

## INSIDE

### EXTENSIBILITY

is a system design principle where the implementation of the operating system takes into consideration future growth. It is a systemic measure of the ability to extend the operating system and the level of effort required to implement the extension.

Extensions can involve the addition of new functionality or the modification of existing functionality.

### WHY

Provide the mechanism to change and improve capabilities of the system without impacting the core system function

### WHO CARES

Network Administrators who are measured on system uptime and want to deliver rapid service restoration when failures happen

### WHAT IS NEXT

Stable software should be table-stakes, but managing large-scale deployments at scale with the familiar tools you know today? That is novel.

## Data Center Class

# Extensible Operating System

*What is Arista's core operating system architecture and structure? How flexible and extensible is it? Can you really add/drop functions as claimed?*

The Arista Extensible Operating System, or EOS, is the most advanced network operating system available. It combines modern-day software and O/S architectures, transparently restartable processes, open platform development, Linux kernel, and a stateful publish/subscribe database model that enables a much more scalable ongoing software development.

Some examples of the EOS architecture in action are:

- The live patching of a software component such as Spanning Tree or a routing protocol can be accomplished on an active system without full system restart.
- With the hardware device drivers residing in the user space of the operating system Arista has compiled multiple ASIC device drivers into the same binary of EOS. A customer can take the EOS

binary from the smallest Arista switch and load it into the largest and it boots fully functional.

- The same EOS image can then be loaded into a Virtual Machine for testing, or lab environments.

Arista's EOS is extremely flexible and stable. At the core of EOS is an operating system architectural component unique in network operating systems - the System Data Base, or SysDB for short. SysDB is machine generated software code based on the object models necessary for state storage for every process in EOS.

All interprocess communication in EOS is implemented as writes to SysDB objects. These writes propagate to subscribed agents, triggering events in those agents. i.e when a user-level ASIC driver detects link down it writes this to SysDB, then the LED driver receives an update from SysDB and it reads the state of the port and adjusts the LED status accordingly. SysDB is generated on-demand and compiled automatically as Arista builds EOS software- this reduces risk and error, while

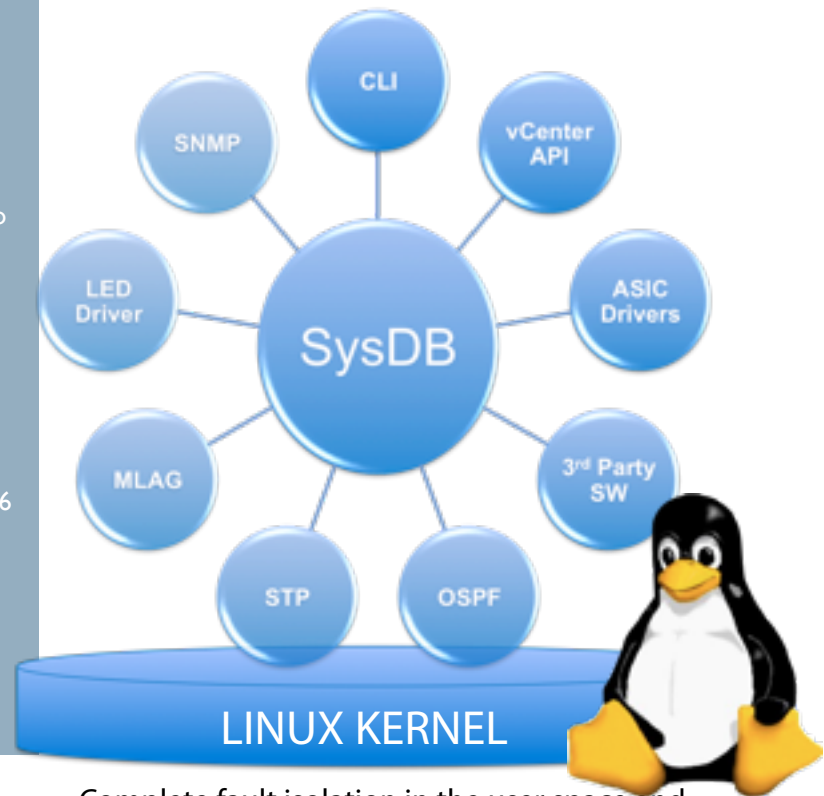
## EOS EXTENSIBILITY

### WHAT CAN I RUN?

**Be Creative!** Arista EOS supports custom development. Want to back up all log files every night to a specific NFS share? How about run MRTG right on your switch? Like the Internet2 PerfSonar performance management apps? Or want to run Nessus to security scan a server when it boots? Like using Screen to ease terminal sessions?

Run what you want, and rest assured we have ample CPU capacity available with a dual-core 1.8Ghz AMD x86 CPU and trusted Linux scheduler.

*Arista's Cloud Network Architecture is about bringing 'openness' and 'extensibility' to network operating systems. Choose your own path.*



improving software feature velocity for customers.

Arista EOS has full Linux shell access for root-level administrators, and makes a broad suite of Linux based tools available to our customers. In the spirit of 'openness' the full SysDB programming model and API set are visible and available via the standard bash shell.

SysDB is not a "walled garden" API, where a limited subset of what Arista uses is made available. All programming interfaces that Arista SW developers use between address spaces within EOS are available to third party developers, and Arista customers.

*How does Arista's software development methodology benefit data centers?*

The software engineering methodology at Arista has significantly benefitted our customers in terms of quality and consistency. Some examples are:

- Complete fault isolation in the user space and through SysDB effectively converts catastrophic events to non-events. The system self-heals from more common scenarios such as memory leaks. Every process is separate, no IPC or shared memory fate-sharing, endian-independent, and multi-threaded where applicable.
- No manual software testing. All automated tests run 24x7 and with the operating system running in emulators and on hardware Arista scales protocol and unit testing cost effectively.
- Keep a single system binary across all platforms. This improves the testing depth on each platform, improves time-to-market, and keeps feature and bug compatibility across all platforms.

EOS, and at its core SysDB, provide a development framework that enables the core concept - extensibility. An open foundation, and best-in-class software development models deliver feature velocity, improved uptime, easier maintenance, and a choice in tools and options.

NETWORK OPERATING SYSTEM EXTENSIBILITY - JANUARY 2010