



Getting started with your own collector for Telemetry

Mike Korshunov, TME
mkorshun@cisco.com
DEVWKS-1980

Cisco *live!*
June 9-13, 2019 • San Diego, CA

#CLUS



Workshop Abstract



- We will review collector options;
 - Will check possible environments for tests;
 - Build & Run your own collector!
-
- Exercise based on: <https://github.com/ios-xr/telemetry-go-collector>

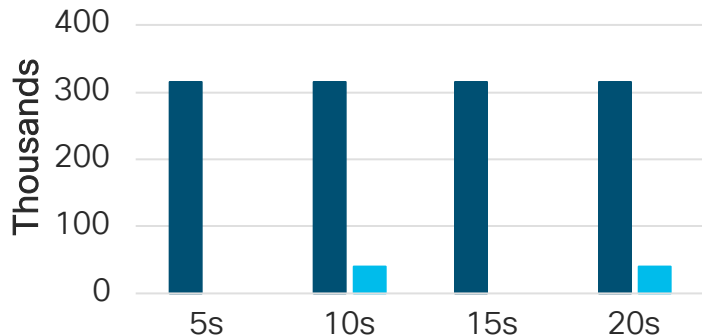
Introduction



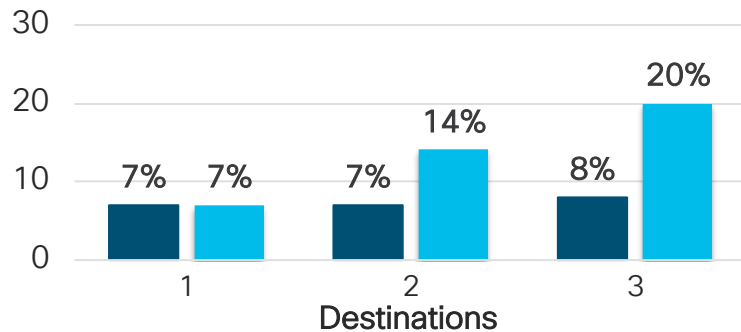
You make security **possible**

“Pushing” More Data Really Does Work Better

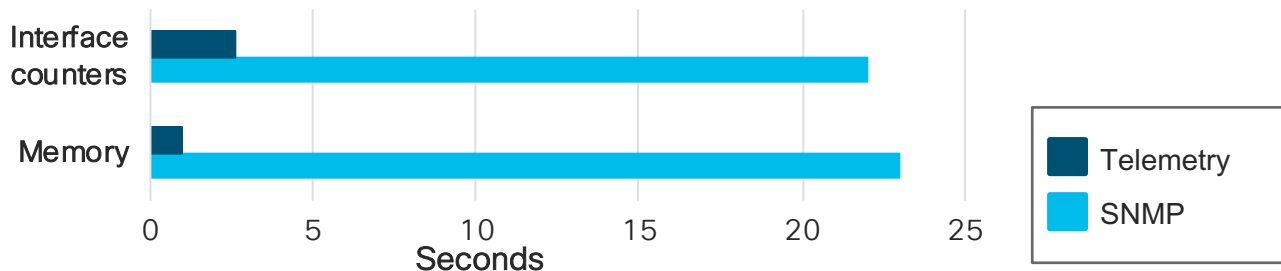
Counters



CPU load

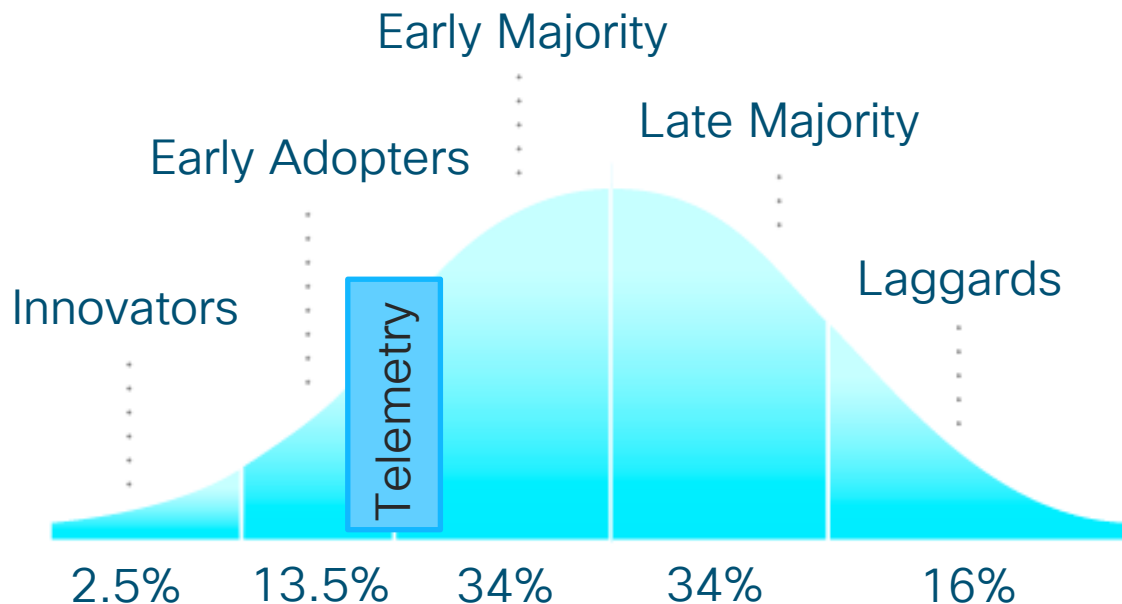


Time to collect all data (NCS5516, 576x100GE)

