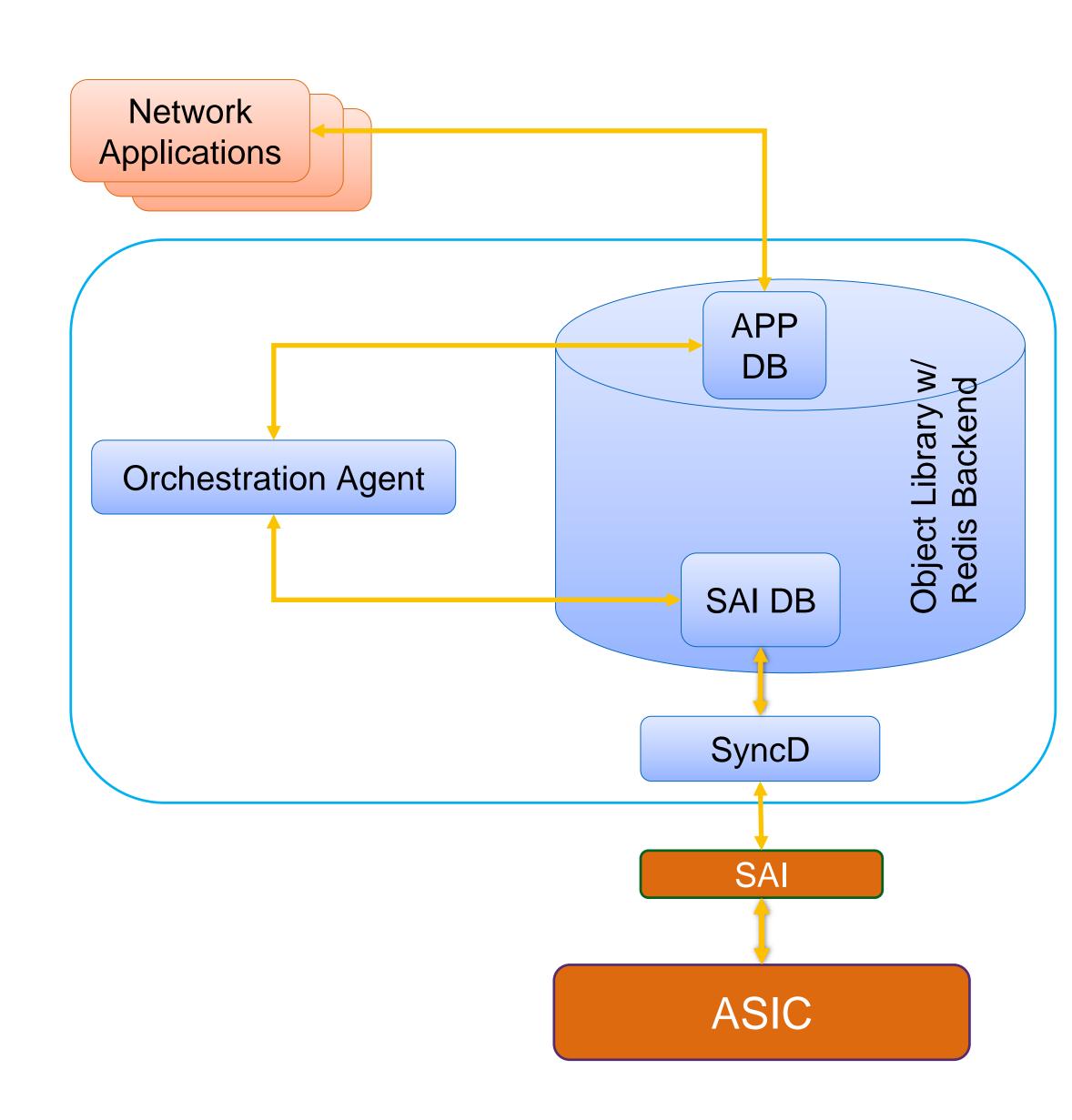
DB backend: redis with object library

SyncD: sync SAI objects