Open. Together.

OCP SUMMIT

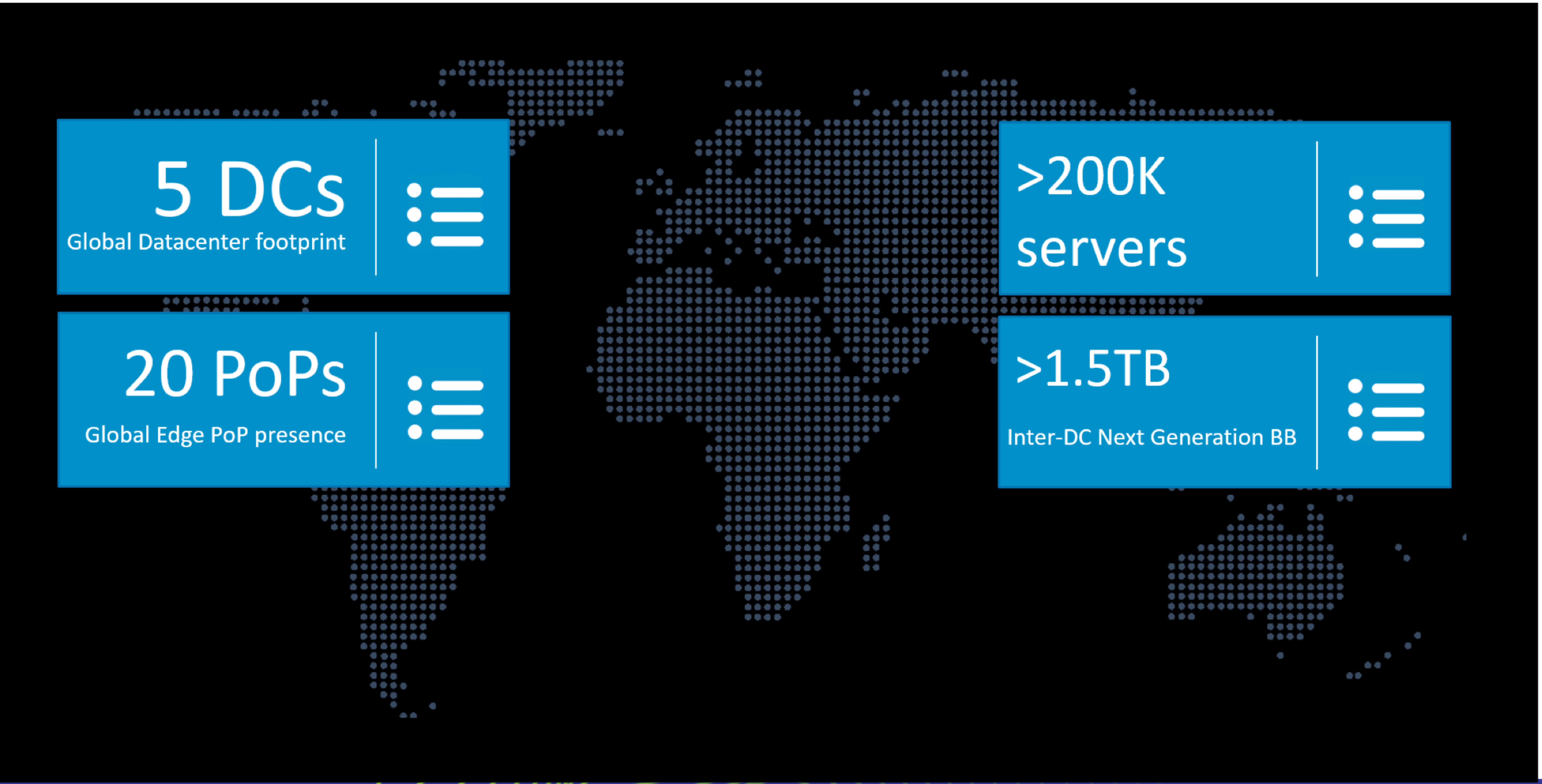
LinkedIn Adoption of OCP SONiC

Zhenggen Xu, Software Engineer, LinkedIn

Open. Together.

