WHO?

(What's Happening with OpenNMS)
UKNOF – Edinburgh, Scotland – 11 Sep 2018

Tarus Balog
tarus@opennms.org
https://adventuresinoss.com

I am not a vendor

"paradigm shifting"

"synergy building"

"market leader"

"k8s!"

"disruption"

"blockchain"

OpenNMS is the world's first carrier-grade network management application platform developed under the open source model.

world's first

• NetSaint	2000-01-10	1323
• OpenNMS	2000-03-30	4141
• Zabbix	2001-03-23	23494
• Nagios	2001-05-03	26589
• RRDTool	2003-01-13	71544
• Groundwork	2006-02-21	160654
• ZenOSS	2006-03-20	163126
• Hyperic	2006-07-17	172556

carrier-grade

OpenNMS was designed from Day One to monitor tens if not hundreds of thousands of devices. Current work is focused on removing those constraints to allow for millions of devices and billions of metrics.

That scalability comes in a number of forms:

- Discreet devices (hundreds of thousands)
- Performance metrics (billions)
- Events per second (tens of thousands)
- Remote monitors (thousands)

application platform

While OpenNMS works "out of the box", it really starts to shine when you customize it. It is highly configurable and offers a myriad of ways to integrate with other systems.

- Full-featured ReST Interface for both configuration and queries, forms the basis for OpenNMS Compass
- Device and event information stored in a database
- Notification system can execute arbitrary commands
- Built-in integration includes
  - RANCID configuration management
  - DNS for provisioning
  - Trouble Ticketing API (RT, Jira, OTRS, Remedy, etc.)

open source

Fully 100% of the OpenNMS source code is available under an Open Source license (as defined by the Open Source Initiative).

The main application is published under the AGPLv3, with various subsystems such as Newts published under more permissive licenses such as the Apache License.

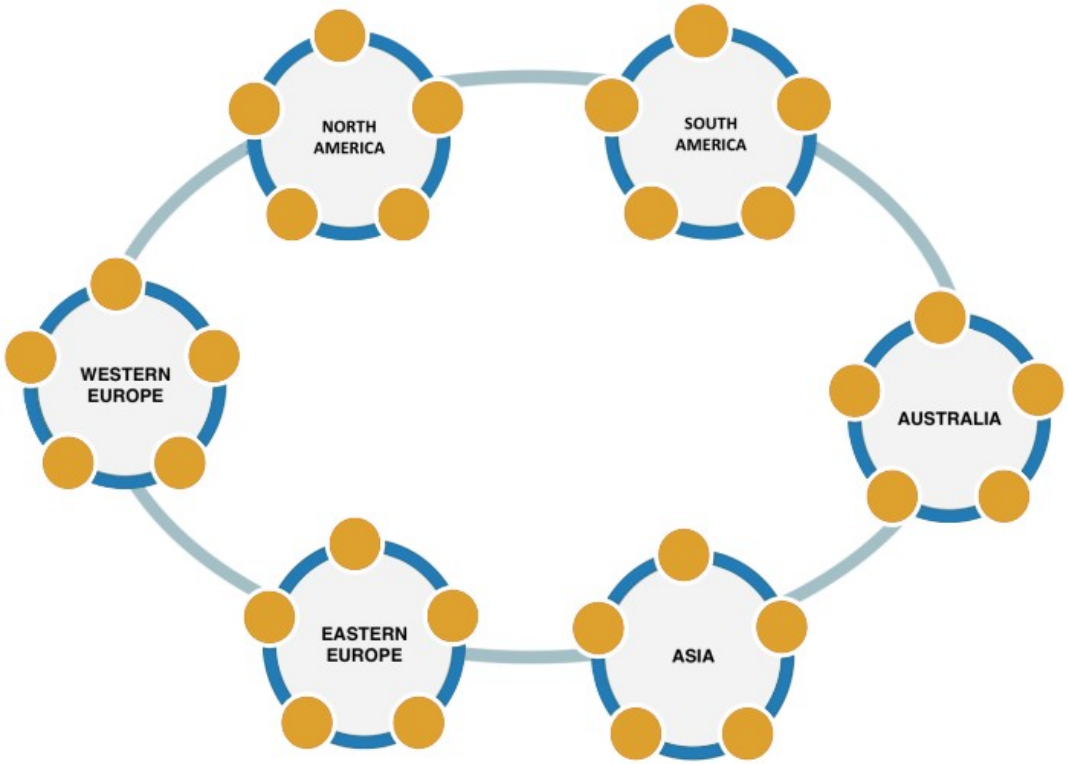
the road to scalability

- Newts
  New Time Series Database
- Minion
  Distributed Monitoring
- OpenNMS Drift (telemetryd)
  Collecting Flow Data