between software and

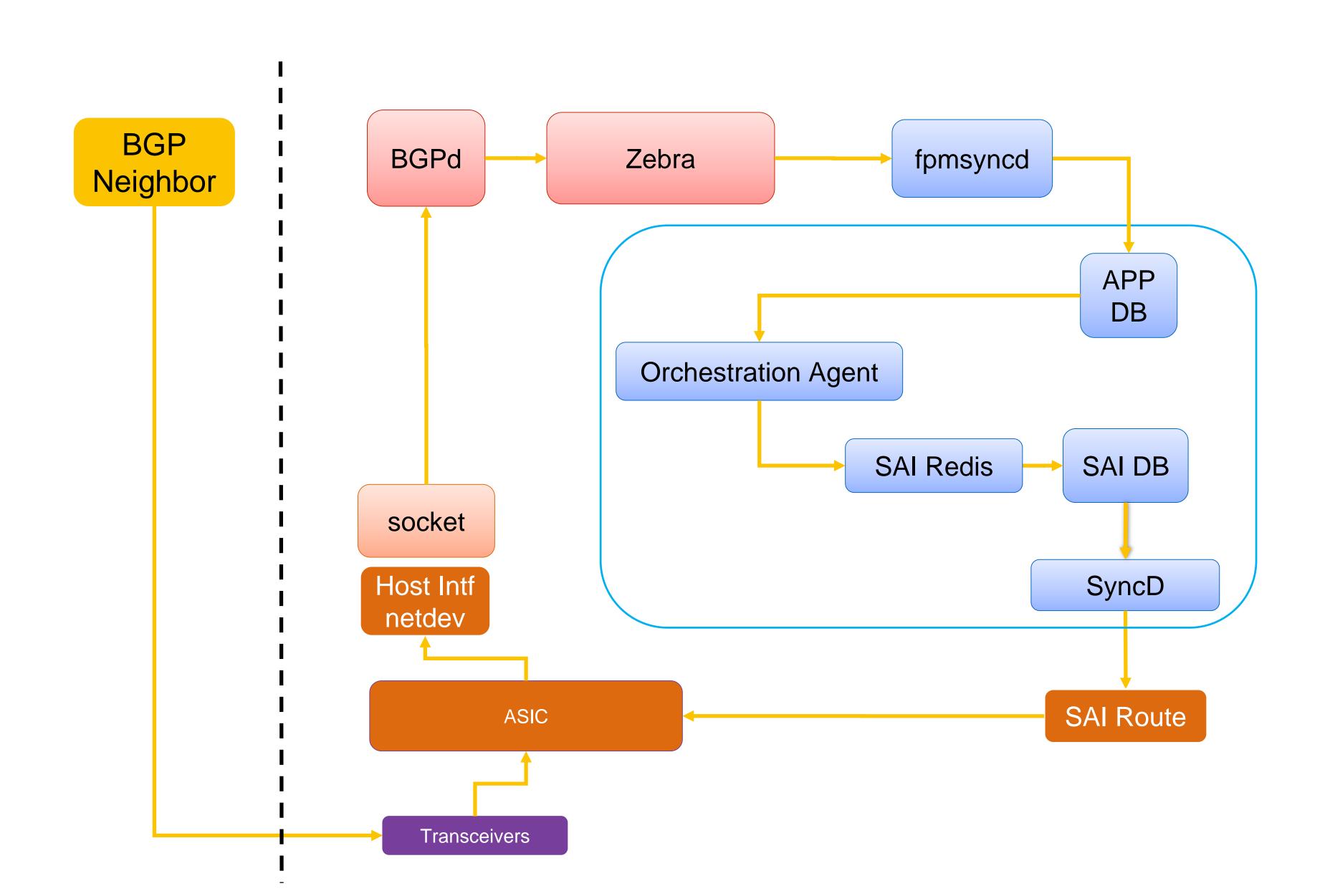
hardware

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

