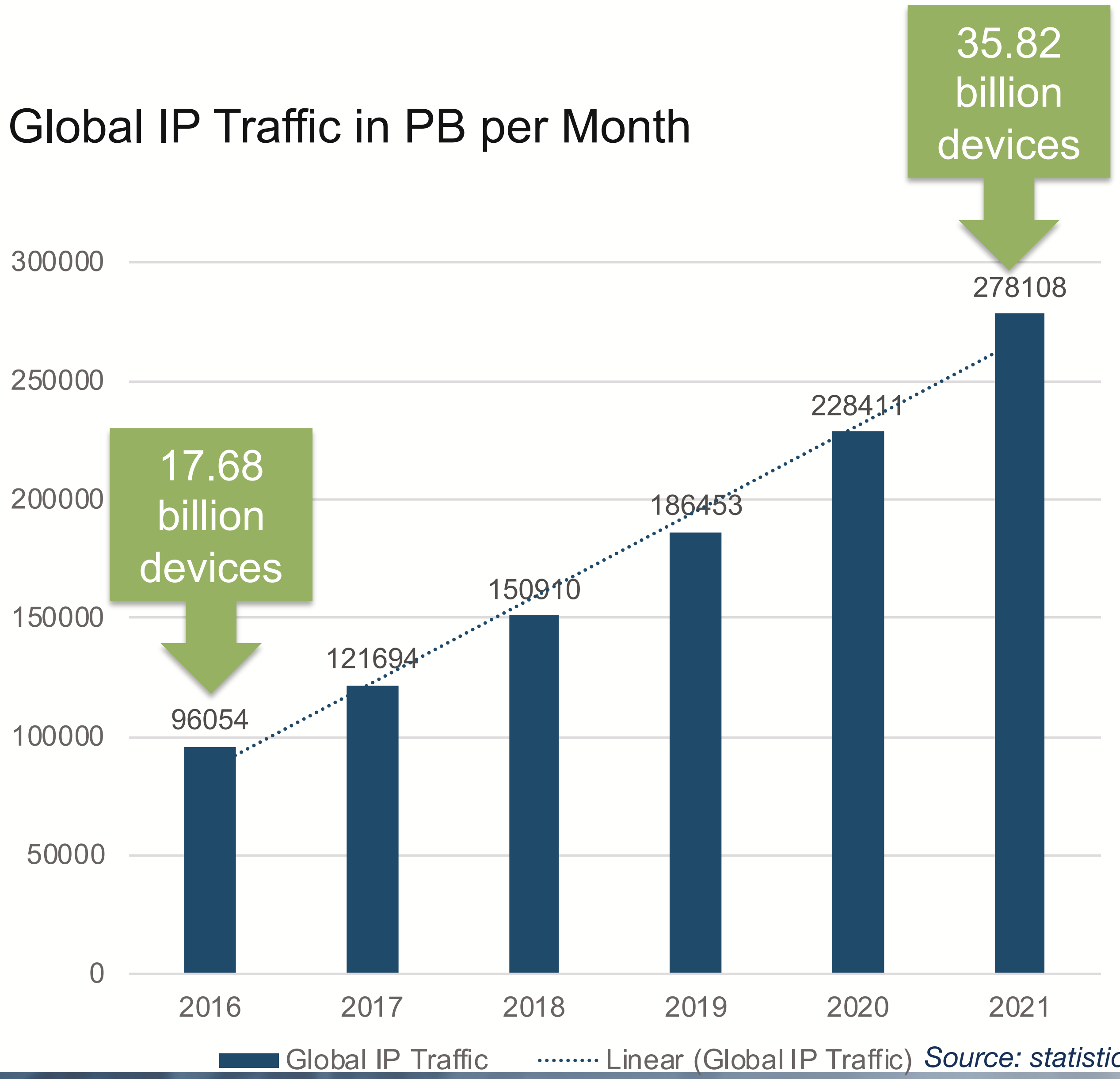
Approaches to Network Automation

Addison Chi
Engineering Manager, Greater China
addison@arista.com

The Case for Network Automation

Global IP Traffic in PB per Month



17.68 billion devices

35.82 billion devices

Source: statistica.com

Out of 278 EB globally, IP traffic in Asia Pacific will reach **108 EB per month** by 2021, growing at a CAGR of 26%

