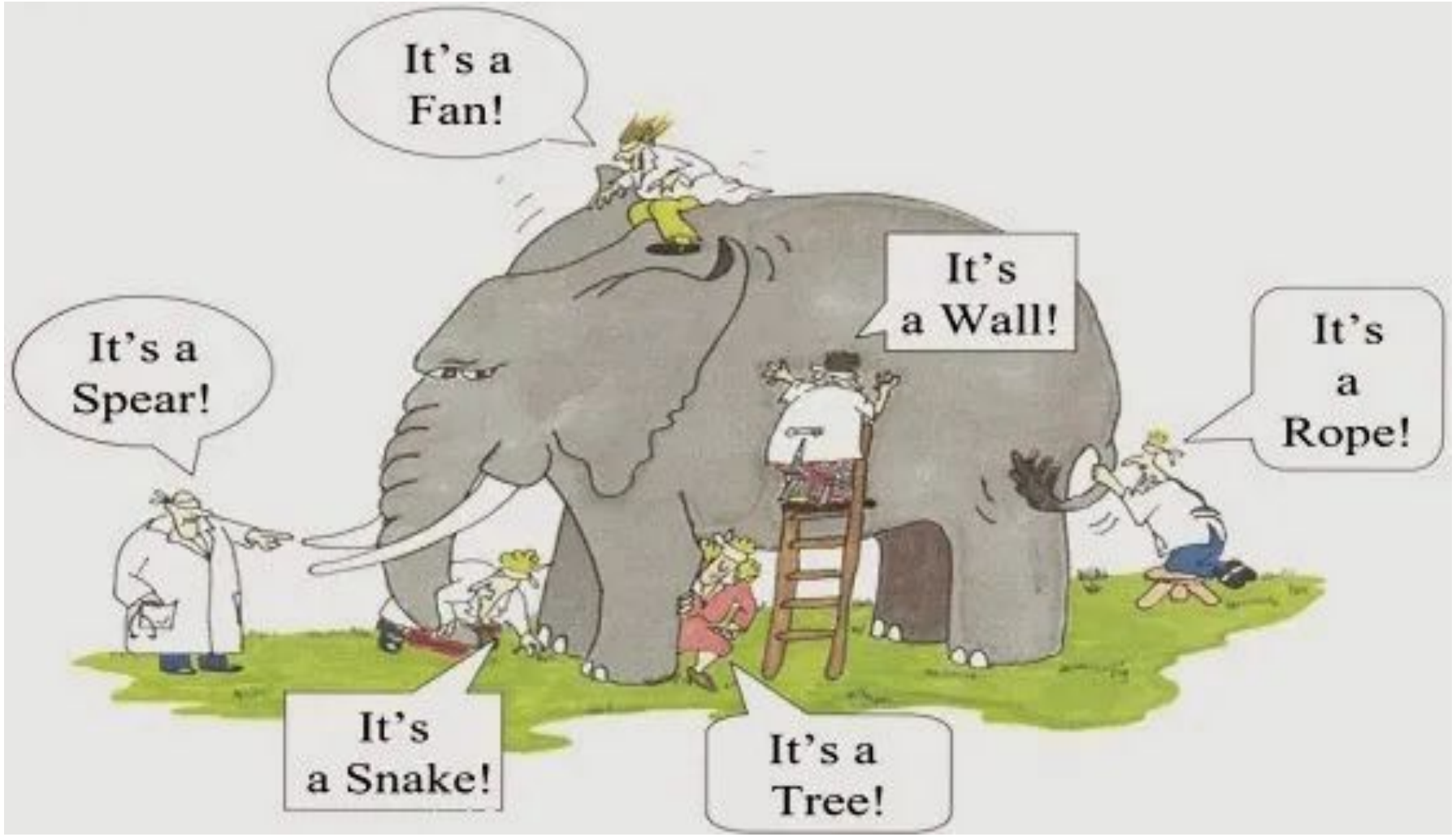
End-to-end troubleshooting in Data Centers

ARISTA

Agenda

- Background
- Why is Troubleshooting different today
- Troubleshooting tools for a modern DC
  - Built in features
- Building a universal monitoring fabric
- Summary

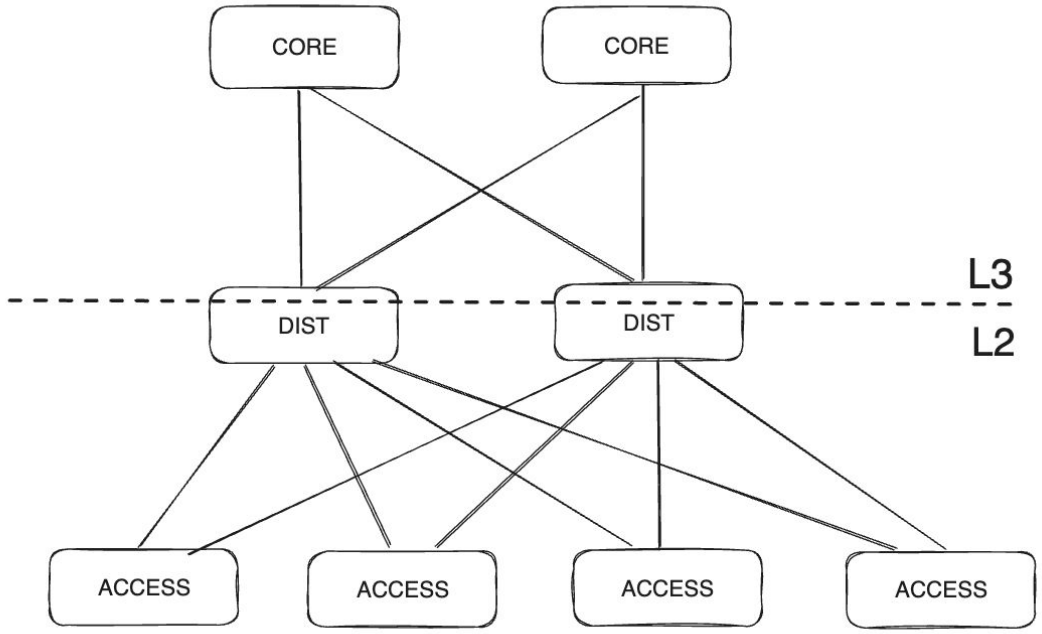
Why Network Visibility?



Why is troubleshooting different today?