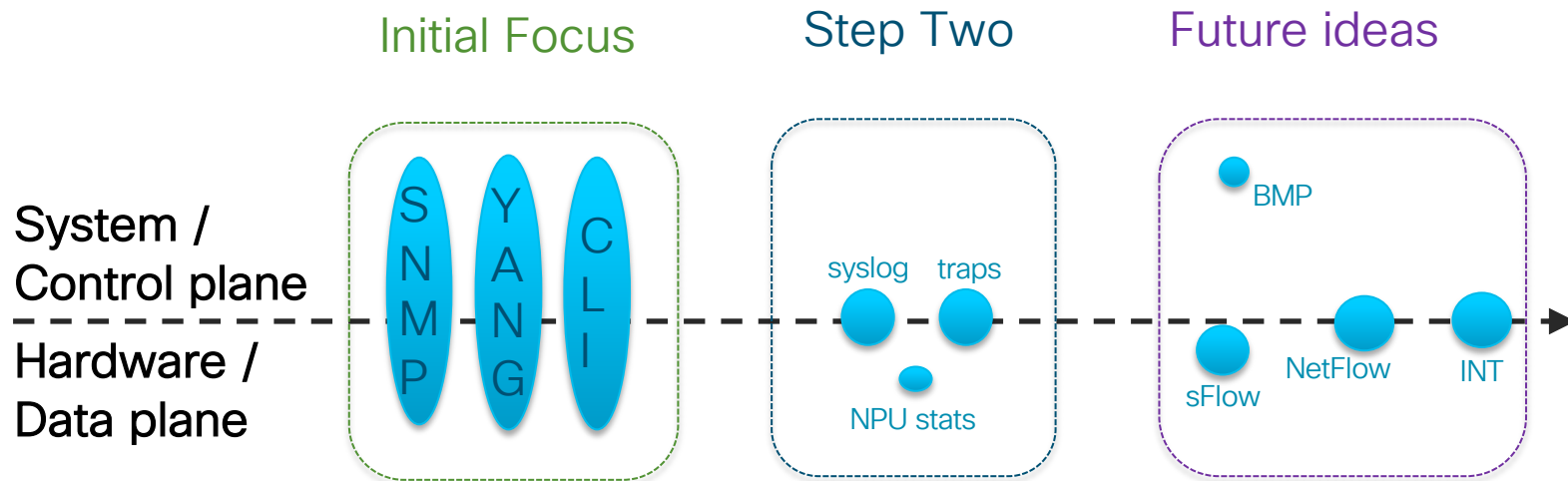


- ✓ More counter data
- ✓ Reduction in CPU load
- ✓ Faster collection