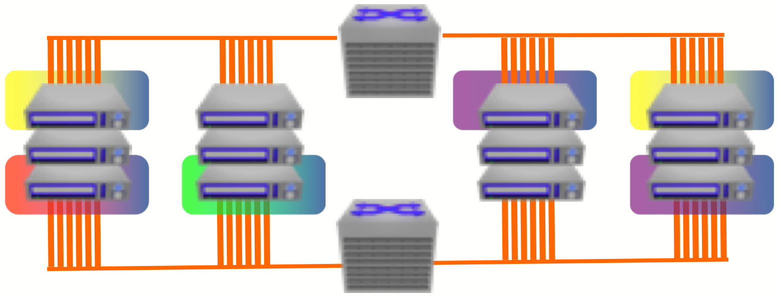
Drivers: IoT, Mobile adoption, Big Data, Machine Learning, Artificial Intelligence, CDN, VoD, AR/VR

Today: 80% of all operating expense is spent in managing daily operations

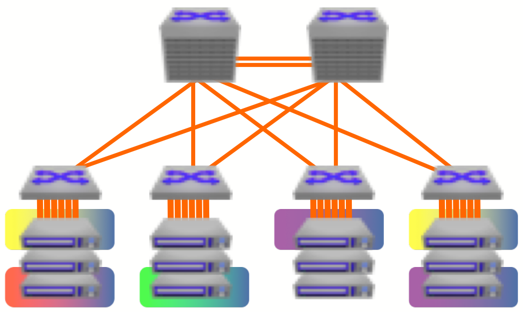
What does this mean for NetOps in 2021?

The Software Defined Data Centre

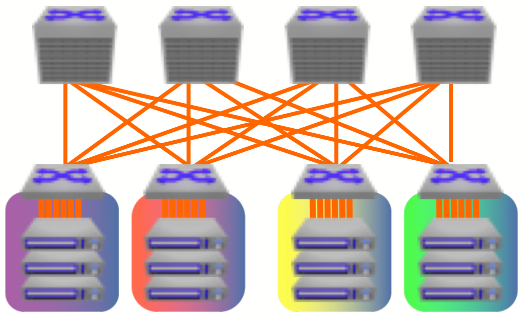
Vertically integrated, proprietary stacks	→	Open technologies, maximum generalization
Infrastructure specific to specific apps	→	Applications abstracted from infrastructure
Vendor lock-in, Forklift refreshes	→	Best-of-breed, continuous innovation
Multiple management domains	→	Homogenous, universal automation
Complex and custom architectures	→	Simple, repeatable and scalable architectures



