

SONiC: Software for Open Networking in the Cloud

SONiC – Software for Open Networking in the Cloud

What is SONiC

- Software components for building network switch with rich functions
- Based on SAI, agnostic to ASIC and switch hardware platforms
- Deployed in MSFT data centers and running production traffic
- 100% open sourced with OCP community support

Contributors



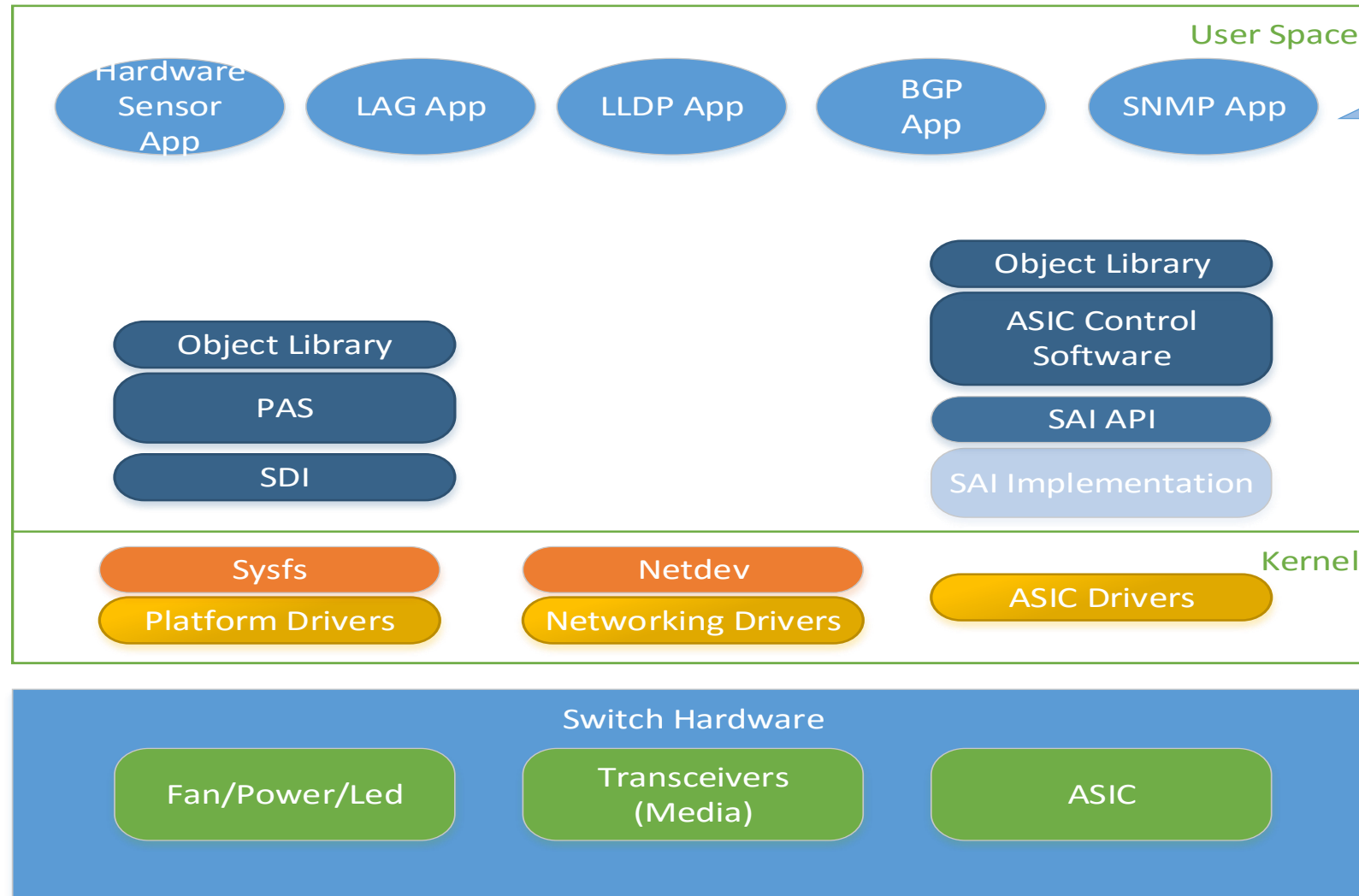
Why SONiC

- A key part of the SDN solution
- Reduce capex with more choices on hardware platforms
- Reduce operation cost with high quality software
- Allow advanced switch management/deployment as servers
- Take the best from the community and contribute to the community