LinkedIn Infrastructure

NETWORKING



5 DCs
Global Datacenter footprint

20 PoPs
Global Edge PoP presence

>200K servers

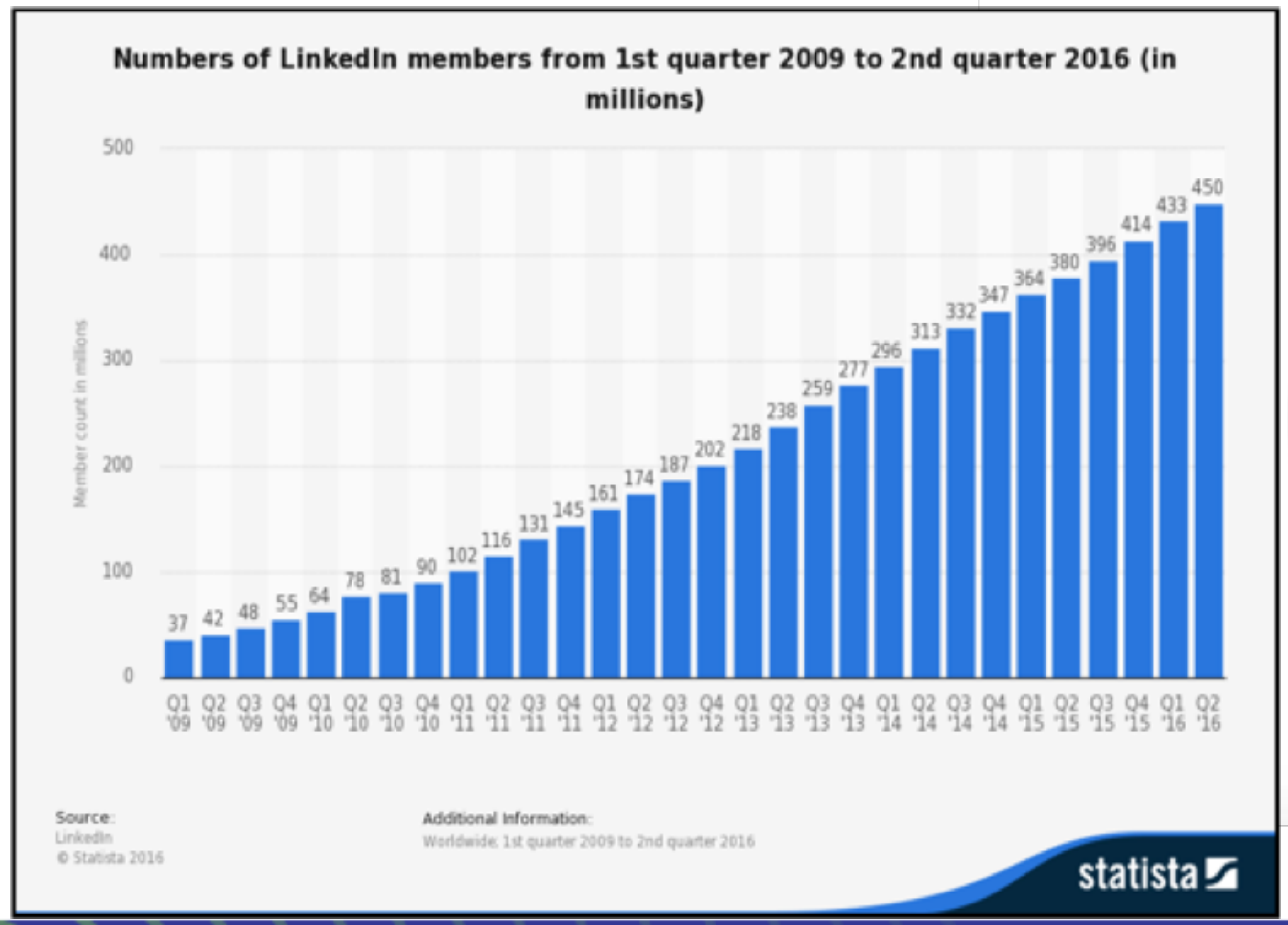
>1.5TB
Inter-DC Next Generation BB

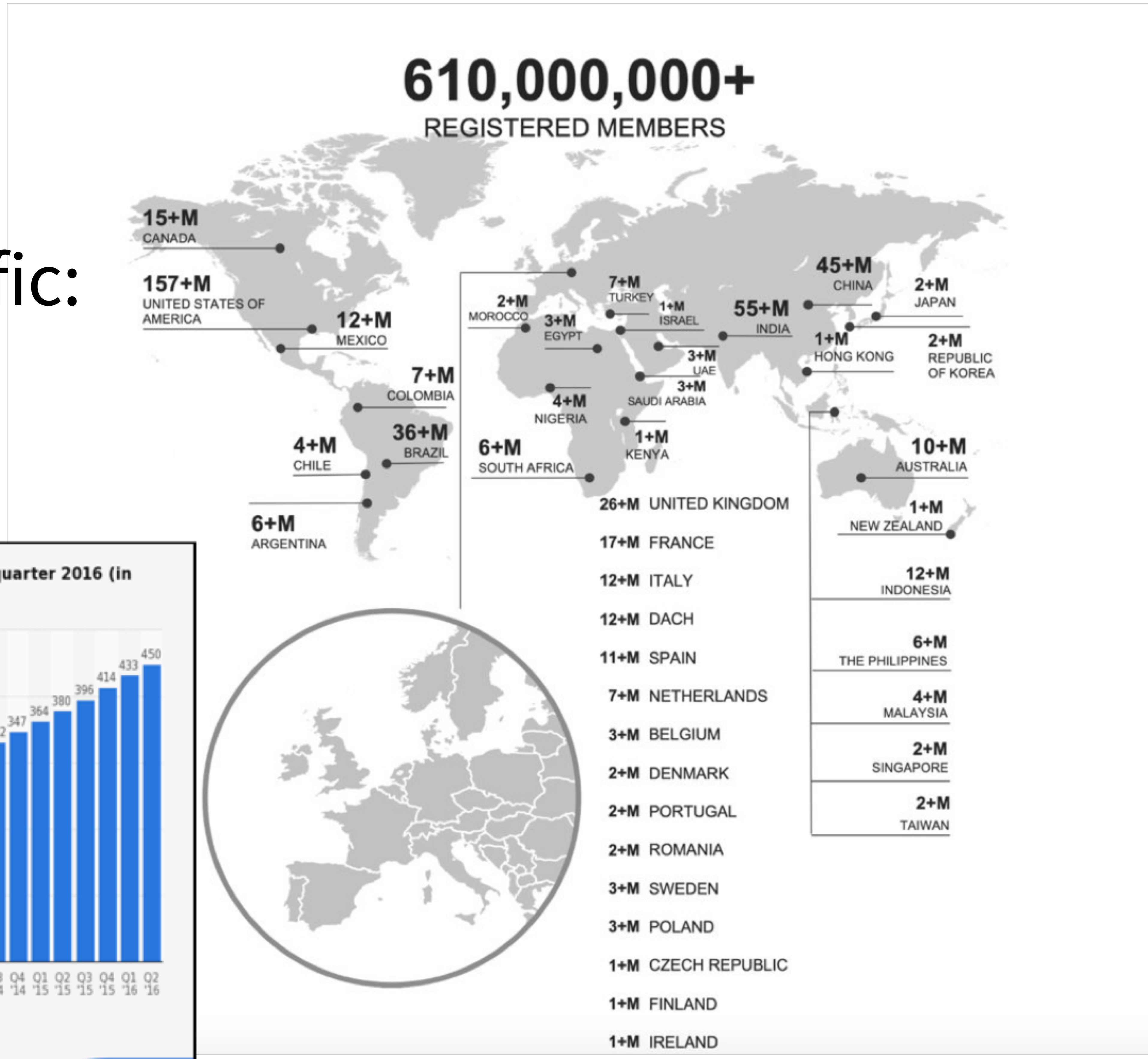
Infrastructure Growth

34% infrastructure growth every year…
High bandwidth demand due to the organic growth.

For every single byte, thousands bytes of east-west traffic:

- Application Call Graph
- Kafka Tracking
- Hadoop / Offline Processing
- Machine Learning





OCP SUMMIT

Open. Together.

Why Build Own NOS?