Three Years of Streaming Telemetry



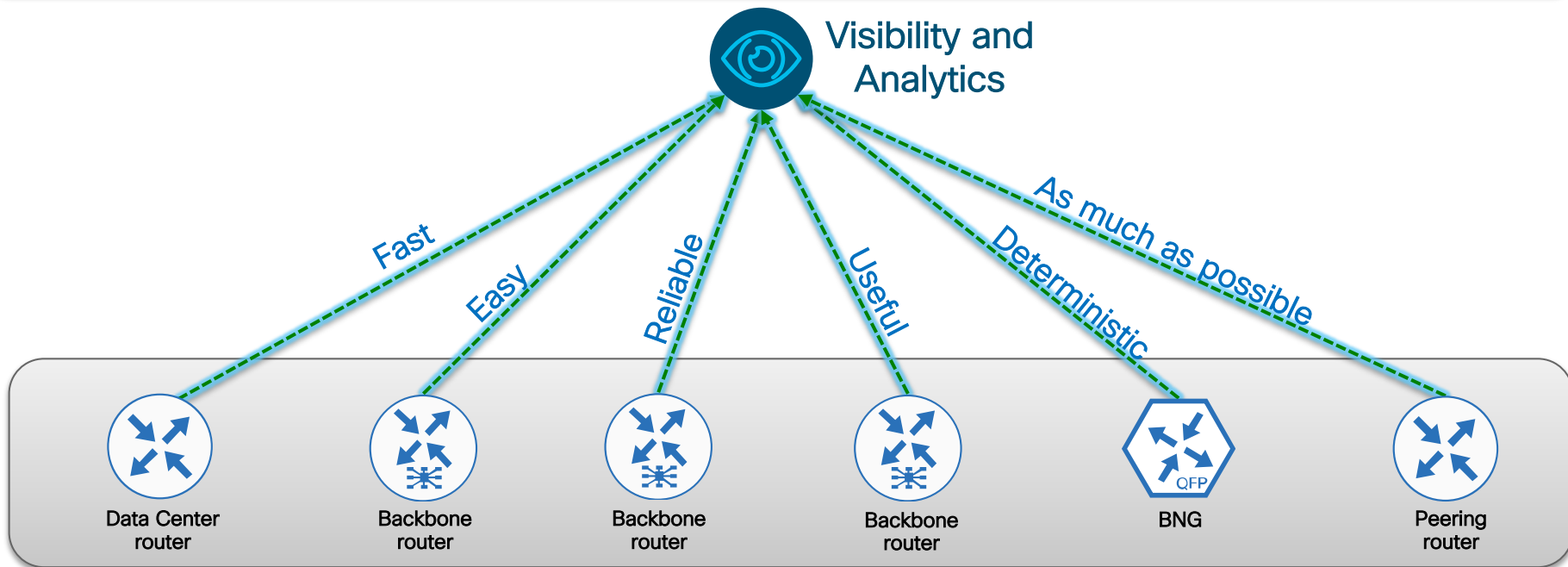
Streaming Telemetry Evolution – Coverage