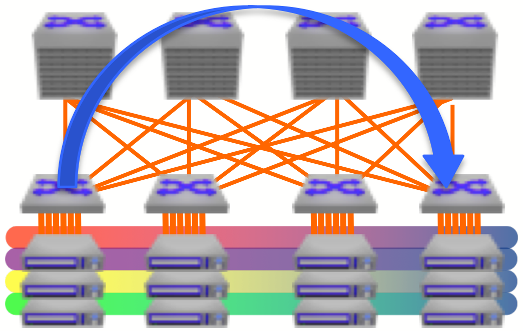
Spline™



Layer 2 / MLAG

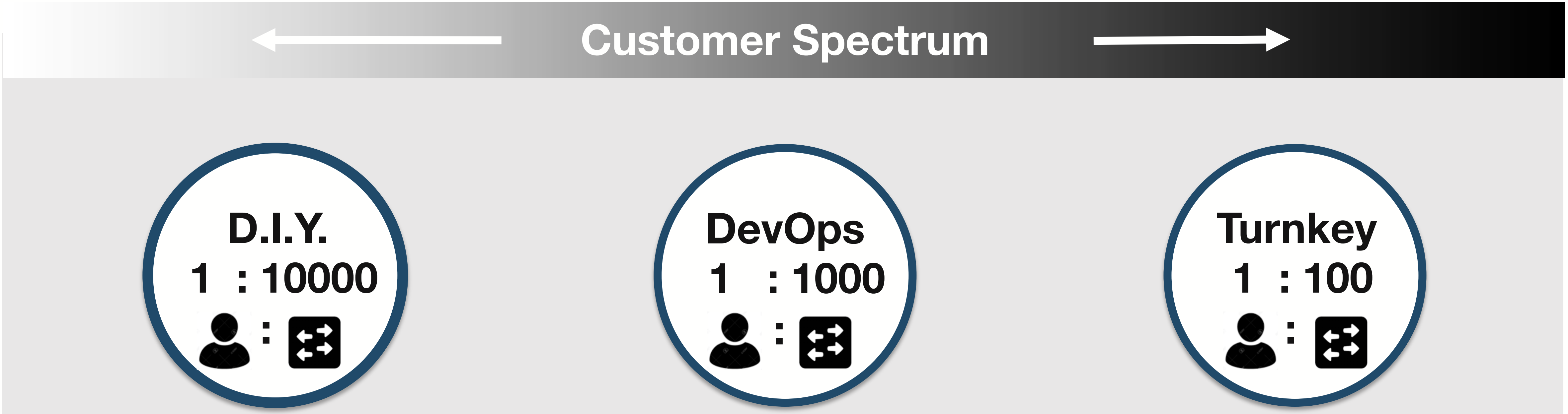


Layer 3 / ECMP



L2 over Layer 3 VXLAN

Approaches to Cloud Network Automation



Customer Spectrum

D.I.Y.
1 : 10000

DevOps
1 : 1000

Turnkey
1 : 100

Custom development for integration

Software development resources & approach

SDK/API

Leveraging existing tools (Puppet, Chef, etc)

Integration & customization via scripts

DevOps Toolkit

Limited solutions today, focus on legacy models

Customers need turnkey solution for cloud-automation!