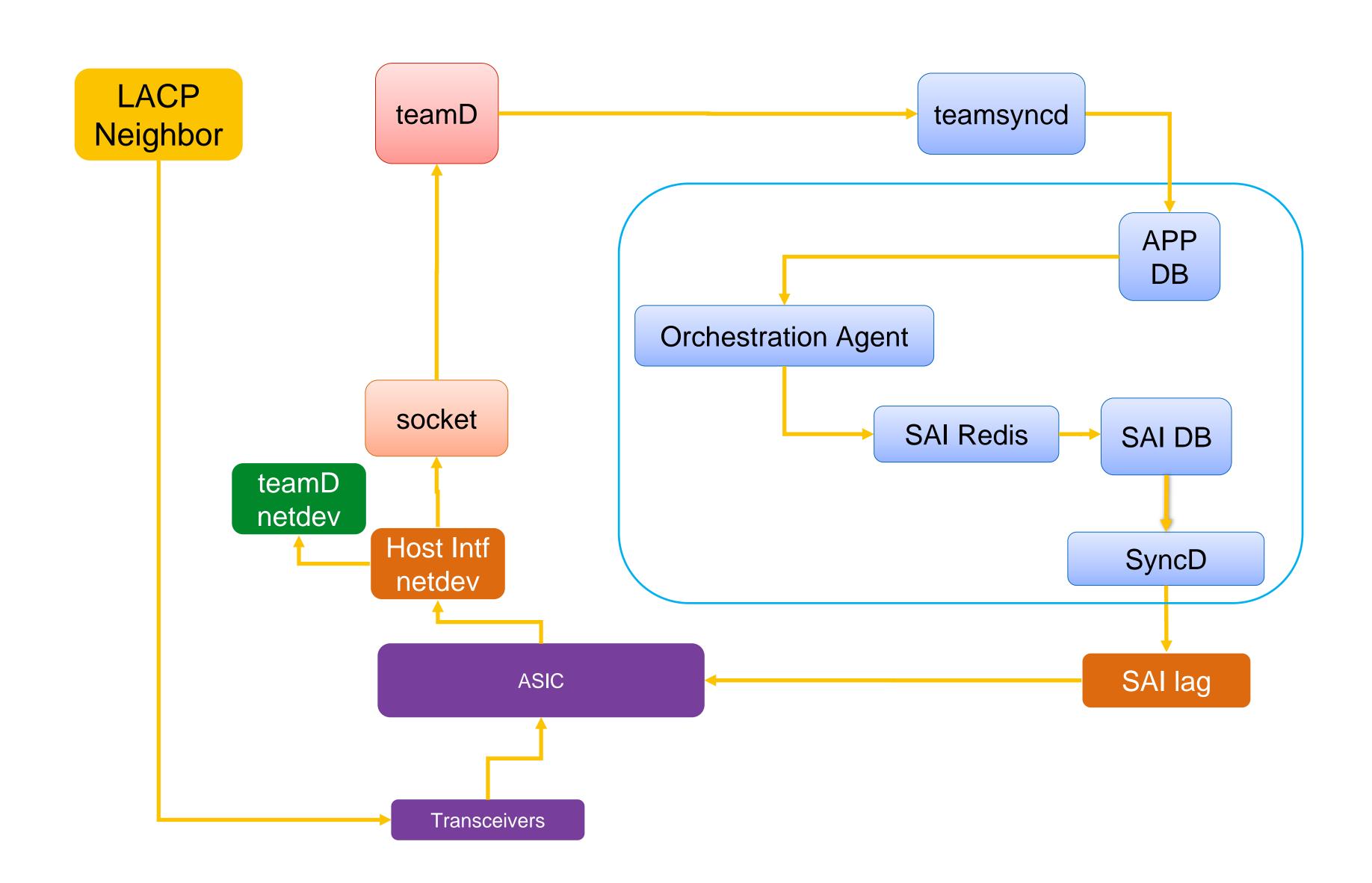
**Key Goal:** Evolve components independently