15 years ago…Data Center Networks

- Applications
  - 3-tiered and fever dependencies
  - Applications were hosted in the same Data Center
- Traffic volumes
  - Relative low traffic volumes
  - North-south traffic
- Data Center Networks
  - Layer 2 based
  - 1/10 G server connections



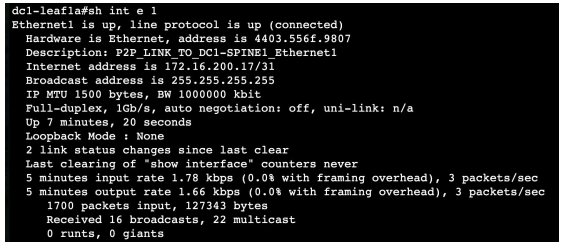
15 years ago…. Troubleshooting

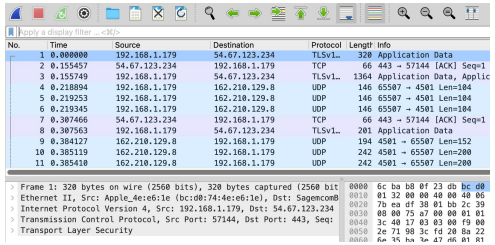
CLI PER DEVICE
Per device/interface CLI troubleshooting

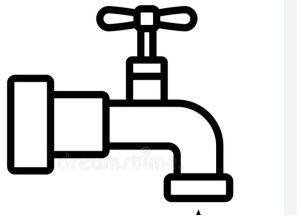
SPAN PORTS
From one switch to Wireshark

TAPS
North – south traffic

SNMP
Polling devices for state every 5 min



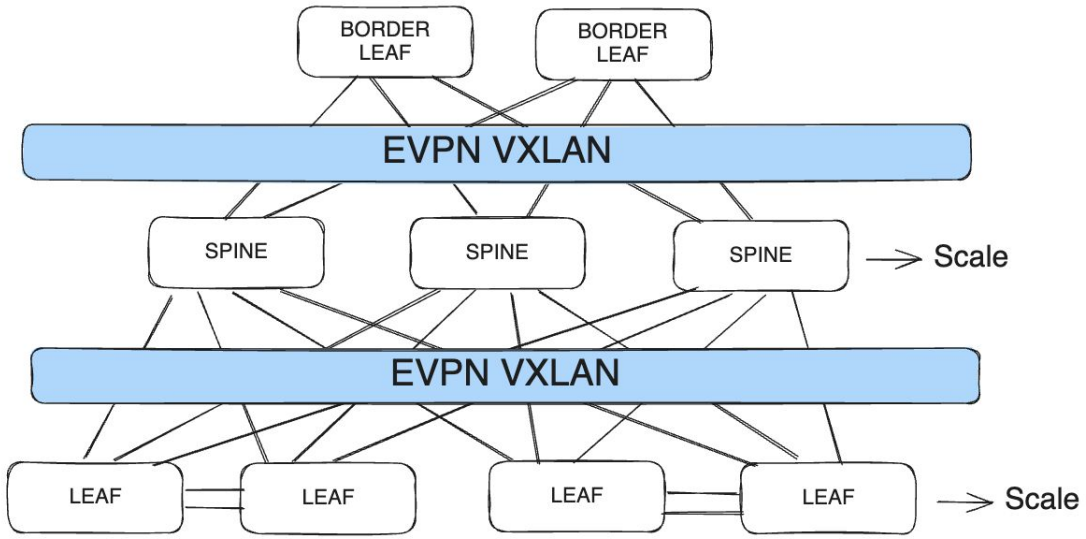




ARISTA

Data Centers of today….

- Distributed applications
  - Anyone anywere
  - No one knows ”communication paths” for an application
- Traffic volumes
  - East-West traffic
- Design
  - Leaf/Spine topology with multiple active paths Leaf switches
  - Multi-tenancy
  - VXLAN overlay
  - 100-200 Gbit/s server connections



Traffic is tunneled….