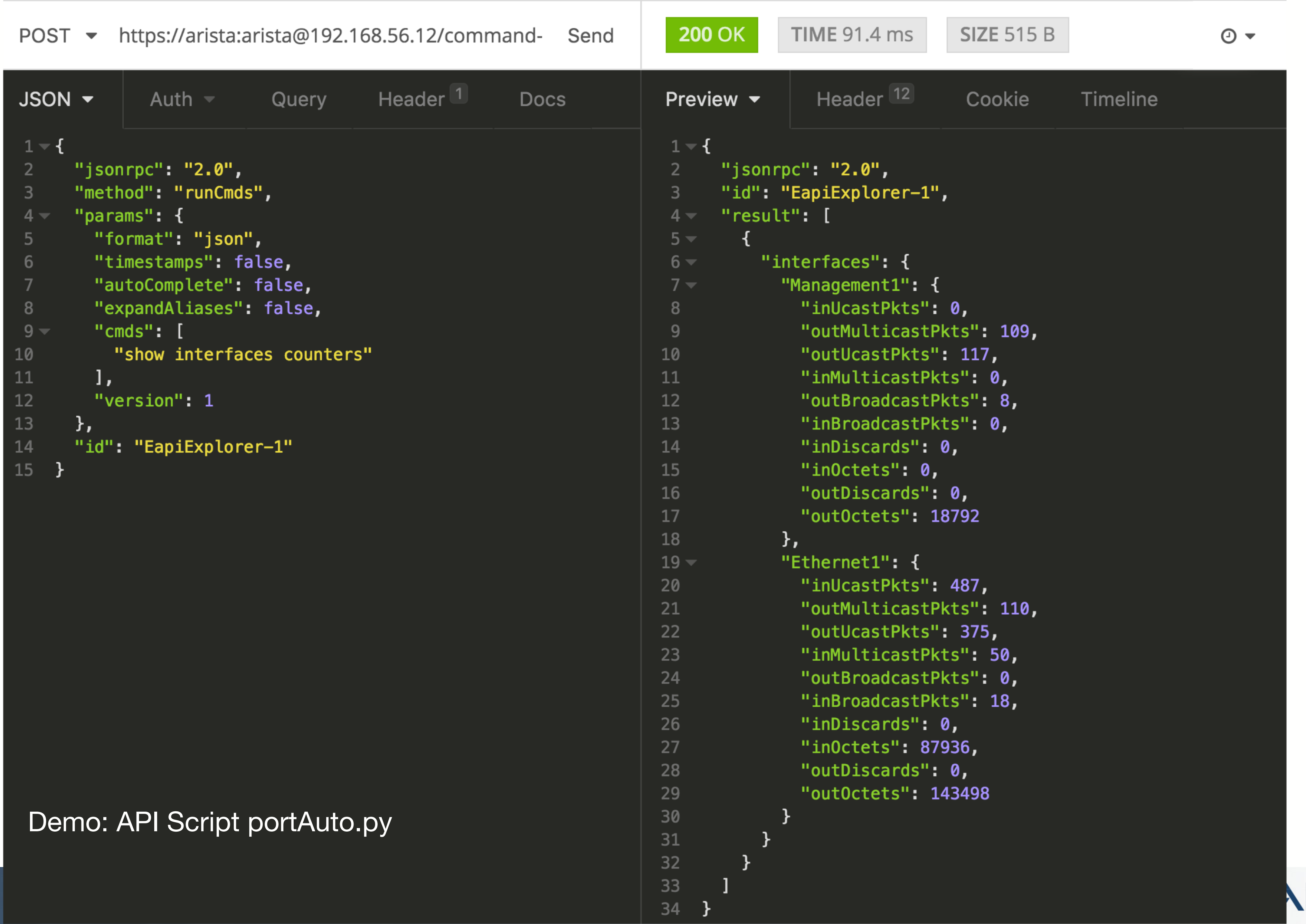
Software Platform

DIY Approach: Software Development Kit

- Write native apps for your switch
- Enables custom control plane
- Low-level integration
- APIs available in many languages – Python, C++ etc
- Use Cases:
  - Program static routes into your network
  - Program custom multicast control planes
  - React to ARP entry or neighbour discovery
  - React to port channels – when LAG changes speeds or member gets added/removed
  - Monitor hardware capacity real-time
  - Add/drop traffic based on bandwidth utilisation, priority, within defined thresholds

DIY Approach: API

- API is a simple method of remotely interacting with a switch without screen scraping
- HTTP or HTTPS and uses JSON (JavaScript Object Notation)
- Full configuration supported – many show commands supported
- API allows CLI Commands to be issued remotely
- API returns the output in a programmable-friendly format (JSON) and generally in key-value pairs
- Useful when you need to automatically read or control a remote switch (automation)!



```
{
  "jsonrpc": "2.0",
  "method": "runCmds",
  "params": {
    "format": "json",
    "timestamps": false,
    "autoComplete": false,
    "expandAliases": false,
    "cmds": [
      "show interfaces counters"
    ],
    "version": 1
  },
  "id": "EapiExplorer-1"
}
```

```
{
  "jsonrpc": "2.0",
  "id": "EapiExplorer-1",
  "result": [
    {
      "interfaces": {
        "Management1": {
          "inUcastPkts": 0,
          "outMulticastPkts": 109,
          "outUcastPkts": 117,
          "inMulticastPkts": 0,
          "outBroadcastPkts": 8,
          "inBroadcastPkts": 0,
          "inDiscards": 0,
          "inOctets": 0,
          "outDiscards": 0,
          "outOctets": 18792
        },
        "Ethernet1": {
          "inUcastPkts": 487,
          "outMulticastPkts": 110,
          "outUcastPkts": 375,
          "inMulticastPkts": 50,
          "outBroadcastPkts": 0,
          "inBroadcastPkts": 18,
          "inDiscards": 0,
          "inOctets": 87936,
          "outDiscards": 0,
          "outOctets": 143498
        }
      }
    }
  ]
}
```

Demo: API Script portAuto.py

DevOps Approach

What is DevOps?

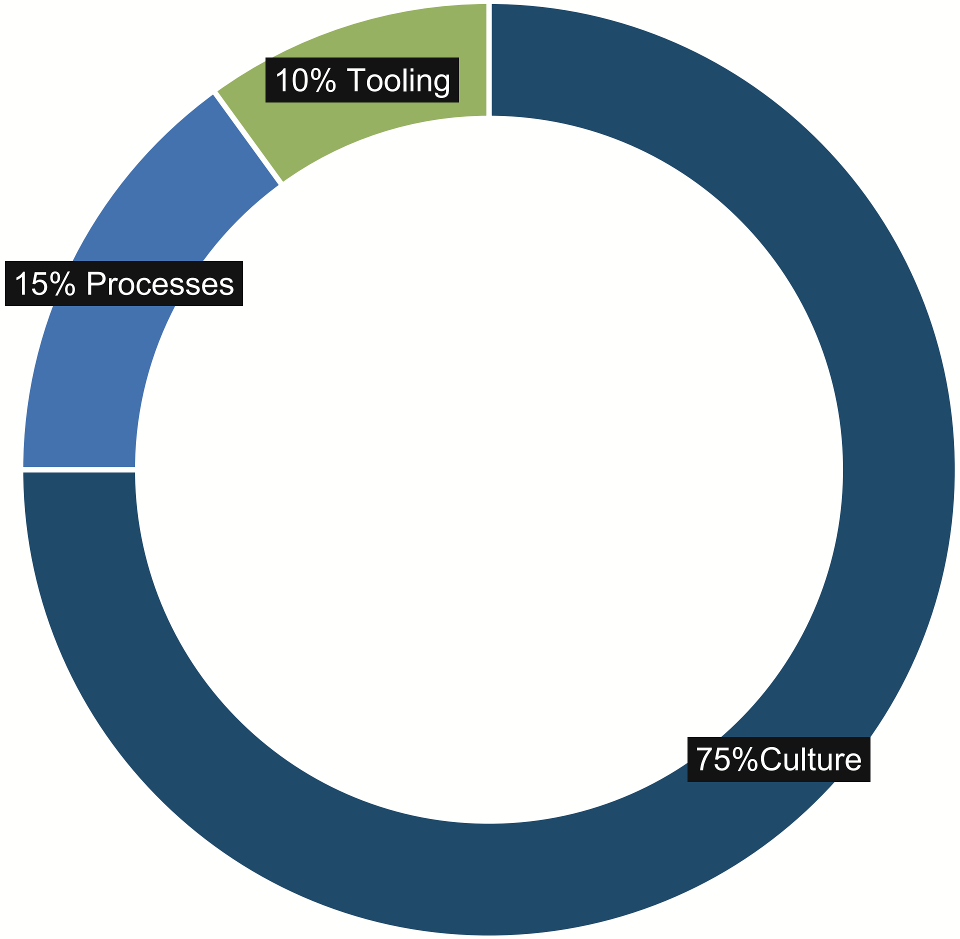
A culture, movement or practice that emphasizes the collaboration and communication of both software developers and other information-technology (IT) professionals while **automating the process** of software delivery and **infrastructure changes**.