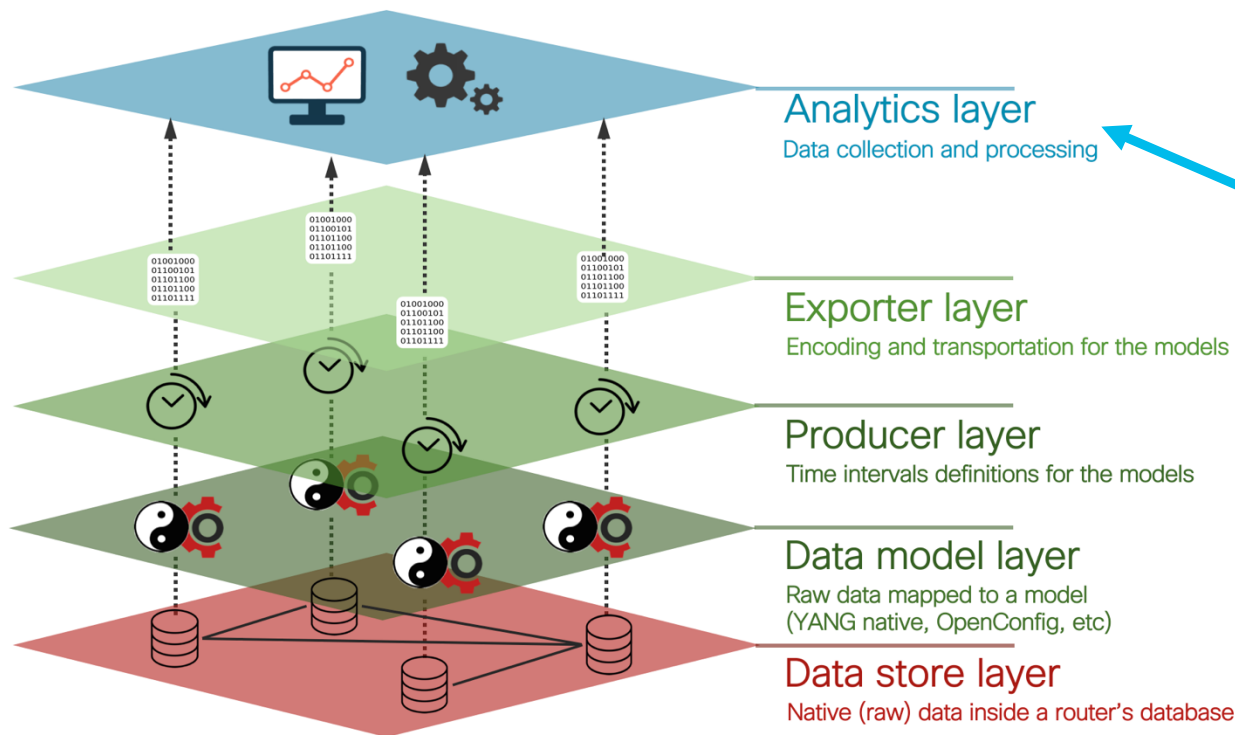
“Scream Stream If You Wanna Go Faster”

Telemetry: an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring.

<https://en.wikipedia.org/wiki/Telemetry>



How Do You See Telemetry?



We are here

Which Telemetry? The Exporter Layer



You make customer experience **possible**

So, What is Telemetry?

Telemetry: an automated communications process by which measurements and other data are collected at remote or inaccessible points and transmitted to receiving equipment for monitoring. The word is derived from Greek roots: *tele* = remote, and *metron* = measure.

<https://en.wikipedia.org/wiki/Telemetry>



meteorology

Cisco *live!*



medicine



wildlife research

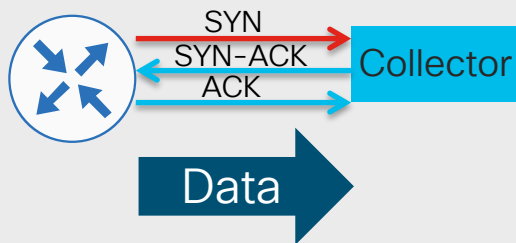


networks

* <http://adsabs.harvard.edu/abs/1987STIN...8913455>;
<https://en.wikipedia.org/wiki/Telemetry>

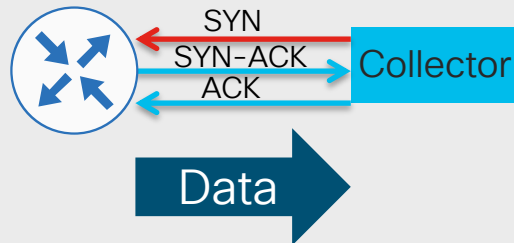
How to Select the Mode: Dial-in versus Dial-out

Dial-Out



- TCP & gRPC (from 6.1.1)
- UDP (from 6.2.1)

Dial-In



- gRPC only (from 6.1.1)