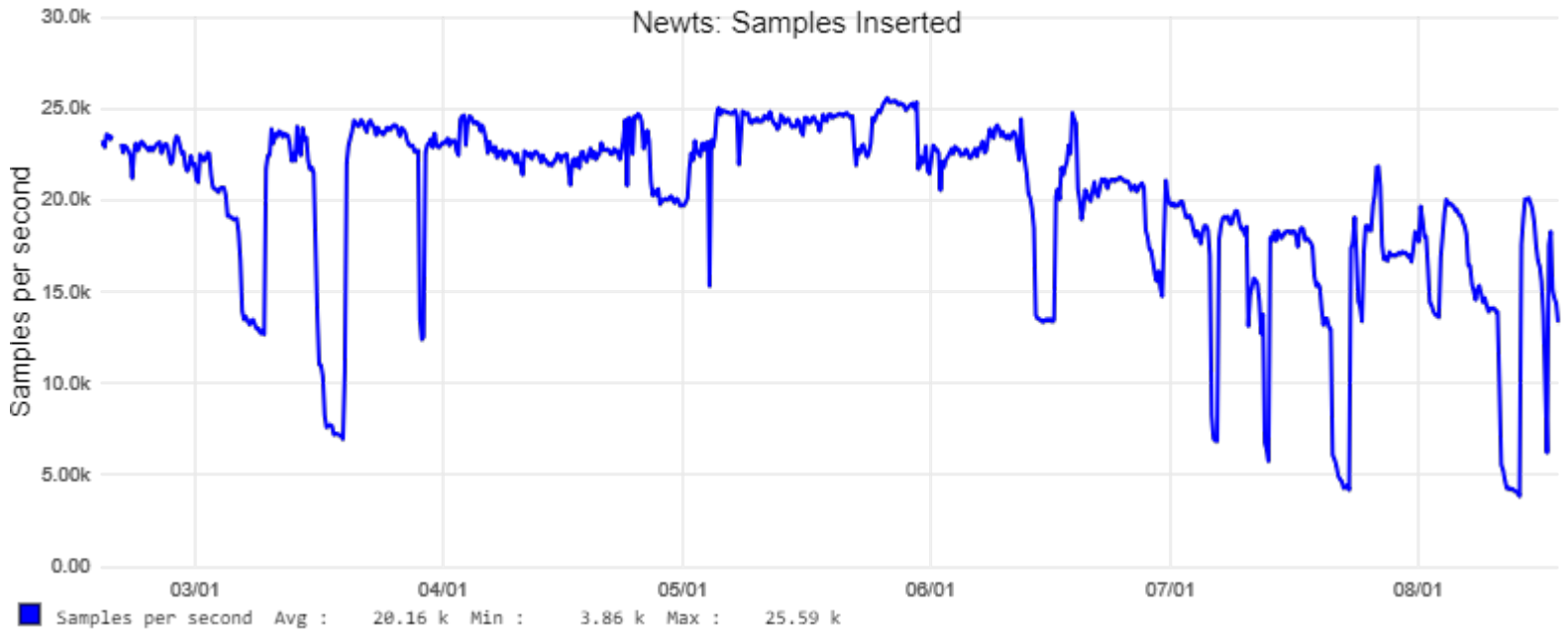
newts (new time series)