GitHub	**Revision control** Git, backups, auditing, peer review, and access-control, blame ☺
	Continuous Integrations (CI) automated testing and workflows
ANSIBLE CHEF puppet	**Configuration management tools** like Ansible / Chef / Puppet **Continuous, consistent, auditing** Eg: New network modules in Ansible 2.1: eos_command, eos_eapi, eos_config
servicenow	**Change Control Management** schedule, authorize, track changes, approval workflows

A different look at DevOps

DevOps isn’t primarily about toolchains or processes - it’s more about culture than anything.

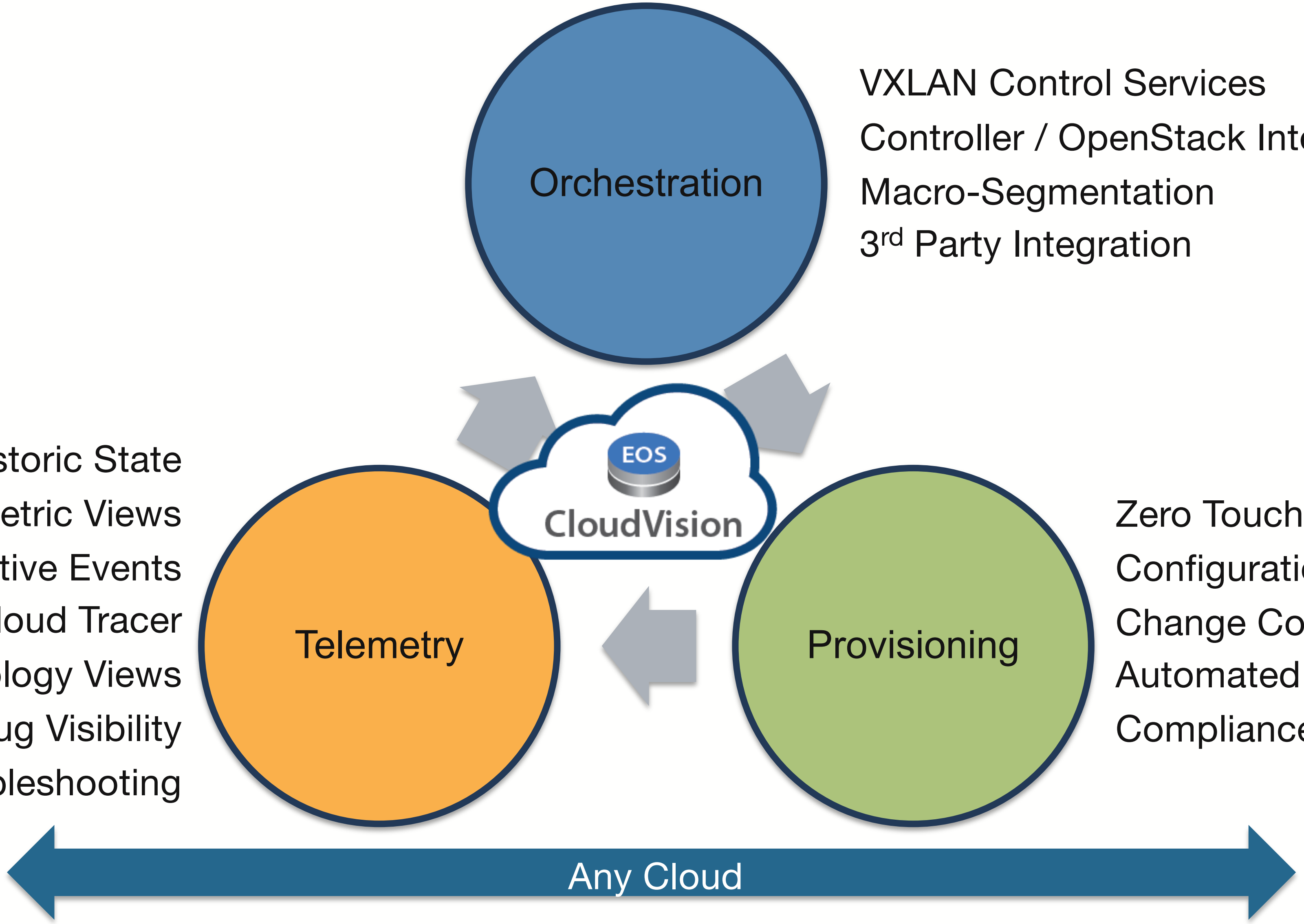
- Change Management
- Automated Testing
- Accelerated deployment
- Infrastructure as code
- Security & Compliance Audits
- Monitoring
- Increased availability
- Get your life back - **Spend more time doing <u>architecture</u>… and less adding VLANs!**