How to Select the Mode: Dial-in versus Dial-out

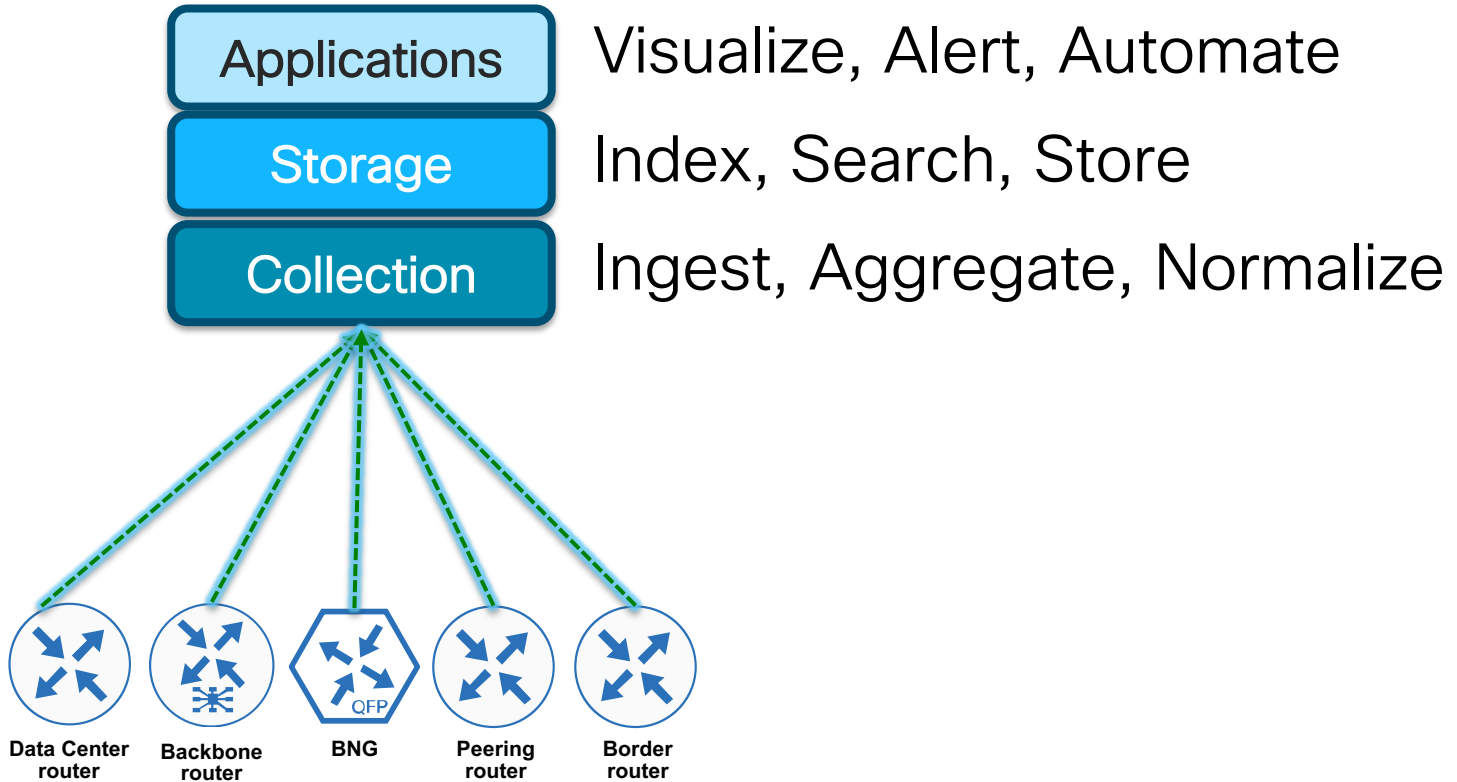
Dial-Out

- Broader flexibility in transport options
- No need to open up ports for inbound management traffic
- Anycast & Load-balancing

Dial-In

- A single channel (config and streaming)
- Listening port on the router
- No router configuration
- Transient connection
- Only gRPC/gNMI available