Troubleshooting of today

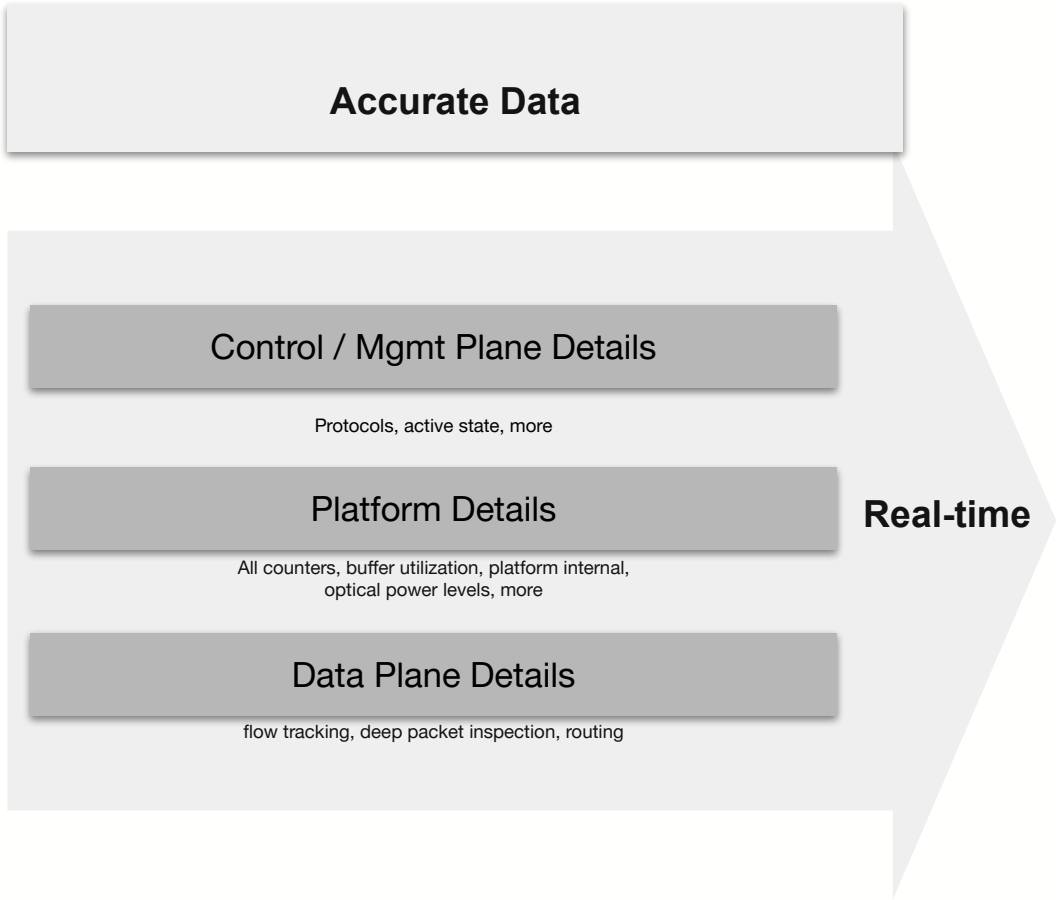
What to expect from the switches

Questions

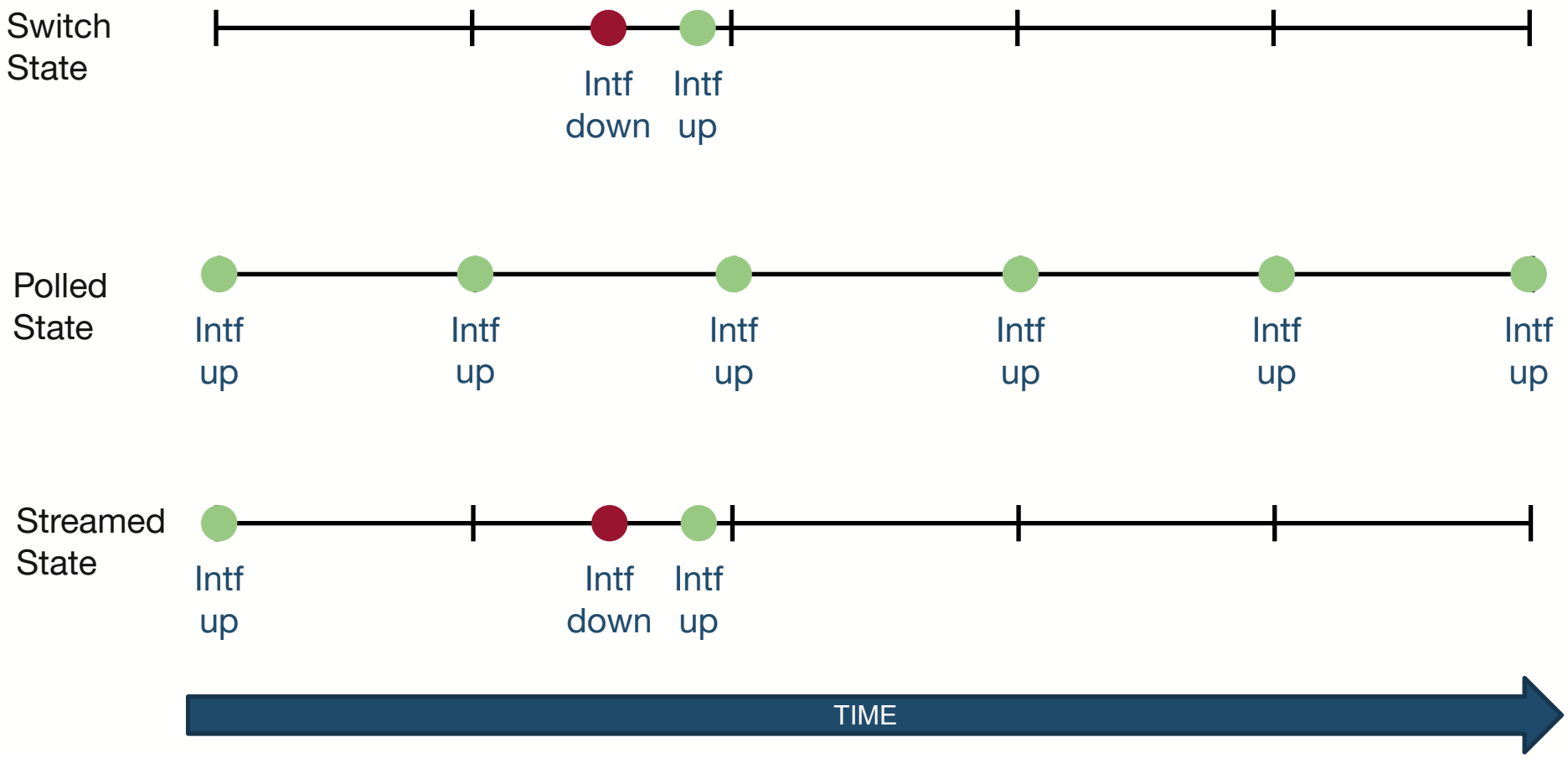
- What built-in features and tools are typically available in switches?
- How can my Network Monitoring System enhance troubleshooting efforts?
- Where should I capture network traffic to supply NPM, APM, and security tools?
- How can I develop a universal solution for troubleshooting?

1. Streaming Telemetry

- Streaming telemetry is a push mechanism
- Every events/state/counters continuously streamed from all devices
- Store data in a Time-series database
- Vendor specific Telemetry Paths & OpenConfig standardization