DevOps Approach: Configuration Management Tools

	ANSIBLE	puppet	CHEF	SALTSTACK
EOS Integration	Built-in	Forge modules	Cookbook	Minions
EOS Agent	None	EOS SWIX	el6 RPM	Yes
Architecture	Push	Pull	Pull	Continuous
Transport	SSH/SSL	SSL	SSL	ZeroMQ
Language	Python	Ruby	Ruby	Python Napalm
Community	Huge	4000	3000	Growing
Price	Free/Paid Ansible Tower	Free/Paid Puppet Enterprise	Free/Paid Chef Automate	Free/Paid Saltstack Enterprise

Turnkey Approach: Network Automation Pillars



VXLAN Control Services
Controller / OpenStack Integration
Macro-Segmentation
3rd Party Integration

Real-time/Historic State
Device/Metric Views
Events / Adaptive Events
Cloud Tracer
Topology Views
Bug Visibility
Forensic Troubleshooting

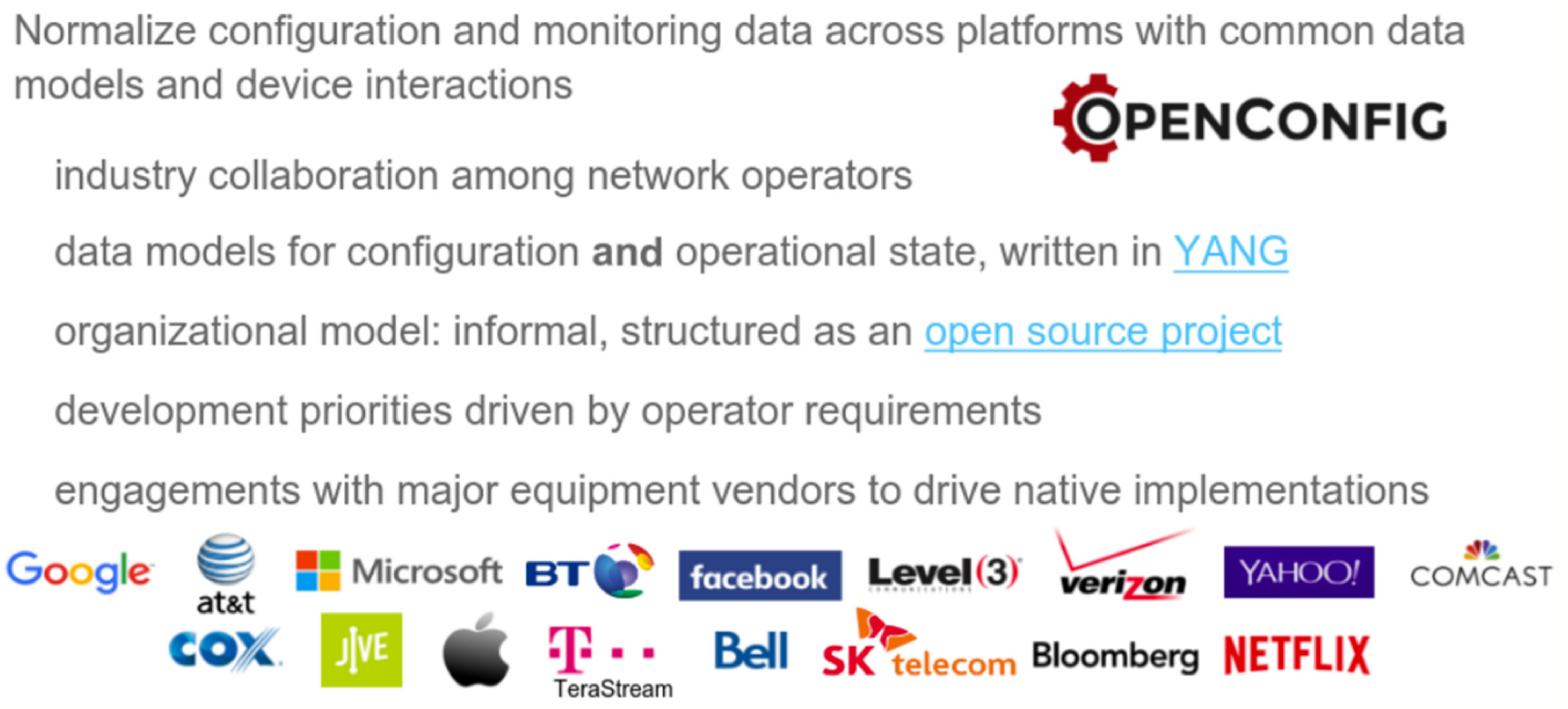
Zero Touch & Ongoing
Configuration Management
Change Control Management
Automated Upgrade/ Rollback
Compliance Audit