NETWORKING

Freedom and Choice

Flexibility
Customization
Modularity

Move Fast

Growth!
Scale
Evolve
Code & Innovate

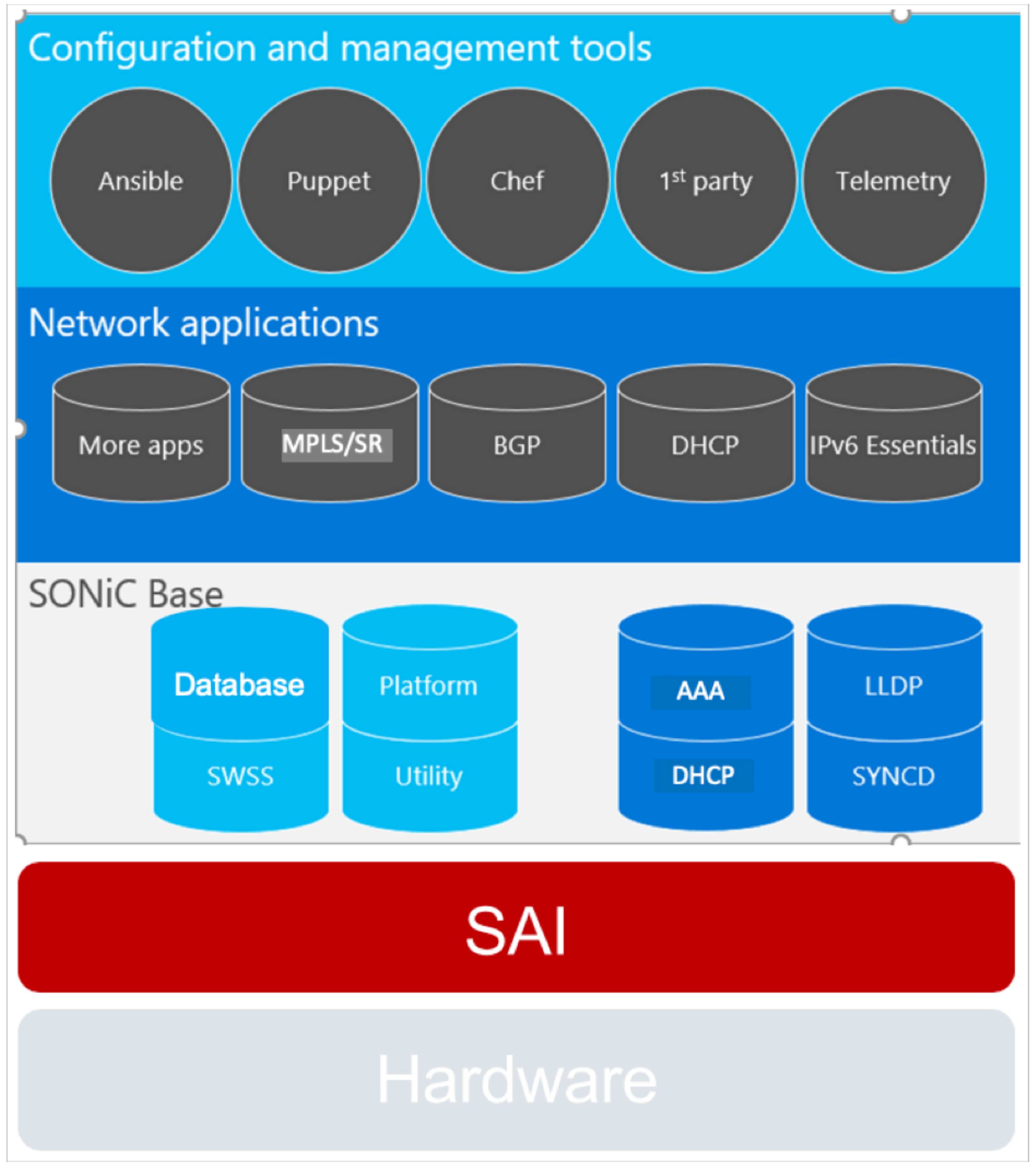
Independence

Channel
Procurement
Build Strategy
Ownership

Control

Quality
Maintenance
Risks
Security

Why SONiC?



NETWORKING

-- Linux based
Reuse networking stack, leverage same tools, experience.

-- Containerized
Pick the best components for the jobs at each layer.
Incremental upgrade

-- Platform agnostic
Switch Abstraction Interface (SAI)

-- Large community

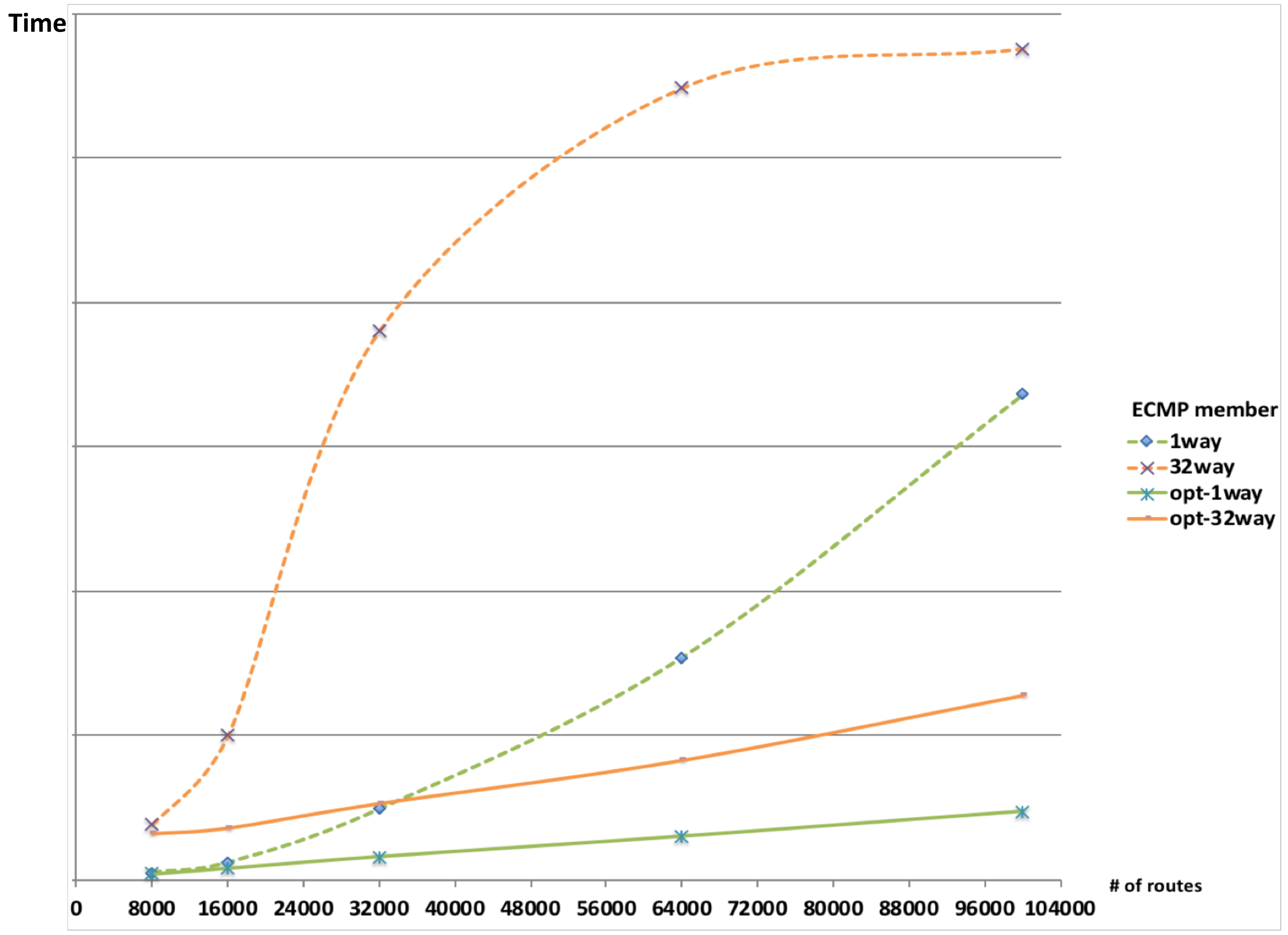
LinkedIn SONiC development

NETWORKING

- FRR protocol stack integration
- Fully IPv6 support – IPv6 ACL, IPv6 link local etc
- BGP convergence – FIB acceleration, SAI improvements.
- BGP docker warm restart
- SONiC system warm reboot and SWSS warm restart (collaboration effort)
- White-box onboarding
- AAA --- manage switches like servers
- ZTP integration
- Open19 onboarding
- Incremental upgrade of dockers and packages
- LinkedIn tools integration (telemetry and more)