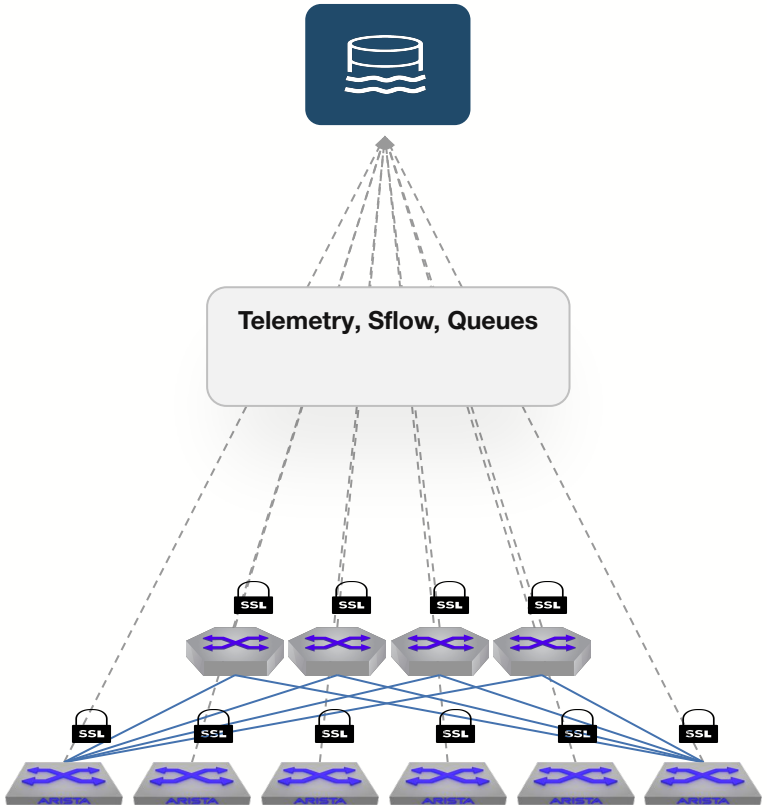
Polling Data == Missing Data



ARISTA

Benefits of Streaming Telemetry

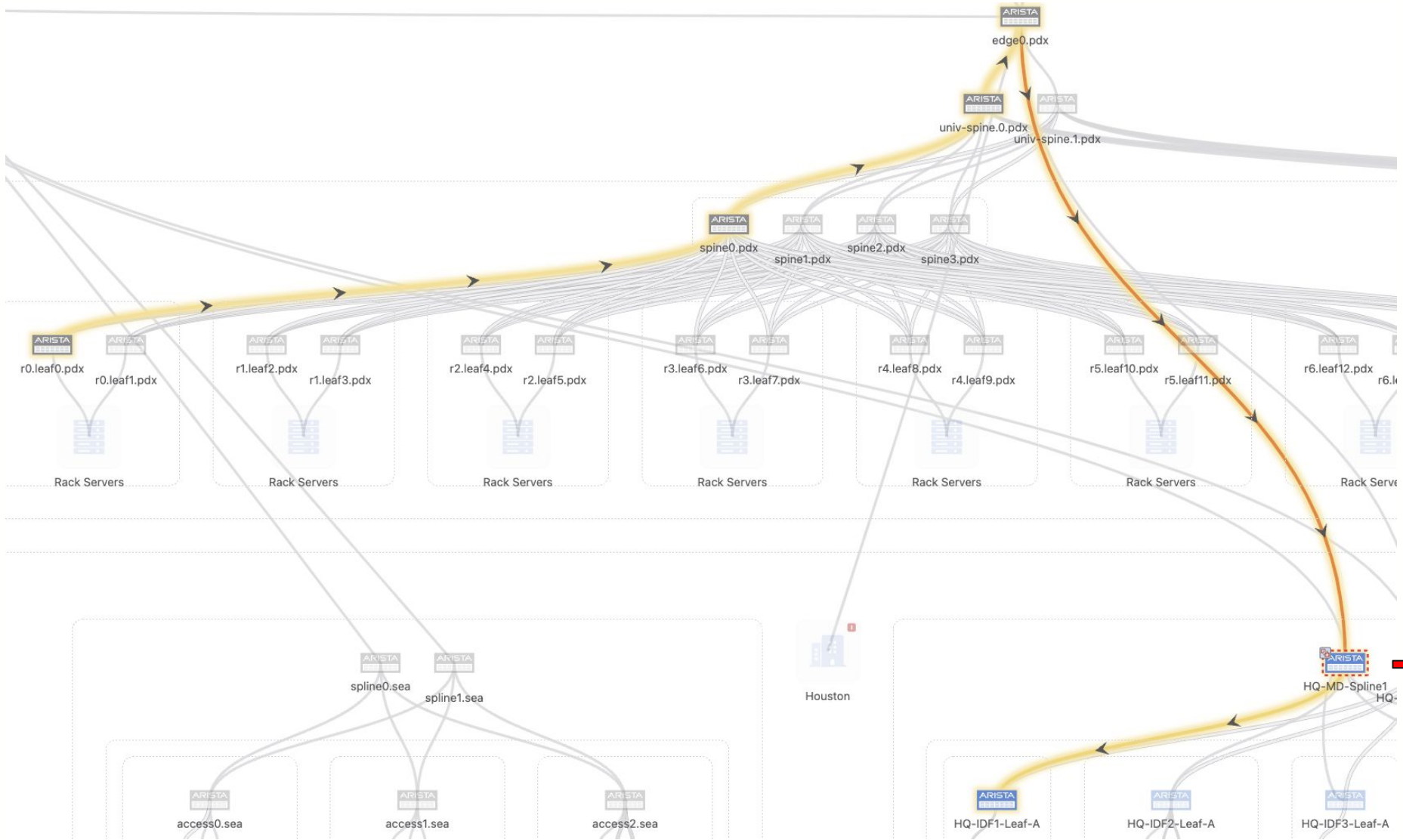
- Large amount of telemetry data streamed to a Data Lake
- Data in the Data Lake accessed by switch vendor Network Monitoring Systems and/or 3rd party applications
- Use of ML/AI will provide baselining, anomaly detection
- NetOps will have a real-time operational view of the network
- Dashboards and events based on real-time data



2. Inband Network Telemetry

- Visualize a flow through the network with per-hop latency
- Each switch insert their state onto the packet
- Provide answers to questions
  - Which path did my packet take
  - How long did it queue at each switch
  - Who did it share the queues with
  - Which node in the packet path did congestion originate
  - Congestion levels and drop counts