### How Routing Works in SONiC



#### How LAG Works in SONiC



### Demo: SONiC Walk Through

EEPROM Port Status TCPDump Redis Quagga and FIB

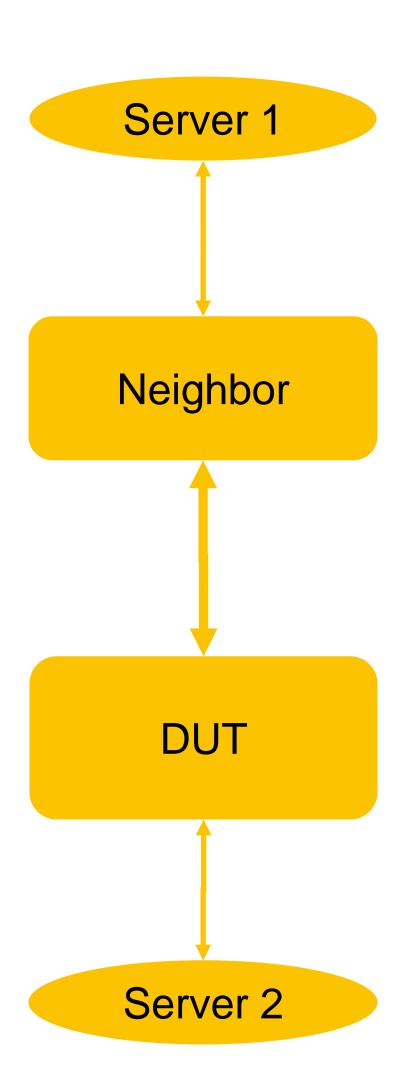
### Demo: Hitless Quagga to GoBGP Migration

#### What is the real scenario?

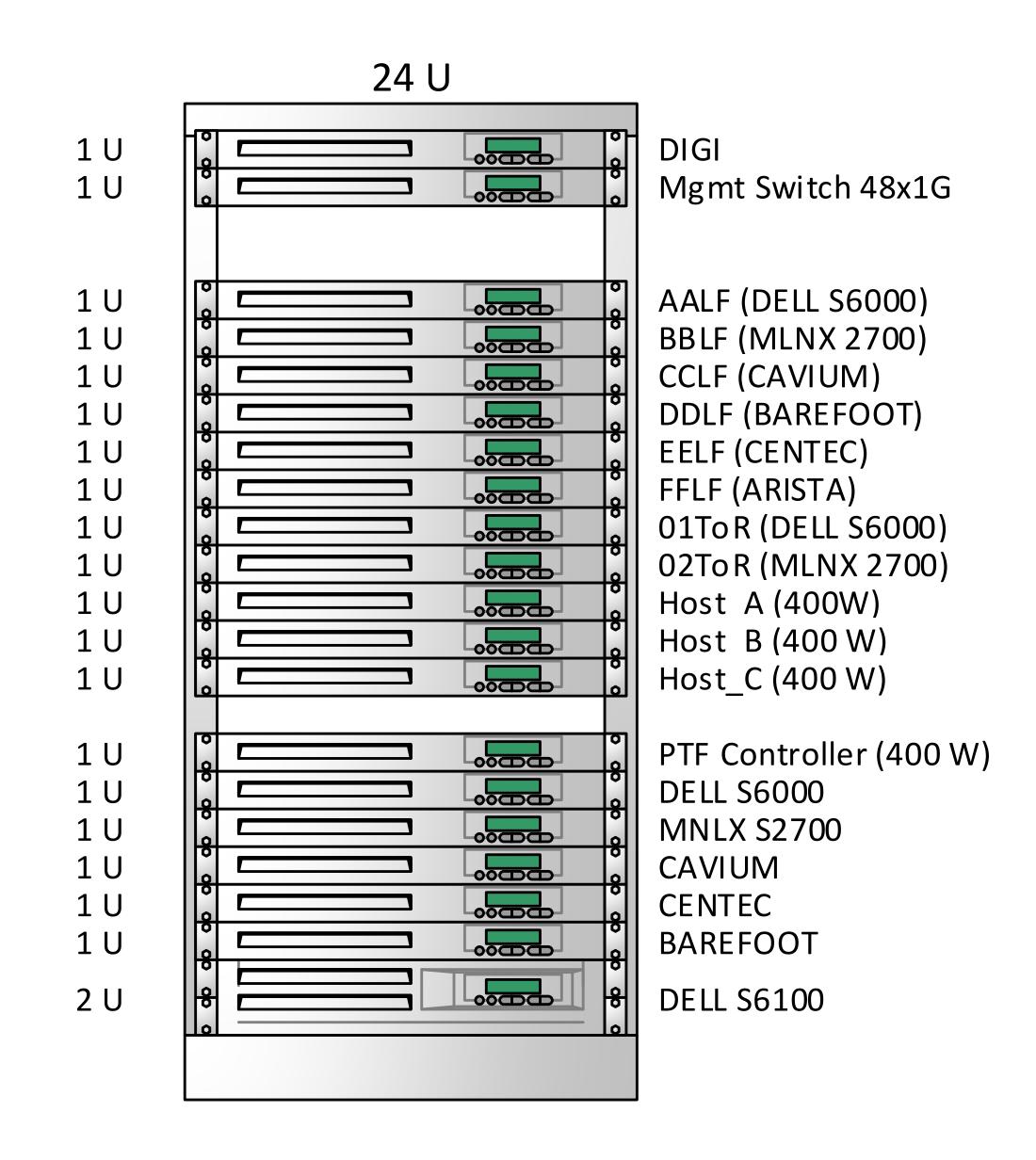
- Hot patching
- Rollout new features, e.g. BMP