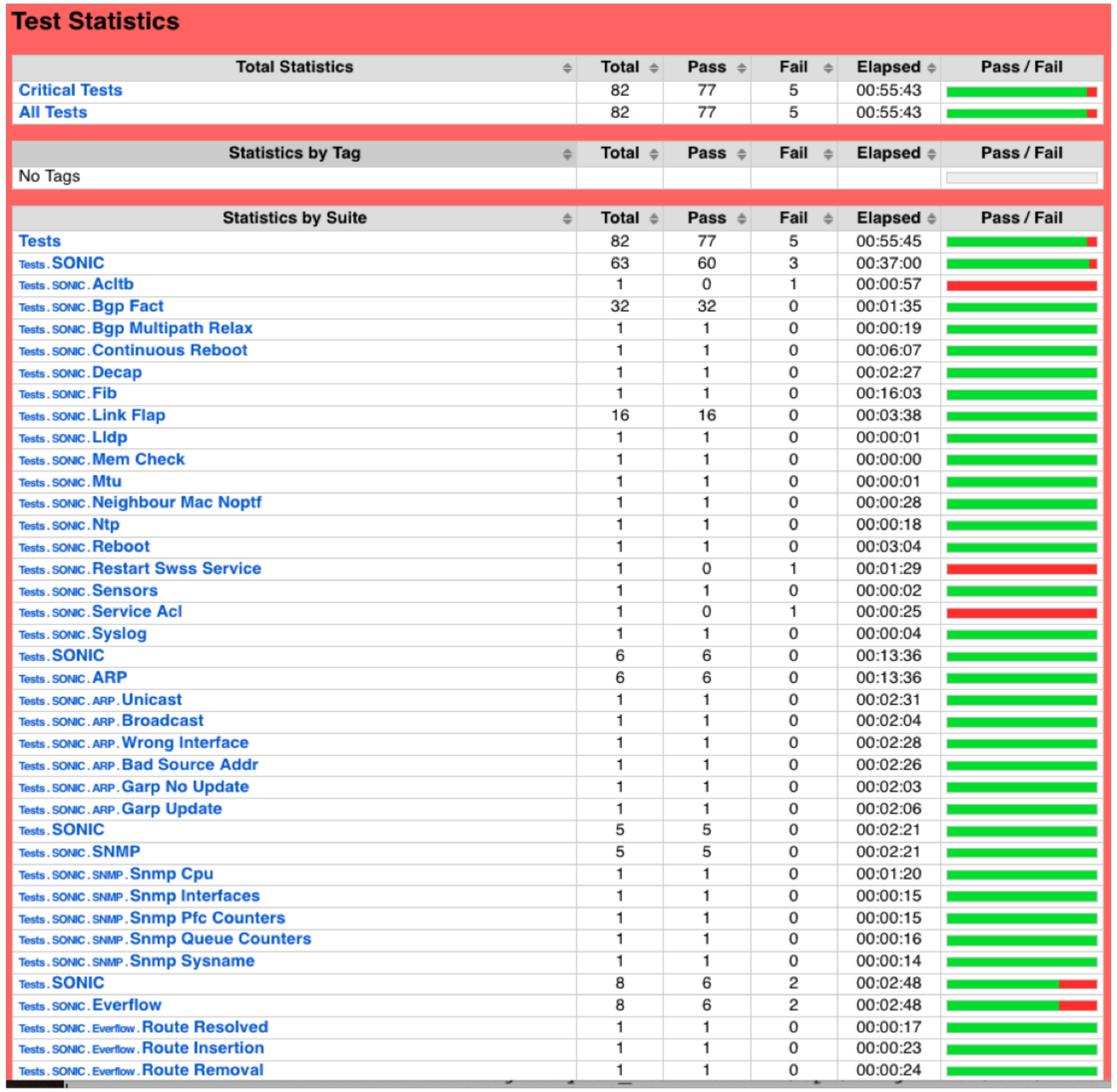
* Green ones are contributed back to the community

SONiC lessons: Routes installation convergence



NETWORKING

SONiC testing

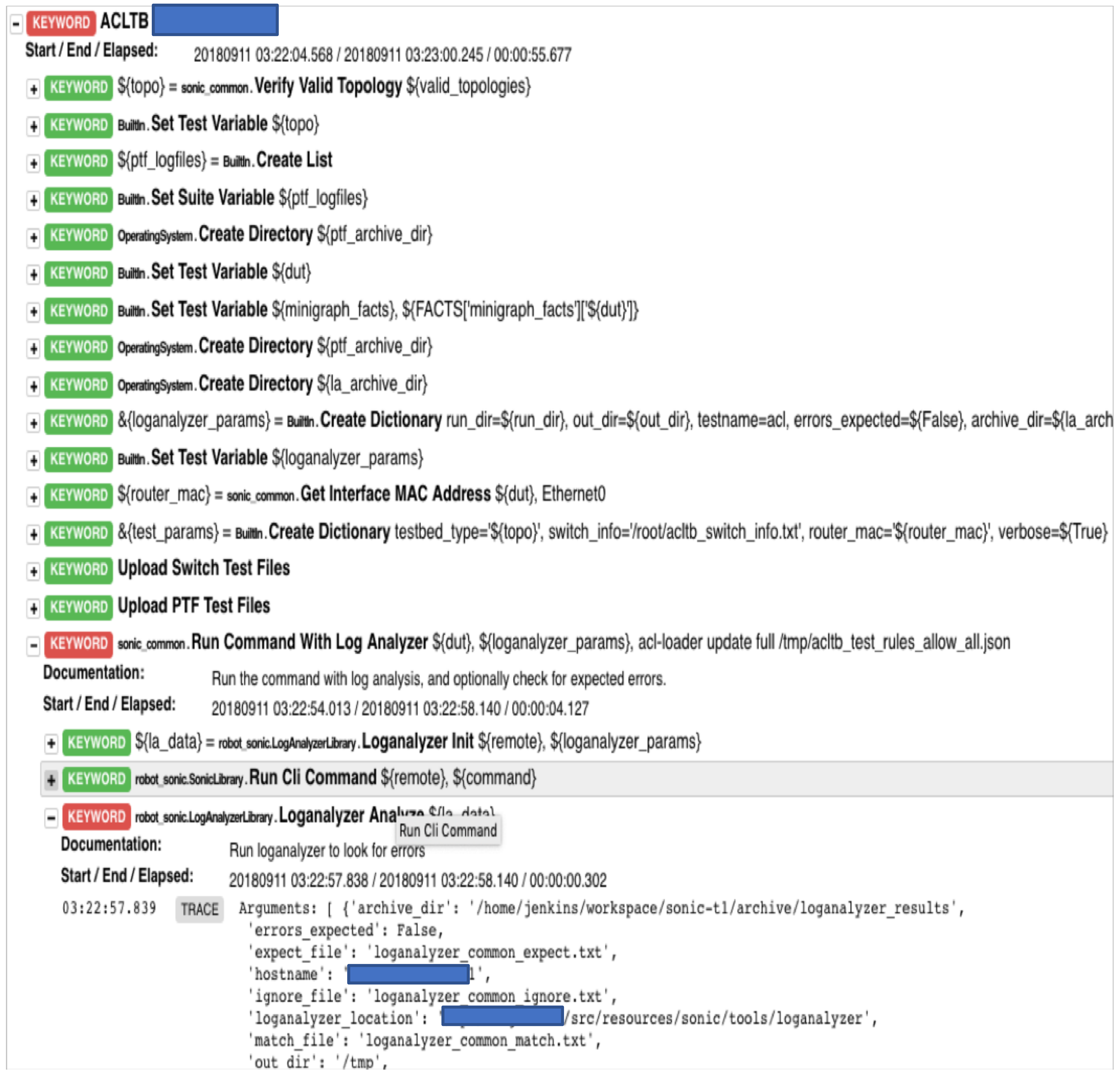


Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	82	77	5	00:55:43	🟩🟥
All Tests	82	77	5	00:55:43	🟩🟥