Demo: CVP

Automation Can Be Difficult

- Different vendors have different CLI’s
- Different vendors have different API’s
- Different vendors use different modules for Ansible, Saltstack, Chef, Puppet, etc.
- Different vendors return the same data in different formats (JSON, XML, etc)
- There is zero consistency in network automation today

Enter OpenConfig



Normalize configuration and monitoring data across platforms with common data models and device interactions

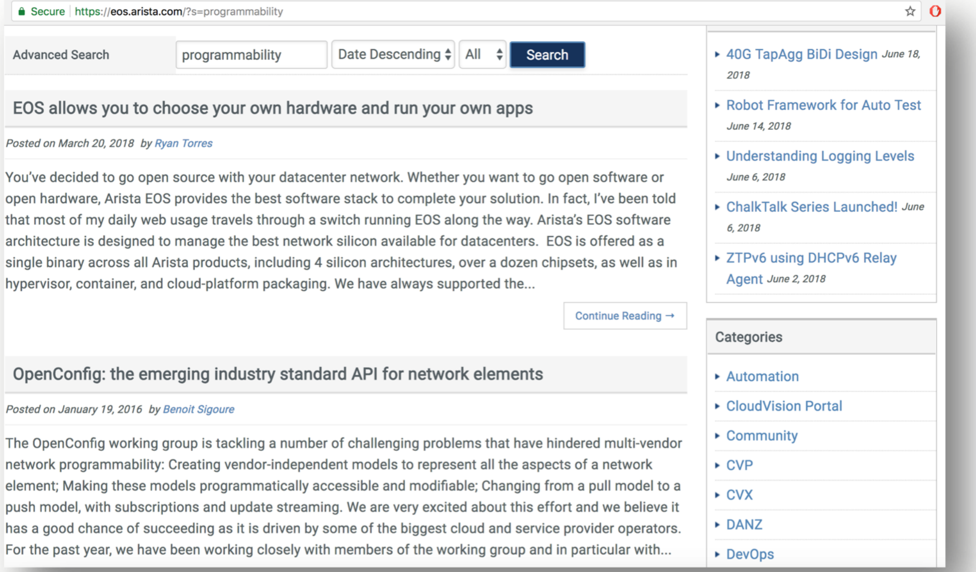
- industry collaboration among network operators
- data models for configuration **and** operational state, written in YANG
- organizational model: informal, structured as an open source project
- development priorities driven by operator requirements
- engagements with major equipment vendors to drive native implementations

Starting a Culture for Automation

- Start with ad hoc commands or simple one-liners
- Show value to the organization by demonstrating quicker provisioning times with fewer errors
- Begin conversations about treating infrastructure as code
- Find your friendly developers/QA teams and pair up with them – we find that the most successful organizations will pair a developer with a network resource
- Remember that this is also a huge cultural change that requires buy in from everyone – top down

Where Can I Learn More?