https://newts.io

apache cassandra

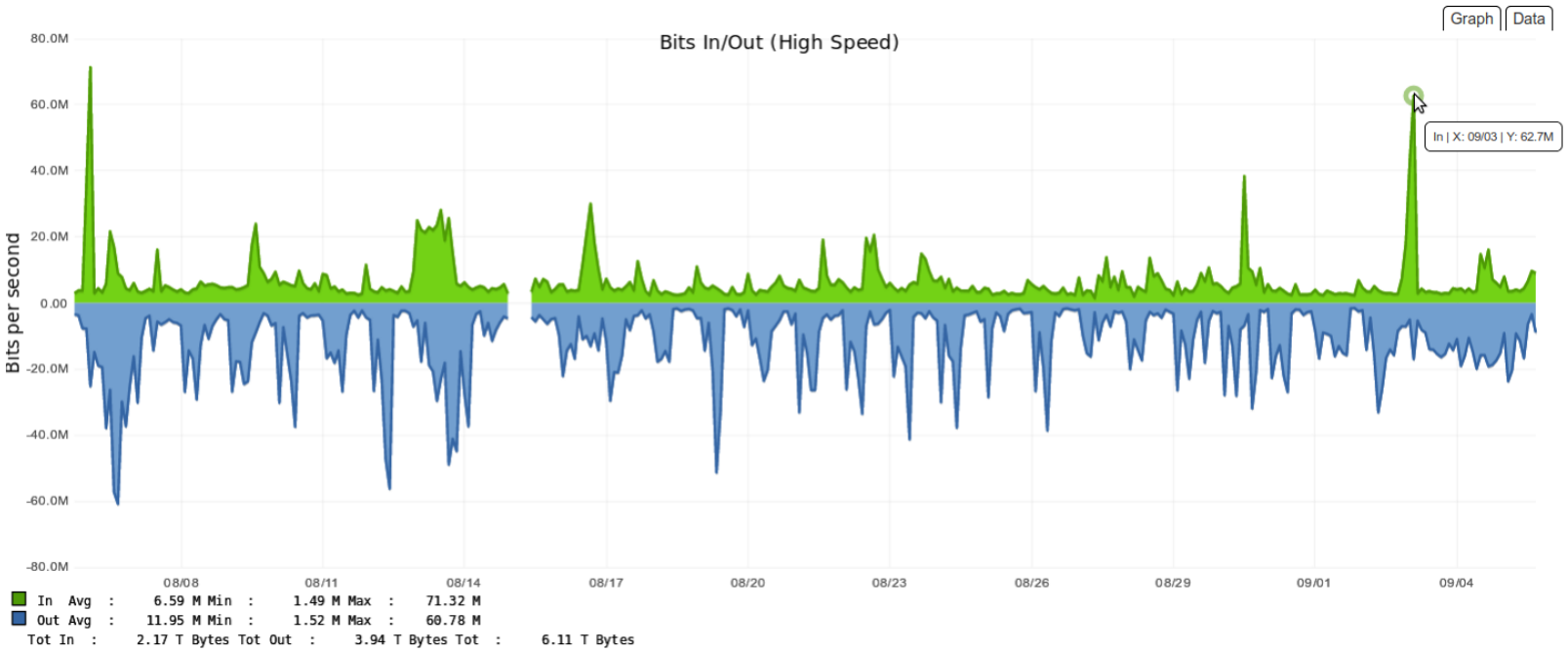


six month sample rate



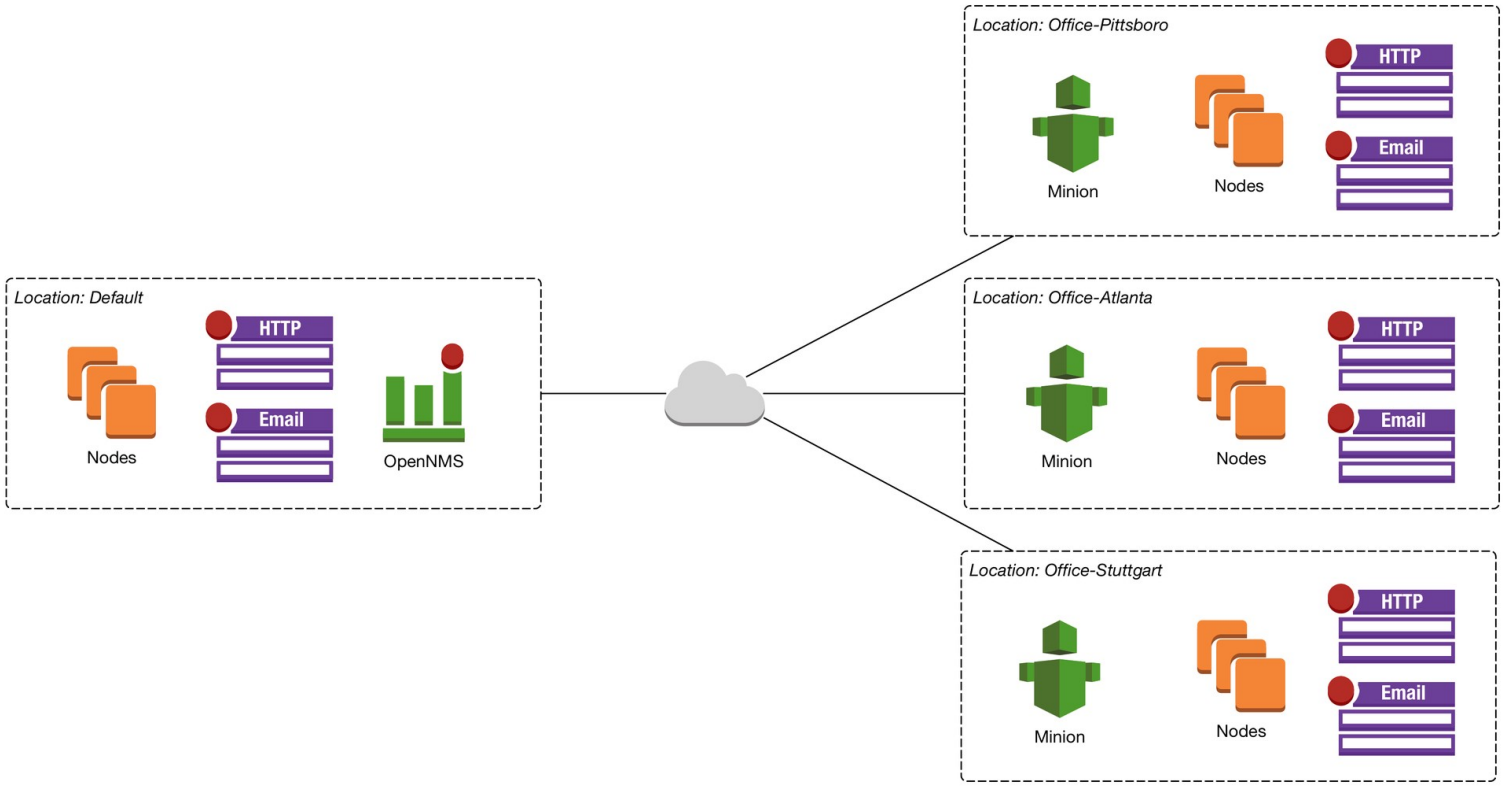
20,000 inserts/sec = 1.73B inserts/day

backshift



opennms minion

distributed functionality



minion setup

Each Minion Requires Five Attributes:

- USERNAME="minion"
- PASSWORD="minion"
- OPENNMS="http://opennms.example.com:8980"
- BROKER="tcp://opennms.example.com:61616"
- LOCATION="Edinburgh"