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

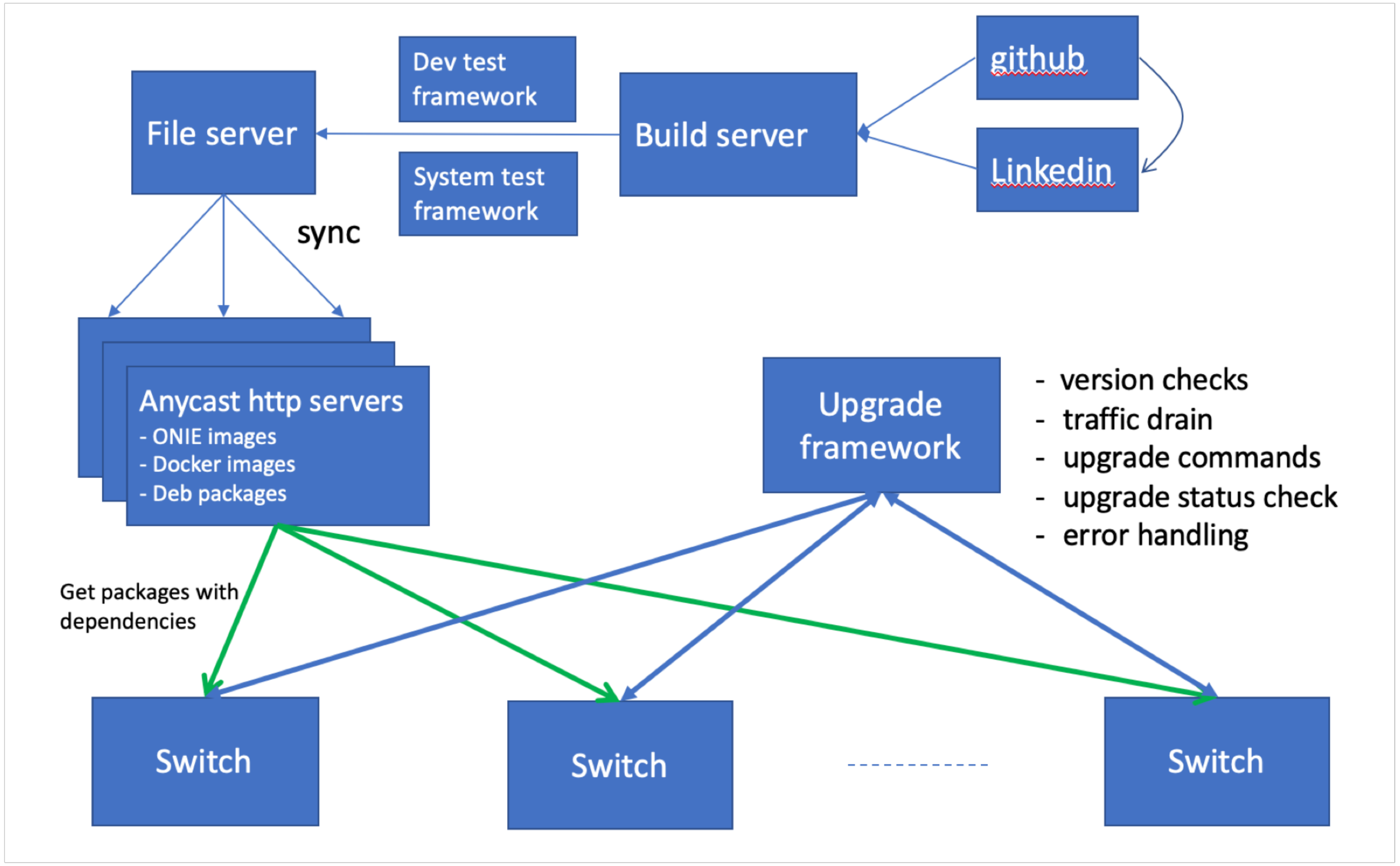
Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Tests	82	77	5	00:55:45	🟩🟥
Tests . SONIC	63	60	3	00:37:00	🟩🟥
Tests . SONIC . Acltb	1	0	1	00:00:57	🟥
Tests . SONIC . Bgp Fact	32	32	0	00:01:35	🟩
Tests . SONIC . Bgp Multipath Relax	1	1	0	00:00:19	🟩
Tests . SONIC . Continuous Reboot	1	1	0	00:06:07	🟩
Tests . SONIC . Decap	1	1	0	00:02:27	🟩
Tests . SONIC . Fib	1	1	0	00:16:03	🟩
Tests . SONIC . Link Flap	16	16	0	00:03:38	🟩
Tests . SONIC . Lldp	1	1	0	00:00:01	🟩
Tests . SONIC . Mem Check	1	1	0	00:00:00	🟩
Tests . SONIC . Mtu	1	1	0	00:00:01	🟩
Tests . SONIC . Neighbour Mac Noptf	1	1	0	00:00:28	🟩
Tests . SONIC . Ntp	1	1	0	00:00:18	🟩
Tests . SONIC . Reboot	1	1	0	00:03:04	🟩
Tests . SONIC . Restart Swss Service	1	0	1	00:01:29	🟥
Tests . SONIC . Sensors	1	1	0	00:00:02	🟩
Tests . SONIC . Service Acl	1	0	1	00:00:25	🟥
Tests . SONIC . Syslog	1	1	0	00:00:04	🟩
Tests . SONIC	6	6	0	00:13:36	🟩
Tests . SONIC . ARP	6	6	0	00:13:36	🟩
Tests . SONIC . ARP . Unicast	1	1	0	00:02:31	🟩
Tests . SONIC . ARP . Broadcast	1	1	0	00:02:04	🟩
Tests . SONIC . ARP . Wrong Interface	1	1	0	00:02:28	🟩
Tests . SONIC . ARP . Bad Source Addr	1	1	0	00:02:26	🟩
Tests . SONIC . ARP . Garp No Update	1	1	0	00:02:03	🟩
Tests . SONIC . ARP . Garp Update	1	1	0	00:02:06	🟩
Tests . SONIC	5	5	0	00:02:21	🟩
Tests . SONIC . SNMP	5	5	0	00:02:21	🟩
Tests . SONIC . SNMP . Snmp Cpu	1	1	0	00:01:20	🟩
Tests . SONIC . SNMP . Snmp Interfaces	1	1	0	00:00:15	🟩
Tests . SONIC . SNMP . Snmp Pfc Counters	1	1	0	00:00:15	🟩
Tests . SONIC . SNMP . Snmp Queue Counters	1	1	0	00:00:16	🟩
Tests . SONIC . SNMP . Snmp Sysname	1	1	0	00:00:14	🟩
Tests . SONIC	8	6	2	00:02:48	🟩🟥
Tests . SONIC . Everflow	8	6	2	00:02:48	🟩🟥
Tests . SONIC . Everflow . Route Resolved	1	1	0	00:00:17	🟩
Tests . SONIC . Everflow . Route Insertion	1	1	0	00:00:23	🟩
Tests . SONIC . Everflow . Route Removal	1	1	0	00:00:24	🟩



```
- KEYWORD ACLTB [redacted]
Start / End / Elapsed:    20180911 03:22:04.568 / 20180911 03:23:00.245 / 00:00:55.677
 + KEYWORD ${topo} = sonic_common.Verify Valid Topology ${valid_topologies}
 + KEYWORD BuiltIn.Set Test Variable ${topo}
 + KEYWORD ${ptf_logfiles} = BuiltIn.Create List
 + KEYWORD BuiltIn.Set Suite Variable ${ptf_logfiles}
 + KEYWORD OperatingSystem.Create Directory ${ptf_archive_dir}
 + KEYWORD BuiltIn.Set Test Variable ${dut}
 + KEYWORD BuiltIn.Set Test Variable ${minigraph_facts}, ${FACTS['minigraph_facts']['${dut}']}
 + KEYWORD OperatingSystem.Create Directory ${ptf_archive_dir}
 + KEYWORD OperatingSystem.Create Directory ${la_archive_dir}
 + KEYWORD &{loganalyzer_params} = BuiltIn.Create Dictionary run_dir=${run_dir}, out_dir=${out_dir}, testname=acl, errors_expected=${False}, archive_dir=${la_arch
 + KEYWORD BuiltIn.Set Test Variable ${loganalyzer_params}
 + KEYWORD ${router_mac} = sonic_common.Get Interface MAC Address ${dut}, Ethernet0
 + KEYWORD &{test_params} = BuiltIn.Create Dictionary testbed_type='${topo}', switch_info='/root/acltb_switch_info.txt', router_mac='${router_mac}', verbose=${True}
 + KEYWORD Upload Switch Test Files
 + KEYWORD Upload PTF Test Files
 - KEYWORD sonic_common.Run Command With Log Analyzer ${dut}, ${loganalyzer_params}, acl-loader update full /tmp/acltb_test_rules_allow_all.json
   Documentation:        Run the command with log analysis, and optionally check for expected errors.
   Start / End / Elapsed: 20180911 03:22:54.013 / 20180911 03:22:58.140 / 00:00:04.127
   + KEYWORD ${la_data} = robot_sonic.LogAnalyzerLibrary.Loganalyzer Init ${remote}, ${loganalyzer_params}
   + KEYWORD robot_sonic.SonicLibrary.Run Cli Command ${remote}, ${command}
   - KEYWORD robot_sonic.LogAnalyzerLibrary.Loganalyzer Analyze ${la_data}
     Documentation:        Run loganalyzer to look for errors
     Start / End / Elapsed: 20180911 03:22:57.838 / 20180911 03:22:58.140 / 00:00:00.302
     03:22:57.839  TRACE  Arguments: [ {'archive_dir': '/home/jenkins/workspace/sonic-t1/archive/loganalyzer_results',
                                      'errors_expected': False,
                                      'expect_file': 'loganalyzer_common_expect.txt',
                                      'hostname': '[redacted]1',
                                      'ignore_file': 'loganalyzer_common_ignore.txt',
                                      'loganalyzer_location': '[redacted]/src/resources/sonic/tools/loganalyzer',
                                      'match_file': 'loganalyzer_common_match.txt',
                                      'out_dir': '/tmp',
```