Inband Network Telemetry- visualization



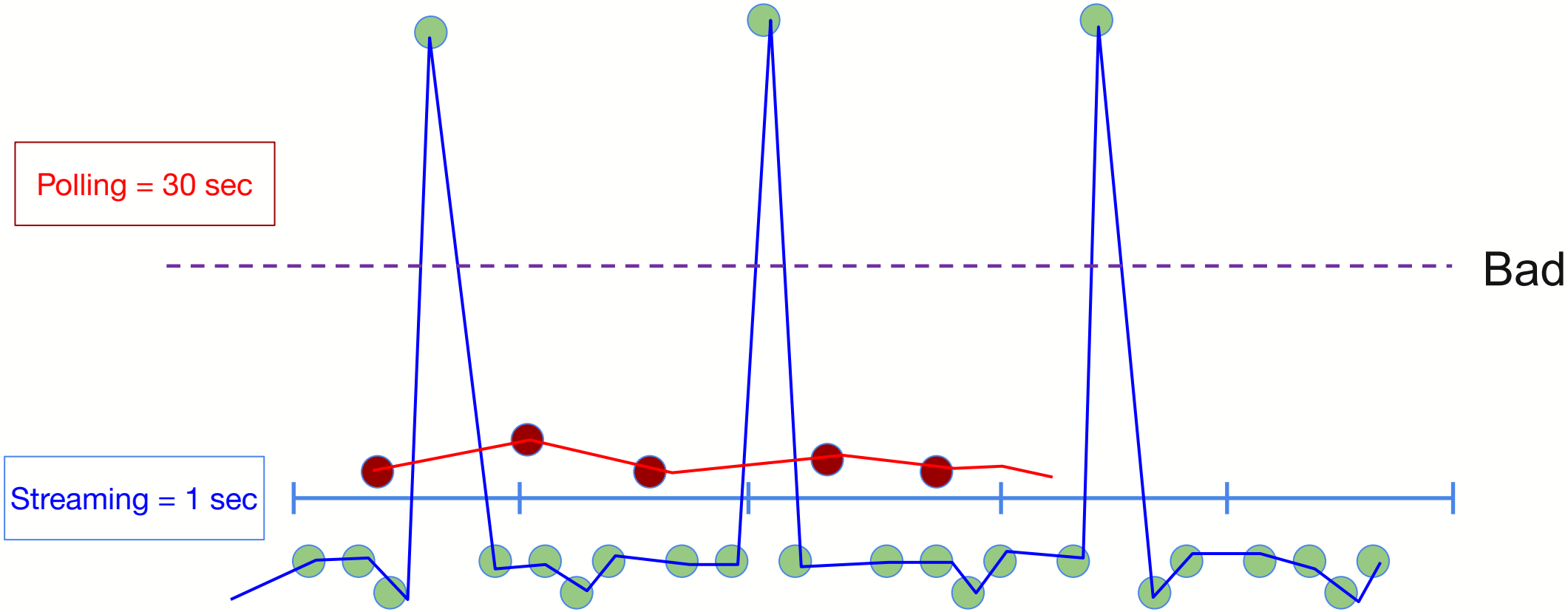
Path Recorded: 21:37:56.518

Bytes: 787.0 MB (47.06%) **Packets:** 27k

- r0.leaf0.pod0.pdx
  - Egress Interface: Ethernet32/0/1
  - Max Latency: 6,92 µs
  - Min Latency: 2,84 µs
  - Avg Latency: 3,2 µs
- spine0.pod0.pdx
  - Ingress Interface: Ethernet32/1/1
  - Egress Interface: Ethernet33/1/1
  - Max Latency: 3,3 µs
  - Min Latency: 365 ns
  - Avg Latency: 1,39 µs
- core0.pdx
  - Ingress Interface: Ethernet33/2/1
  - Egress Interface: Ethernet34/2/1
  - Max Latency: 5,35 µs
  - Min Latency: 105 ns
  - Avg Latency: 767 ns
- edge0.pdx
  - Ingress Interface: Ethernet34/3/1
  - Egress Interface: Ethernet35/3/1
  - Max Latency: 437,3 µs
  - Min Latency: 233,7 µs
  - Avg Latency: 151,6 µs
- HQ-MD-Spline1
  - (Detected a packet drop on this device for this path)
  - Ingress Interface: Ethernet35/4/1
  - Egress Interface: Ethernet36/4/1
  - Max Latency: 48,3 µs
  - Min Latency: 241,5 µs
  - Avg Latency: 136 µs

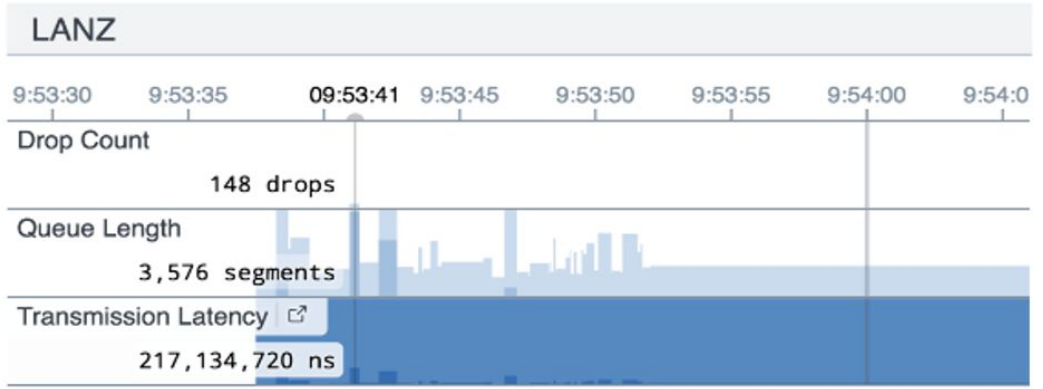
ARISTA

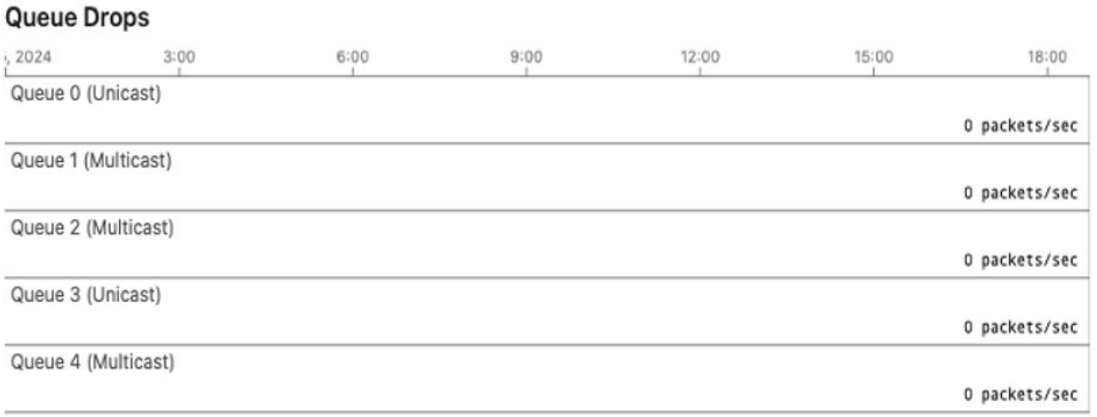
3. Microbursts …. to catch a performance killer



Latency Analyzer

- Real-time visibility of microbursts and the hardware buffers
- Monitors output queue lengths to provide congestion information for each individual interfaces
- threshold based reports for
  - latency added
  - packets dropped
  - queue depth on interface during congestion and micro congestion per interface





ARISTA

Hands down….. CLI is sometimes (*read always*) helpful

Some features handy when need to do debug and troubleshoot locally on a switch

Mirroring to CPU

- Local mirroring of traffic to CPU
- Analyzed locally without the need of a remote port analyzer
- Apply filters to the mirroring traffic

```
switch(config)#monitor session ingressSession source Ethernet 1 rx
switch(config)#monitor session ingressSession destination Cpu

switch(config)#monitor session egressSession source Ethernet 2 tx
switch(config)#monitor session egressSession destination Cpu

switch(config)#monitor session bothSession source Ethernet 3 both
switch(config)#monitor session bothSession destination Cpu
```

Mirror on drop

- allows monitoring of IP flow drops occurring in the ingress pipeline
- When drops are detected, it is sent to the control plane where it is processed and then sent to configured collectors