broker choices

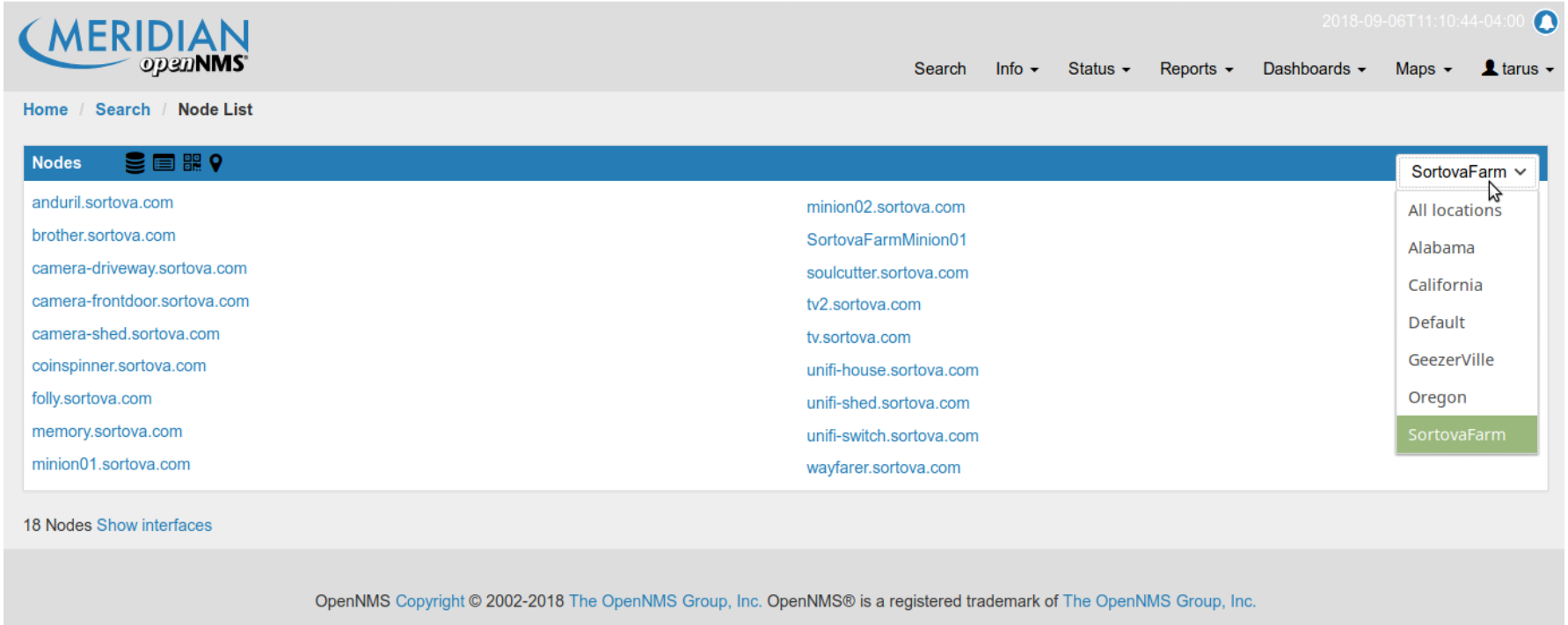
ActiveMQ

- Built-in
- Good for high latency situations
- Extra steps to secure

Kafka

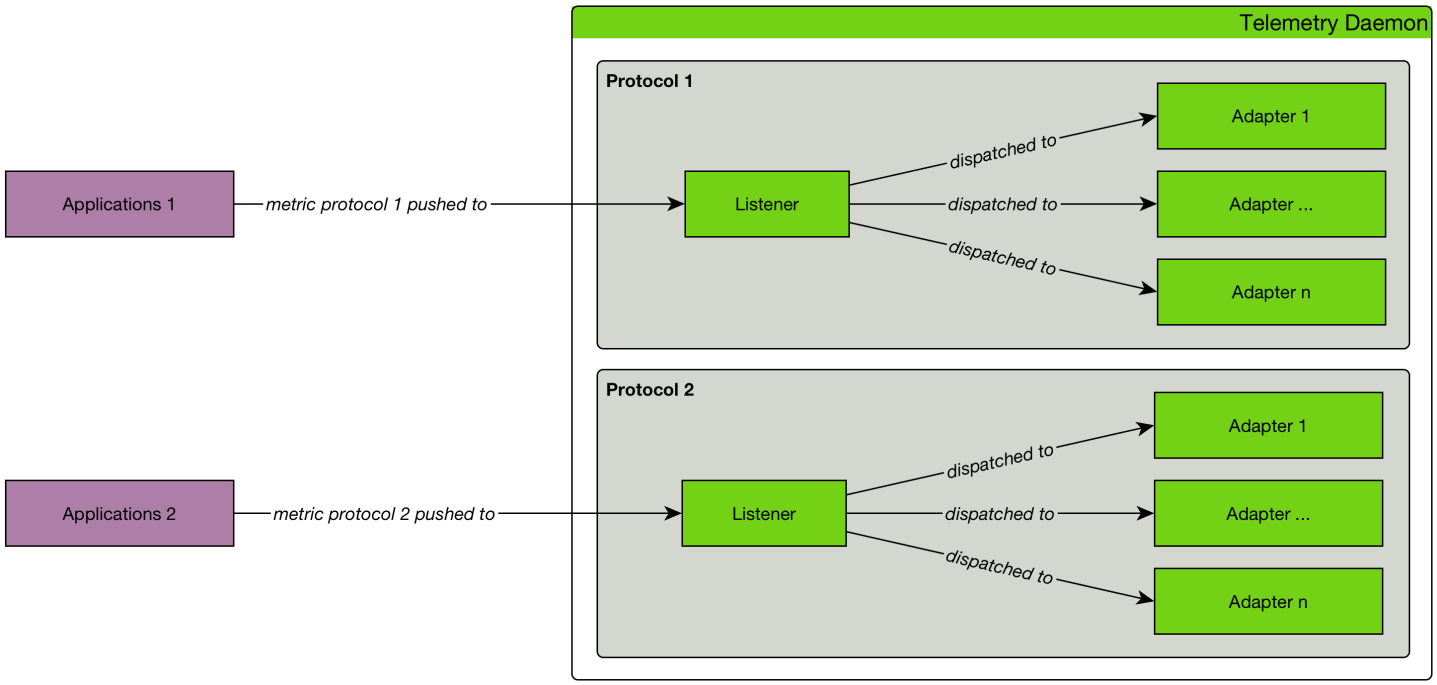
- External
- Very scalable
- Easier to consume data