NETWORKING

SONiC upgrade at scale



- version checks
- traffic drain
- upgrade commands
- upgrade status check
- error handling

NETWORKING

SONiC monitoring: Collector, Data-Store and More

NETWORKING

Collector cluster collects all telemetry data through gRPC

- Polling
- Streaming
- Dial in
- Dial out

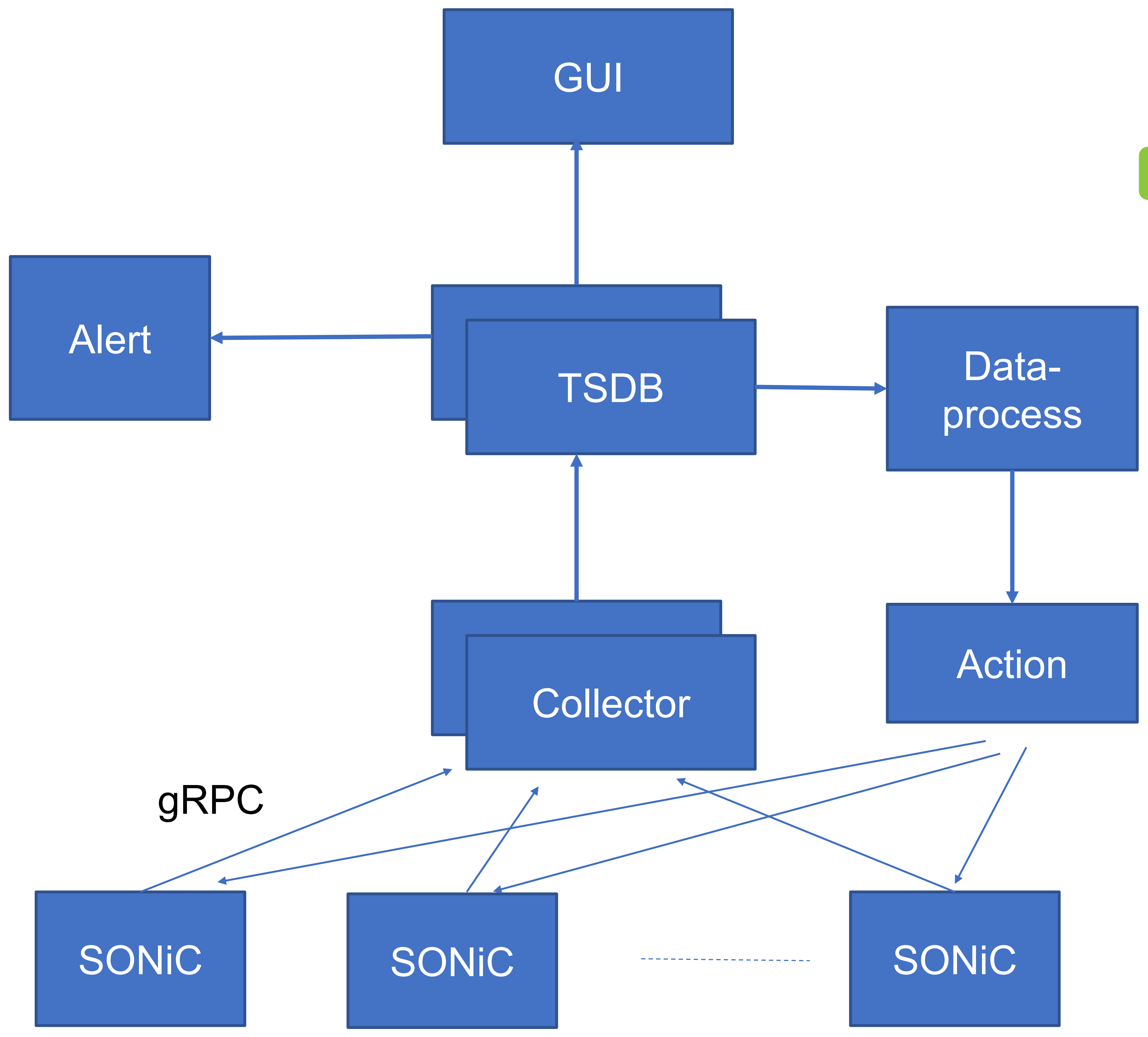
Data is saved to time-series DB.