#### How is this achieved?

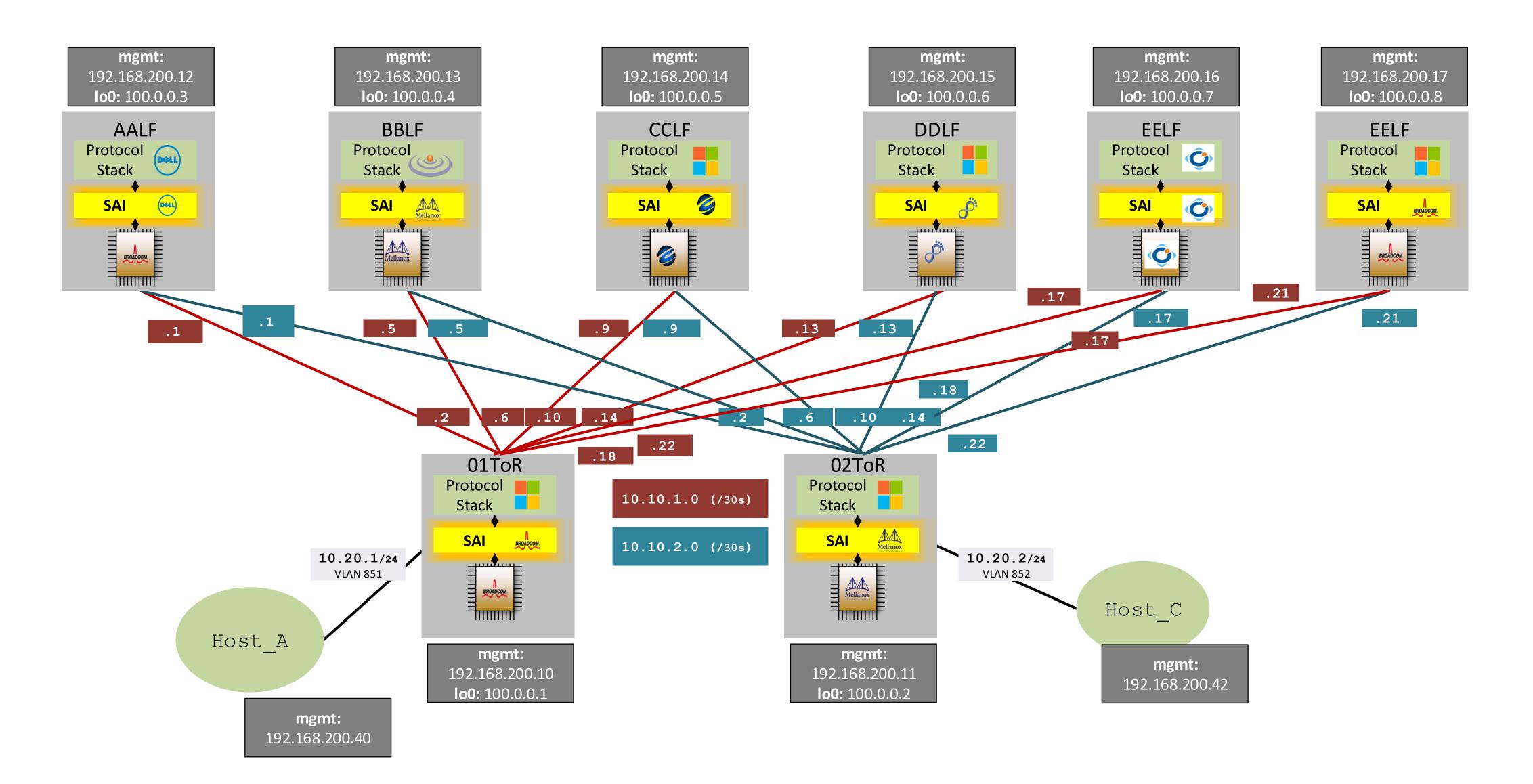
- 1. Neighbor supports graceful restart
- 2. DUT request OA to freeze FIB
- 3. DUT uninstalls Quagga
- 4. DUT installs GoBGP
- 5. DUT wait for route convergence
- 6. DUT request OA to unfreeze FIB