A Very Basic Analytics Platform Architecture



Step 1. Get the image & Run the container

```
cisco@cisco-desktop:~$ docker pull golang
```

```
Using default tag: latest
```

```
latest: Pulling from library/golang
```

```
c5e155d5a1d1: Pull complete
```

```
221d80d00ae9: Pull complete
```

```
4250b3117dca: Pull complete
```

```
3b7ca19181b2: Pull complete
```

```
aa24759e848f: Pull complete
```

```
927e9eaeed19: Pull complete
```

```
66293f4dacbd: Pull complete
```

```
Digest: sha256:cf0b9f69ad1edd652a7f74a1586080b15bf6f688c545044407e28805066ef2cb
```

```
Status: Downloaded newer image for golang:latest
```

```
cisco@cisco-desktop:~$ docker images | grep golang
```

golang	latest	7ced090ee82e	4 weeks ago
--------	--------	--------------	-------------

```
774MB
```

```
cisco@cisco-desktop:~$ docker run -it --rm --name my-telemetry-collector --network host golang
```

```
root@cisco-desktop:/go#
```

What options for controller do we have?



Pipeline



Telegraf



Your Own

What options for IOS-XRv tests do we have?



Vagrant

<https://xrdocs.io/application-hosting/tutorials/iosxr-vagrant-quickstart>



<https://devnetsandbox.cisco.com/RM/Diagram/Index/883f8ea6-54a1-453e-98f5-fc175a2a90de?diagramType=Topology>



Step 2. Clone the repo & Deal with dependencies

```
root@cisco-desktop:/go# go get -d github.com/ios-xr/telemetry-go-collector
package github.com/ios-xr/telemetry-go-collector: no Go files in /go/src/github.com/ios-xr/telemetry-go-collector
root@cisco-desktop:/go#
root@cisco-desktop:/go/src/github.com# cd /go/src/github.com/ios-xr/telemetry-go-collector
root@cisco-desktop:/go/src/github.com/ios-xr/telemetry-go-collector# ls
Dialout-collector-howto.md  bin      mdt_grpc_dialin  telemetry
telemetry_dialout_collector
README.md                  docs     mdt_grpc_dialout  telemetry_dialin_collector
root@cisco-desktop:~# go get -u github.com/golang/protobuf/protoc-gen-go
root@cisco-desktop:~#
root@cisco-desktop:~# go get -u google.golang.org/grpc
root@cisco-desktop:~#
root@cisco-desktop:/go# go build -o bin/telemetry_dialin_collector github.com/ios-xr/telemetry-go-collector/telemetry_dialin_collector
root@cisco-desktop:/go#
root@cisco-desktop:/go# ls bin/
protoc-gen-go  telemetry_dialin_collector
```

What's the packages inside?

ProtoBufs

- Go support for Protocol Buffers;
- If a proto file specifies RPC services, protoc-gen-go can be instructed to generate code compatible with gRPC

gRPC

- Package implements an RPC system called gRPC;
- Based on HTTP/2;
- A lot of benefits!