- EOS Central
  - https://eos.arista.com/
- Arista Github
  - https://github.com/aristanetworks
- SDK
  - https://github.com/aristanetworks/EosSdk
- API
  - https://github.com/arista-eosplus/pyeapi
- CloudVision
  - https://www.arista.com/en/products/eos/eos-cloudvision



EOS allows you to choose your own hardware and run your own apps

Posted on March 20, 2018 by Ryan Torres

You've decided to go open source with your datacenter network. Whether you want to go open software or open hardware, Arista EOS provides the best software stack to complete your solution. In fact, I've been told that most of my daily web usage travels through a switch running EOS along the way. Arista's EOS software architecture is designed to manage the best network silicon available for datacenters. EOS is offered as a single binary across all Arista products, including 4 silicon architectures, over a dozen chipsets, as well as in hypervisor, container, and cloud-platform packaging. We have always supported the...

Continue Reading →

OpenConfig: the emerging industry standard API for network elements

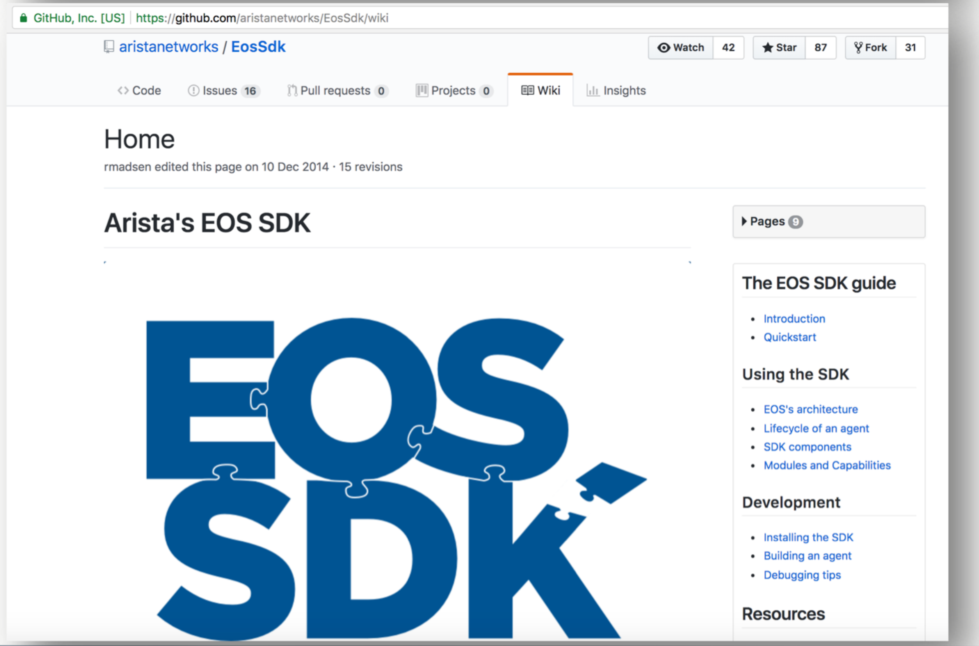
Posted on January 19, 2016 by Benoit Sigoure

The OpenConfig working group is tackling a number of challenging problems that have hindered multi-vendor network programmability: Creating vendor-independent models to represent all the aspects of a network element; Making these models programmatically accessible and modifiable; Changing from a pull model to a push model, with subscriptions and update streaming. We are very excited about this effort and we believe it has a good chance of succeeding as it is driven by some of the biggest cloud and service provider operators. For the past year, we have been working closely with members of the working group and in particular with...

- 40G TapAgg BiDi Design *June 18, 2018*
- Robot Framework for Auto Test *June 14, 2018*
- Understanding Logging Levels *June 6, 2018*
- ChalkTalk Series Launched! *June 6, 2018*
- ZTPv6 using DHCPv6 Relay Agent *June 2, 2018*

Categories

- Automation
- CloudVision Portal
- Community
- CVP
- CVX
- DANZ
- DevOps



Home

rmadsen edited this page on 10 Dec 2014 · 15 revisions

Arista's EOS SDK

The EOS SDK guide

- Introduction
- Quickstart

Using the SDK

- EOS's architecture
- Lifecycle of an agent
- SDK components
- Modules and Capabilities

Development

- Installing the SDK
- Building an agent
- Debugging tips

Resources

Thank You

www.arista.com