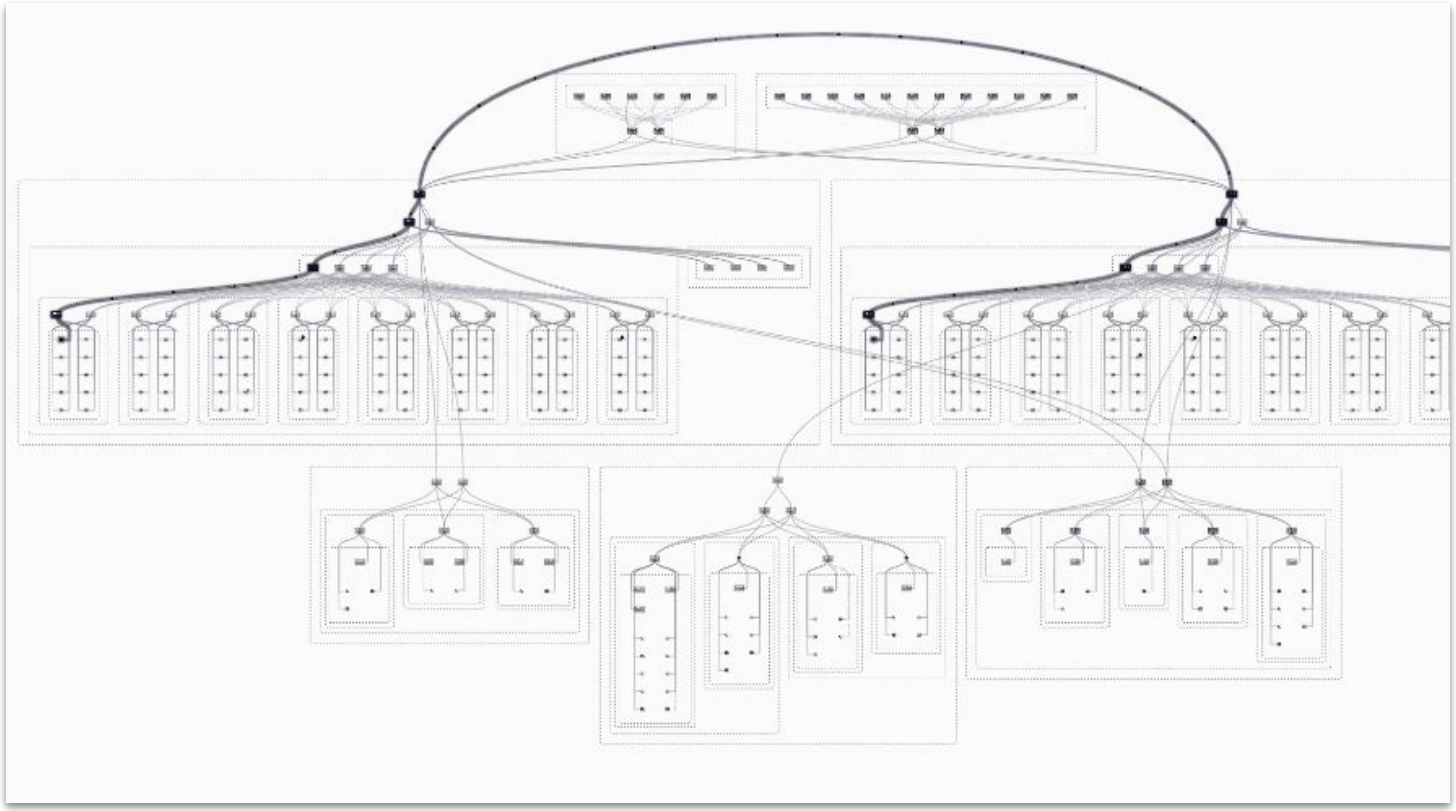
```
switch# show flow tracking mirror-on-drop
Flow Tracking Status
  Type: Mirror on drop
  Running: yes, enabled by the 'flow tracking mirror-on-drop' command
  Sample limit: 10
  Encapsulation: IPv4, IPv6
  Encapsulation filter: IPv4 uRPF, IPv6 uRPF
  Tracker: mod1
    Active interval: 300000 ms
    Inactive timeout: 15000 ms
    Groups:
    Exporter: exp1
      VRF: default
      Local interface: Ethernet3/1 (10.1.0.1, fc00::1)
      Export format: Sflow
      DSCP: 0
      Collectors:
        10.0.0.1 port 4739
        fc00::15 port 4739
```


ARISTA

Troubleshooting Host-to-Host

sFlow Visualization

- sFlow offers comprehensive, network-wide visibility
- Leverage L2-L7 end to end visibility from source to destination
- sFlow runs on all devices, without impacting dataplane forwarding
- sFlow utilizes statistical sampling to collect data efficiently
- sFlow is a cost-effective solution for network monitoring



sFlow

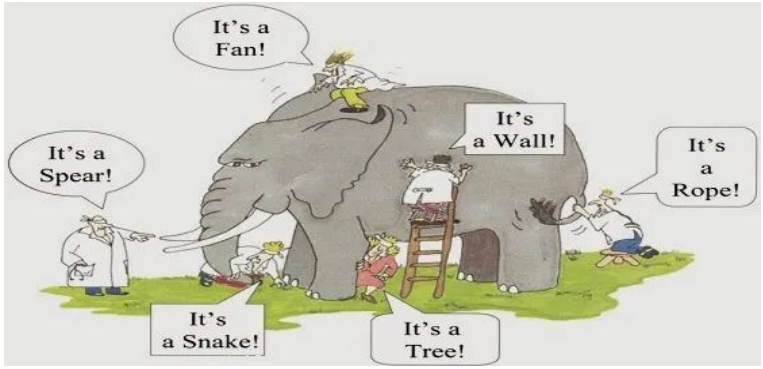
- The sFlow datagram, sent by the sFlow Agent, includes a significant amount of detailed data
  - Packet header (eg MAC,IPv4,IPv6,TCP,UDP,ICMP….)
  - Input/output ports
  - QoS
  - VLAN
  - Source/destination prefix
  - Next hop address
  - Source AS, Source Peer AS
  - ….. and many more