sort by location

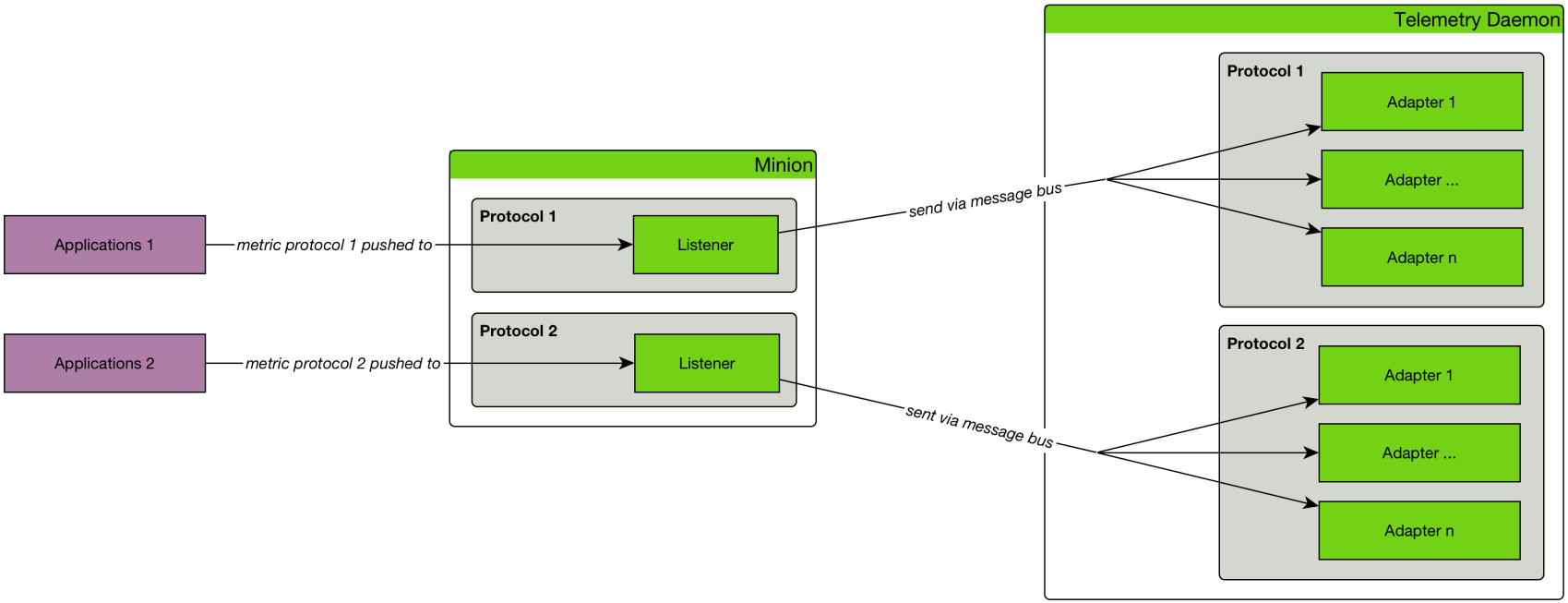


opennms drift

telemetryd



minion support for listener



data sources

Performance Metrics

- Junos Telemetry Interface (JTI)
- Cisco NX-OS

Flow Data

- IPFIX
- Netflow v5
- Netflow v9
- sFlow

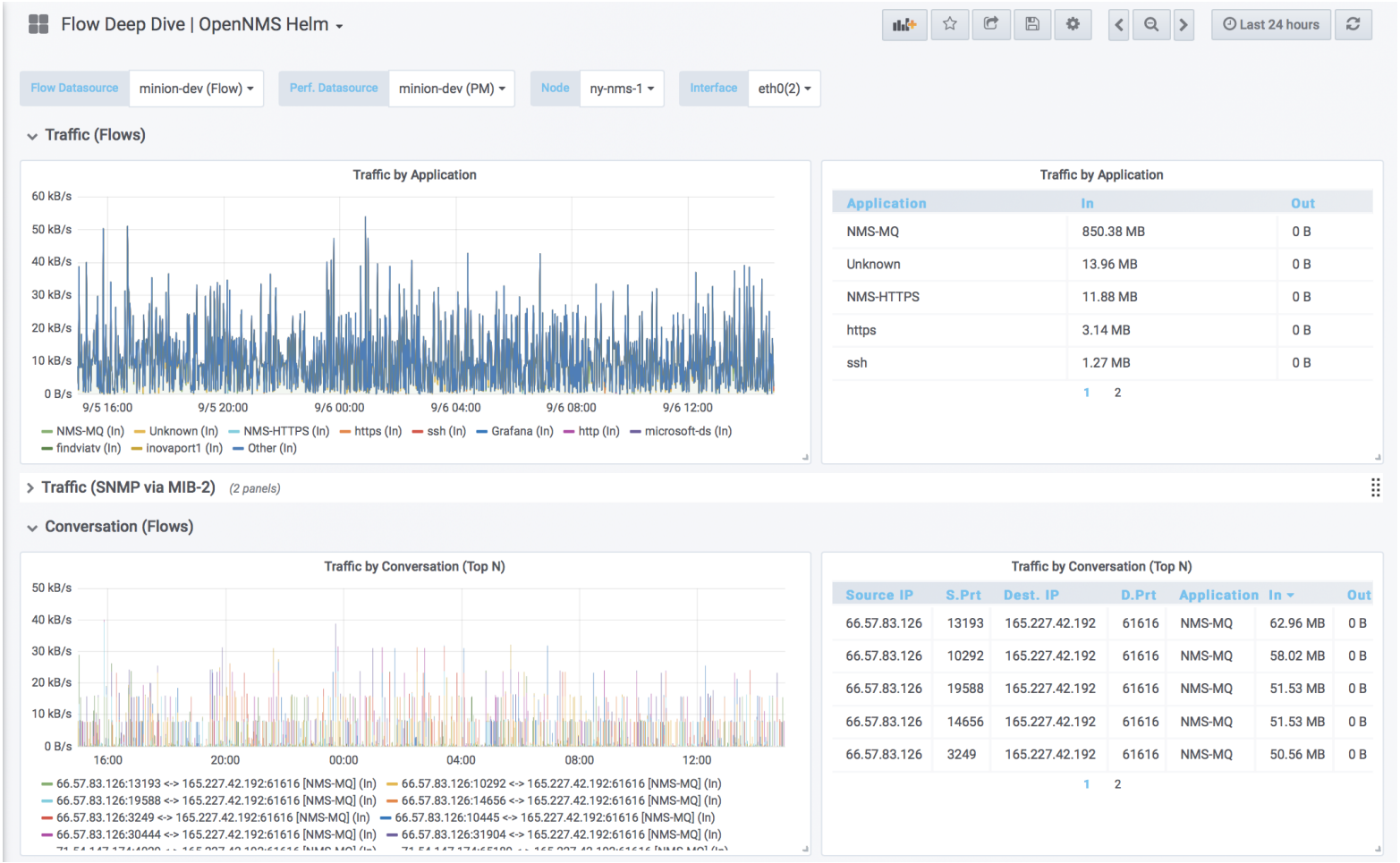
flow document

Field	Description
`@timestamp`	Timestamp in milliseconds at which the flow was sent by the exporter.
`location`	Monitoring location at which the flow was received. This will be `Default` unless you are using Minion.
`netflow.bytes`	Number of bytes transferred in the flow.
`netflow.last_switched`	Timestamp in milliseconds at which the last packet of the flow was transferred.
`netflow.direction`	`ingress` or `egress`
`netflow.first_switched`	Timestamp in milliseconds at which the first packet of the flow was transferred.
`netflow.last_switched`	Timestamp in milliseconds at which the last packet of the flow was transferred.
`netflow.input_snmp`	SNMP interface index on which packets related to this flow were received.
`netflow.output_snmp`	SNMP interface index on which packets related to this flow were forwarded.

how it works

- telemetryd is used to receive and decode flows.
- The telemetryd adapters convert the flows to a canonical flow model and dispatch these to the *flow repository*.
- The *flow repository* enriches the flows and persists them to *Elasticsearch*
  - Flows are tagged with an application name via the *Classification Engine*.
  - Metadata related to associated nodes such as ids and categories are also added to the flows.
- The REST API supports generating both summaries and time series data from the flows stored in the flow repository.
- *OpenNMS Helm* is used to visualize the flow data using the flow datasource that interfaces with the OpenNMS REST API.