Journey

09/14	SAI first release
03/15	OCP demo with basic router functionalities
09/15	SAI acceptance to OCP, 1st SONiC deployment in production
03/16	SONiC open sourced, 3 SAI releases per year
06/16	CO4 all AP T1s on SONiC
12/16	SONiC V2 release, SAI 1.0 release

SONiC Architecture



Many more apps to come. Build, borrow or buy.

Applications

SONiC Open Source

Other source/bin

Linux Kernel Interface

Drivers

Hardware Components

SONiC Feature Roadmap

Current

BGP

ECMP

WRED

LAG

SNMP

Syslog

LLDP

NTP

TcpDump

AD based AAA

Image install using ONIE

Incremental update

In Progress

EverFlow
/Mirror

Fast Reload

ECN

ACL

IPv6/BGP
test

DHCP
Relay/Test

RDMA/
Test

QoS/
Test

Netbouncer
/Test

CoPP/
Test

VLAN
/Test

MMU

Looking Forward

Warm Reboot

Critical Resource Monitoring

VxLAN

And more... driven by
community

	Feature	Design	Implementation	Test
R1 (10/30)	LAG			
	Copp			
R2 (11/30)	DHCP Reply			
	VLAN			
	RDMA			
	QoS			
	Netbouncer/IP Decap			
	ACL (MLNX)			
R3 (12/30)	IPv6			
	Everflow/Mirror (MLNX)			
	ECN (Pavel)			
R4 (1/30)	Fast Reload			
Future	Critical Resource Monitoring			
	Static break out ports			
	Warm Reboot			
	Vxlan			

FAQ

- Linux OS version: Debian Jessie 8.4 - Kernel 3.16.0-4-amd64
- SAI version: <https://github.com/opencomputeproject/SAI/tree/v0.9.4>
- How to build SONiC: <https://github.com/Azure/sonic-buildimage>
- How to deploy SONiC: <https://github.com/Azure/SONiC/wiki/How-to-Deploy-Sonic>
- How to write PTF test: <https://github.com/Azure/SONiC/wiki/HOWTO-write-a-PTF-Test>
- Developer guide: <https://github.com/Azure/SONiC/wiki/Sonic-Developers-Guide>

Sonic Software Management

- Software is divided in to two layers
 - Base Image – Linux plus minimum components needed to boot
 - Component management
- Base image is installed using [Open Network Install Environment](#) - [ONIE](#)
 - Power on, ONIE loads
 - Send DHCP request with option 60 (RFC 2132)
 - DHCP server responds with URL containing image file (sonic-generic.bin)
 - Switch downloads image, installs and reboots into Sonic base OS
- After base image install
 - ONIE no longer executes, the base image is just run
 - One can change grub options to cause ONIE to start on reboot

Sonic Management - Automation

- Packages and Containers are installed using Ansible and Docker
- Sonic-mgmt has open sourced playbooks
 - Deployment for installing and updating sonic
 - Test roles for running tests and controlling virtual lab
- Allows for incremental update and rollback
 - Orchestrated by Ansible
 - Allows in service upgrades for most components
 - Docker containers package up most components
 - Debian packages applied to update base image
 - Most incremental updates are expected to be dockerized
- Today reboots are required for some package updates
 - Things that require hardware/device driver support require kernel changes
 - Some hardware support is via kernel modules which may be reloaded without reboot

SONiC Development/Deployment