- Schema-less
- Preserve history

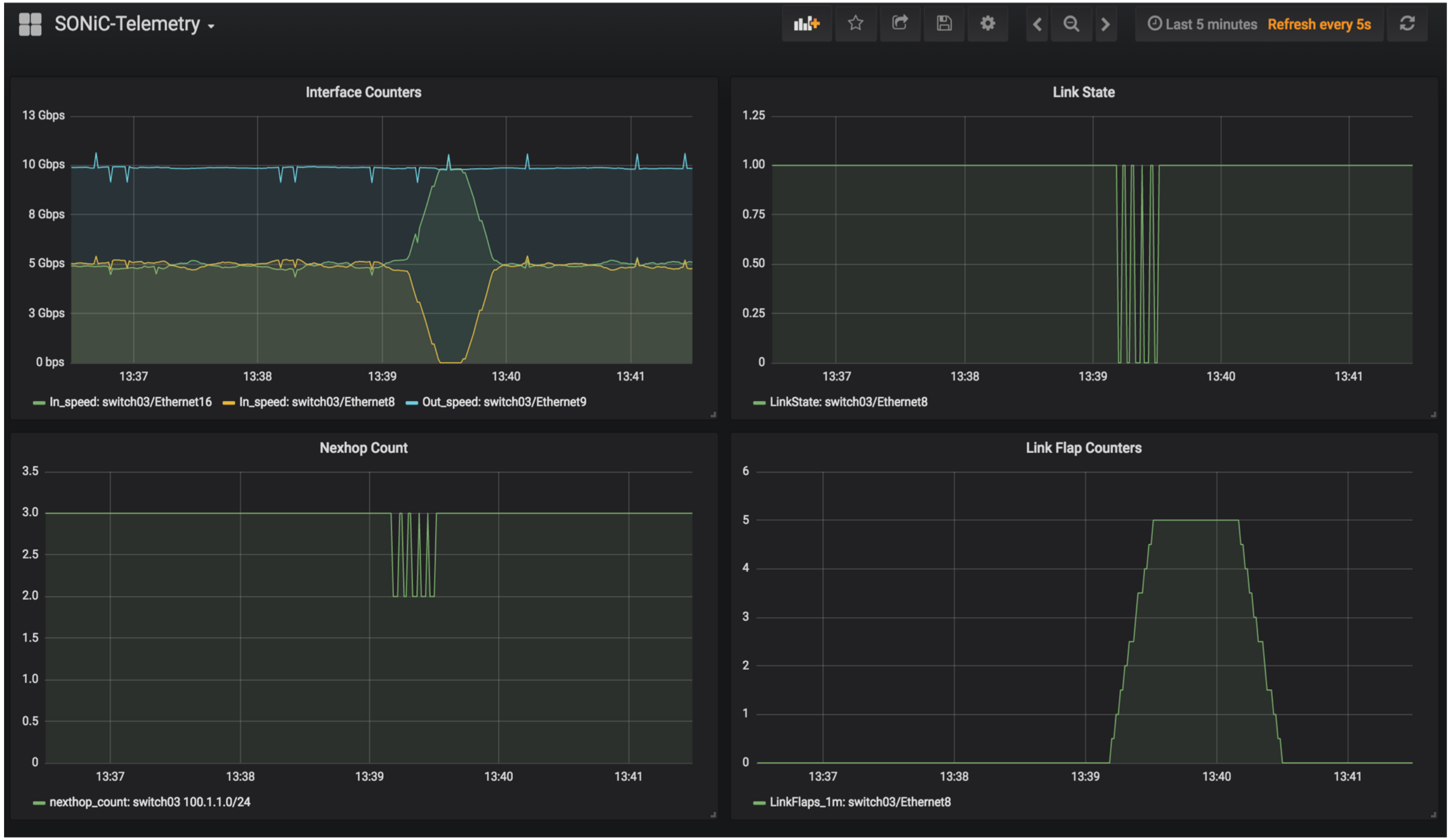
Data displayed on GUI

Rules defined to send events to alert system.

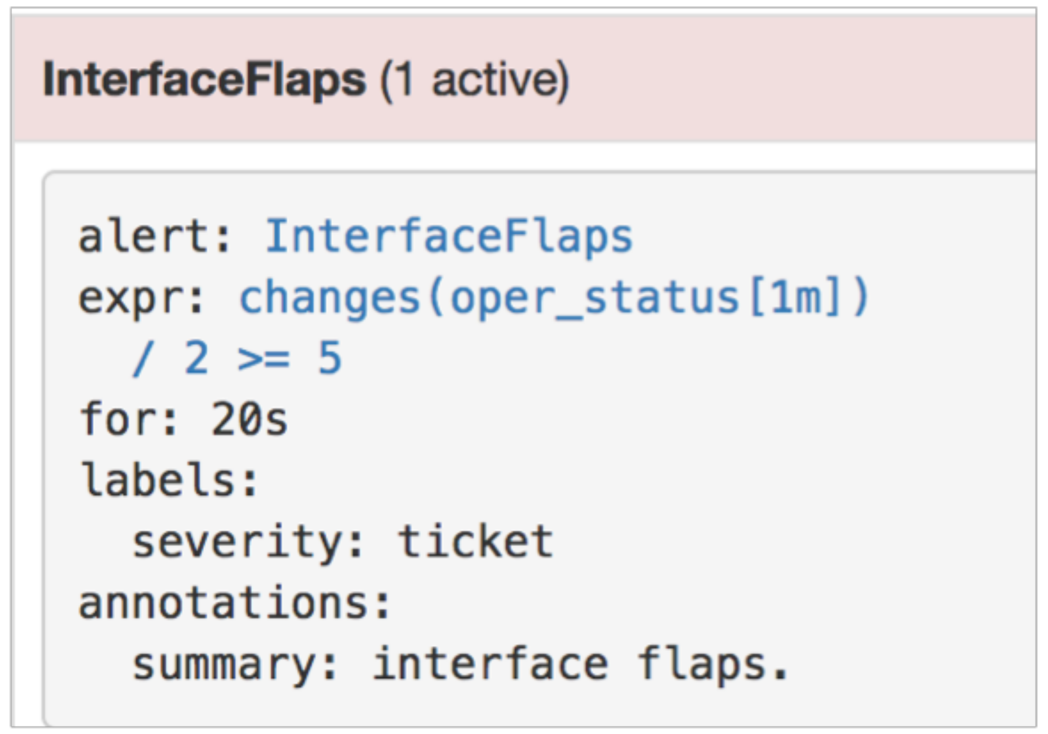
Data could be used for data-processing (ML etc) and applying actions back to the devices.