- Data is visualized in dashboards
- Drill down to each individual flow

ARISTA

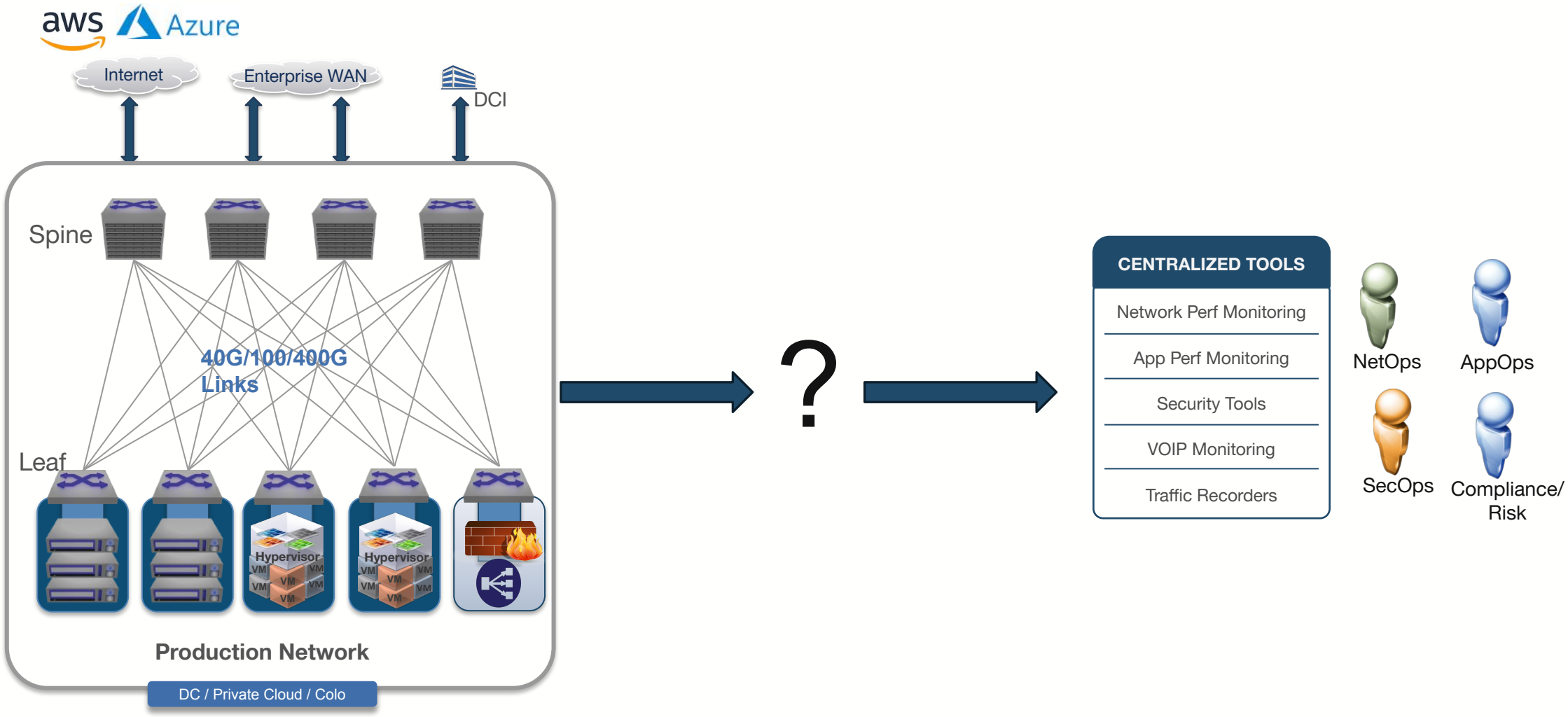
Building a Monitoring Fabric

…back to the six men and the elephant



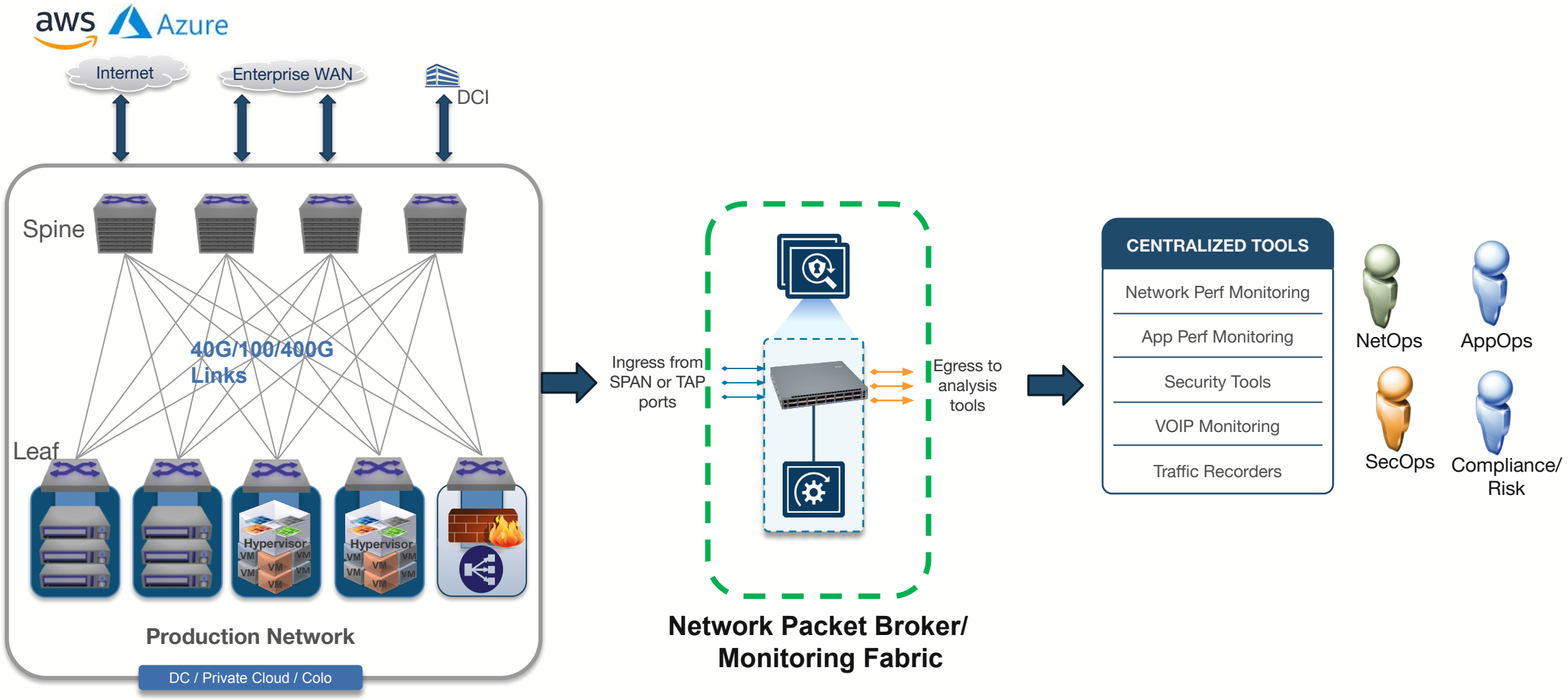
ARISTA

How to get the data to the tools



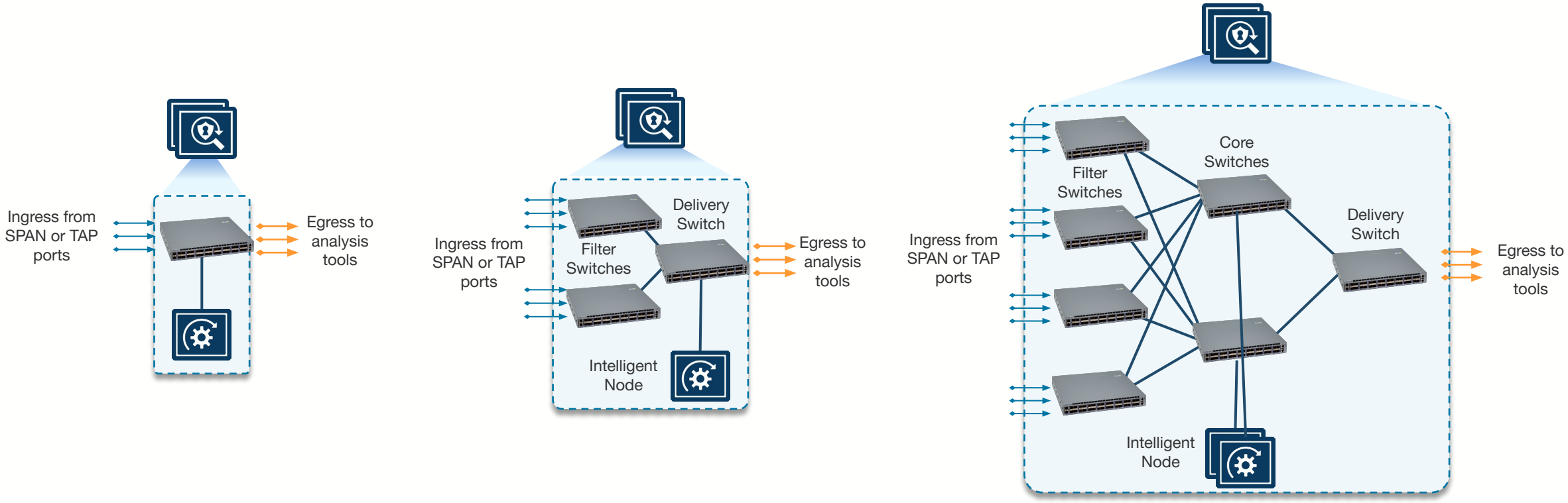
ARISTA

Monitoring Fabric/ Network Packet Broker



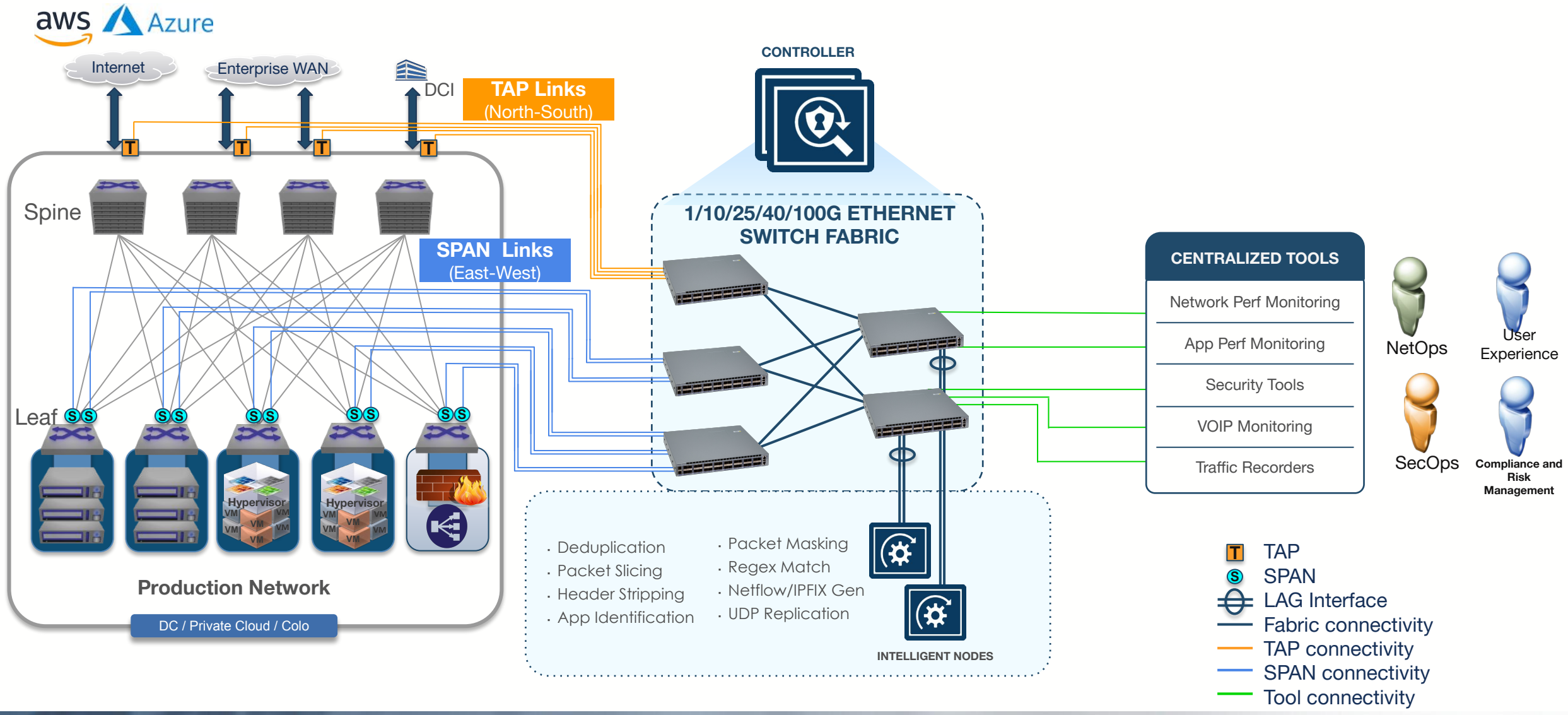
ARISTA

Scale Out the Packet Broker fabric



ARISTA

Data Center Monitoring Fabric



ARISTA

Summary

- Diagnosing application performance issues can be complex in modern large-scale data centers
- Utilize the built-in diagnostics tools provided by your switch vendor
- Allocate resources for tools that aid in effective troubleshooting
- Ensure network data is accessible to all relevant tools and teams
- Packets don’t lie—gaining access to packets are essential for pinpointing the root cause

THANK YOU

ARISTA