opennms helm



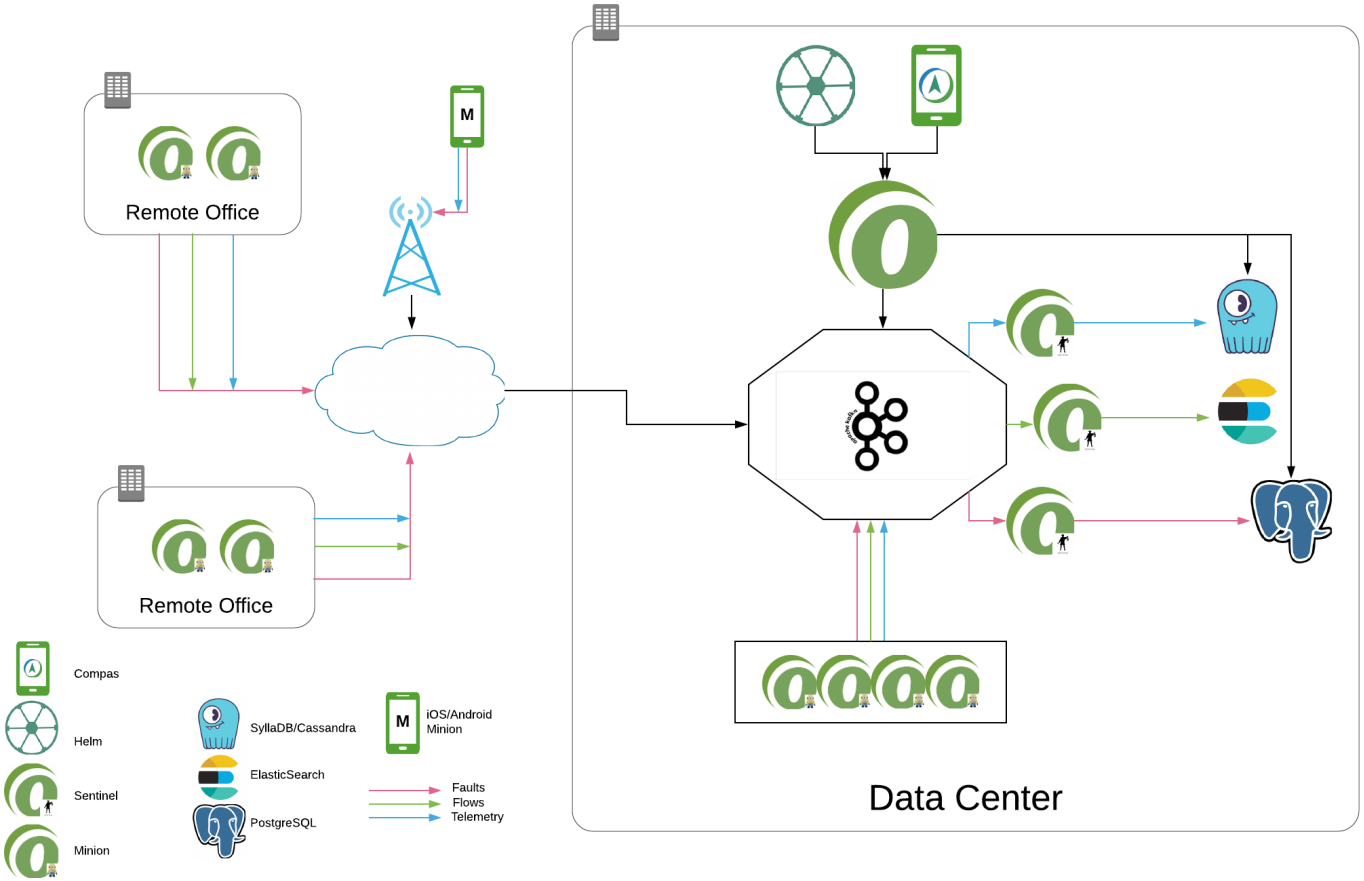
opennms futures

opennms sextant

Advanced alarms correlation

- Alarms can be grouped into objects called “situations”.
- NOC can focus on high level issues
- Currently implements spatial/temporal correlation, ML features planned

opennms sentinel



resources and q&a

The OpenNMS Project:

website: https://www.opennms.org

wiki: https://wiki.opennms.org

demo: https://demo.opennms.org

videos: https://www.youtube.com/user/opennms

chat: https://chat.opennms.com

forum: http://ask.opennms.eu