#### Demo at the Microsoft Booth

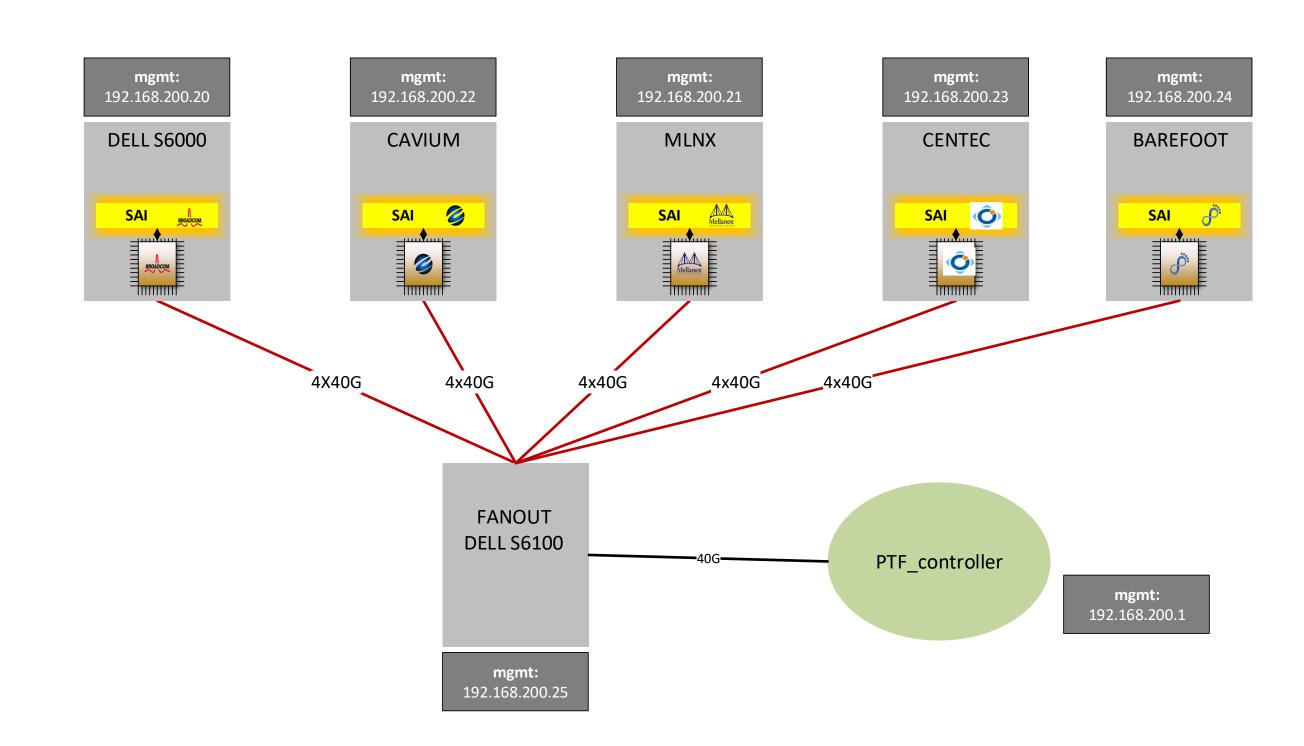


### Feature Functionality Demo



#### 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



### Open Invitation

- Inviting contributions in all areas
  - SAI
  - Hardware platform
  - Base OS platform (Ubuntu/ONL happening)
  - New features and applications
  - Use it!

- Starting Point: http://azure.github.io/SONiC/