



# Deep dive into SONiC

---

## Architecture & Design

Kanza Latif (April 2024)

# Table of Contents

<b>Summary</b>	<b>3</b>
<b>Key Points</b>	<b>3</b>
<b>Conclusion</b>	<b>4</b>

# Summary

The blog highlights SONiC (Software for Open Networking in the Cloud) as a groundbreaking open-source network operating system (NOS) that challenges traditional and proprietary NOS by redefining network infrastructure administration.

## Key Points

### 1.Modular Structure

SONiC's architecture is based on Debian Linux and the Switch Abstraction Interface (SAI) layer, which acts as a translator between silicon and software. This modular design allows SONiC to work with multi-vendor silicon, reducing vendor lock-in and providing a glimpse into the future of network disaggregation.

### 2.Containerized Architecture

SONiC encapsulates all major subsystems within Docker containers, facilitating inter-process communication and ensuring serviceability, extensibility, development agility, and resource efficiency.

### 3.Application Containers

SONiC employs user-space services within containerized modules, including FRR routing stack, LLDP for topology mapping, SNMP server for monitoring, PMON for hardware management, DHCP-Relay for IP assignments, and more. Modularity enables tailored configurations to meet diverse network needs.

### 4.Infrastructure Containers

Switch State Services (SWSS), Database, and Synchronization daemon (SyncD) play fundamental roles in SONiC's management and configurations. SWSS maintains state consistency among various modules and applications, while SyncD synchronizes network state and configuration changes with hardware.

## 5.Switch Abstraction Interface (SAI)

SAI serves as an intermediary between SONiC and hardware-specific switch ASICs, enabling SONiC to work with a wide range of switch hardware without extensive modifications. SyncD leverages SAI API to interact with hardware components using ASIC SDKs.

## 6.Linux Kernel Integration

SONiC leverages Linux kernel capabilities for network isolation, inter-process communication, ACL functionalities, switch memory management, NAT support, and configuration management.

## 7.Configuration Management

SONiC offers multiple ways to configure the NOS, including SONiC-CLI, REST API, and minigraph. Free Range Routing (FRR) CLI and YANG models play crucial roles in defining and managing network configurations.

## Conclusion

SONiC revolutionizes network management with its modular, vendor-agnostic design, broad hardware support, and programmability. Its open-source, containerized architecture empowers network operators to tailor their infrastructure, fostering automation and adaptability in the dynamic data center landscape. SONiC represents a future where open-source, extensible solutions simplify modern networking for administrators.