SONiC telemetry demo snapshot

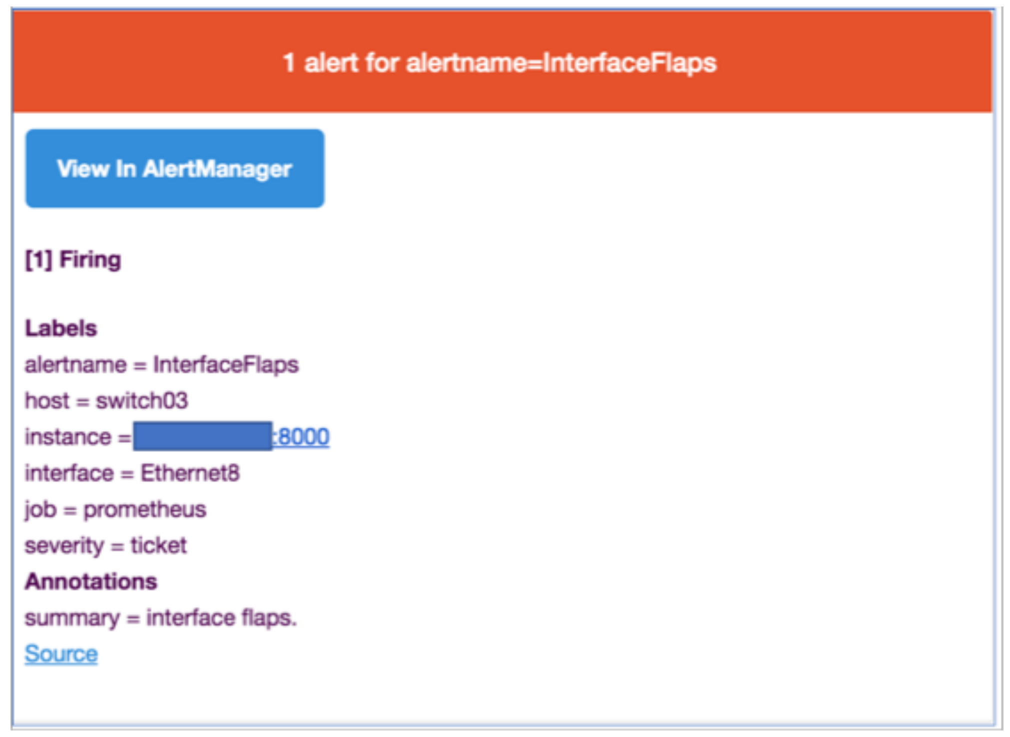


Rules:


InterfaceFlaps (1 active)

```
alert: InterfaceFlaps
expr: changes(oper_status[1m])
  / 2 >= 5
for: 20s
labels:
  severity: ticket
annotations:
  summary: interface flaps.
```

Email alerts received:


1 alert for alertname=InterfaceFlaps

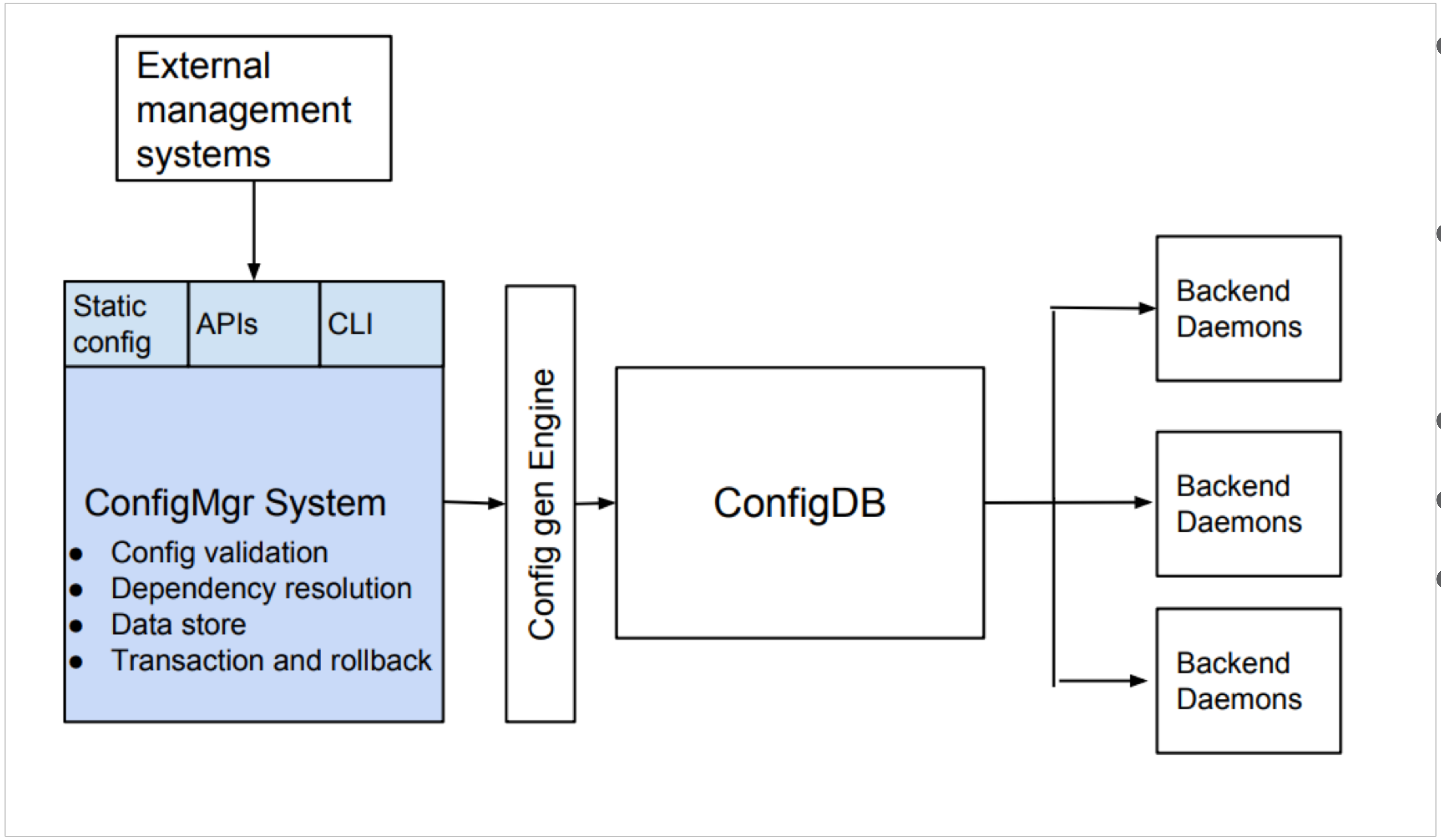
View In AlertManager

[1] Firing

Labels
alertname = InterfaceFlaps
host = switch03
instance = [redacted]:8000
interface = Ethernet8
job = prometheus
severity = ticket
Annotations
summary = interface flaps.
Source

SONiC feature WIP: configuration management

NETWORKING



- SONiC customized Yang based schema – no backend translation
- Syntax and dependencies validation
- Data store and rollback
- Plugin and play
- Programmable northbound interface, preferably gNMI

SONiC feature WIP: dynamic port breakout

NETWORKING

- Fully flexible to delete and add ports at run time
- Port breakout domain and naming can be defined and validated per platform
- Utilize the configuration management system for configuration integrity validation
- Remove and add port related dependencies automatically.
- Empower the flexibility for platforms like Open19 platform to the full extent

Call to Action

SONiC links:

Website: https://azure.github.io/SONiC/
Mailing list: sonicproject@googlegroups.com
Source code: https://github.com/Azure/SONiC/blob/gh-pages/sourcecode.md
Wiki: https://github.com/Azure/SONiC/wiki

Open. Together.

OCP Global Summit | March 14–15, 2019

OCP SUMMIT