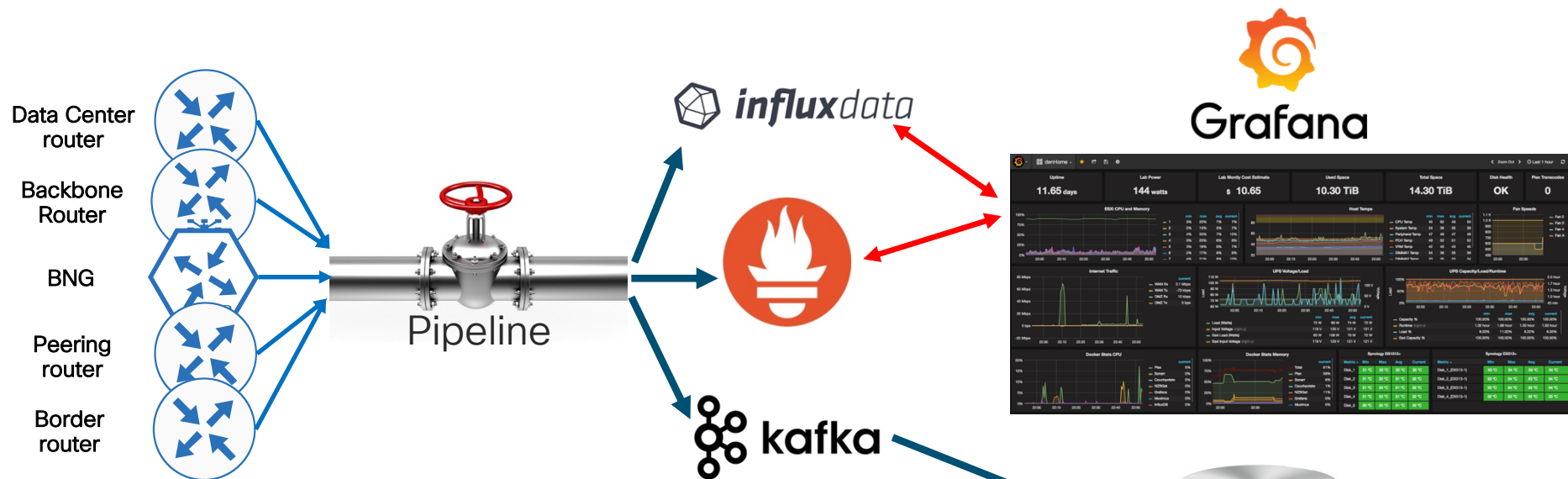
Step 3. Run The Collector!

```
root@cisco-desktop:/go# ./bin/telemetry_dialin_collector -server "host_ip:57751" -subscription
cdp -oper subscribe -username vagrant -password vagrant
mdtSubscribe: Dialin Reqid 30196 subscription [cdp]
{
  "node_id_str": "rtr1",
  "subscription_id_str": "cdp",
  "encoding_path": "Cisco-IOS-XR-telemetry-model-driven-oper:telemetry-model-
driven/destinations/destination",
  "collection_id": 1,
  "collection_start_time": 1560229527690,
  "msg_timestamp": 1560229527692,
  "data_json": [
    {
      "timestamp": 1560229527691,
      "keys": {
        "destination-id": "XRDOCS"
      },
      "content": {
        "id": "XRDOCS",
        "configured": 1,

```

Start Exploring Telemetry Today

Go With Open Source Tools



How to build up the stack:

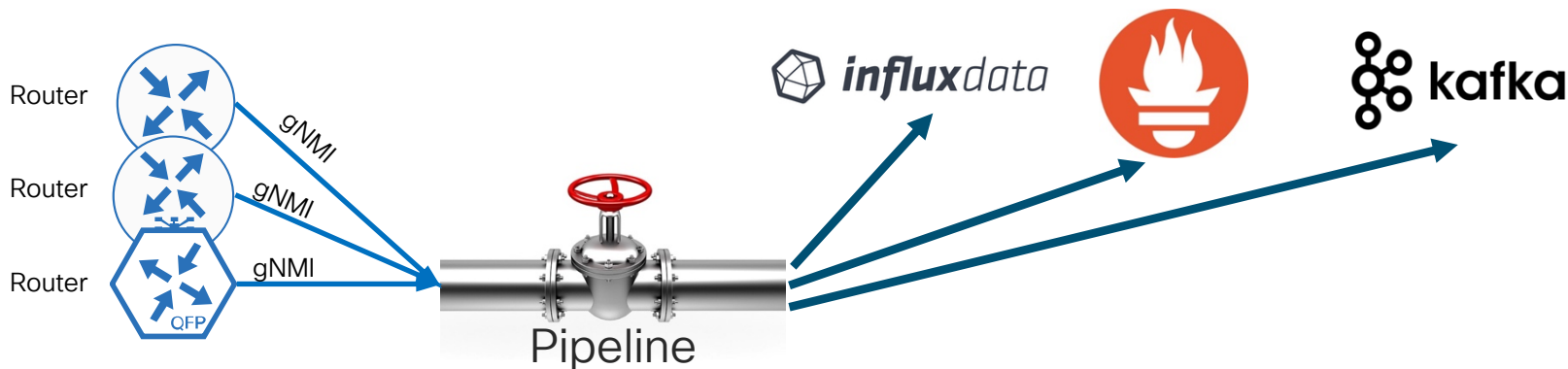
<https://xrdocs.io/telemetry/tutorials/2018-06-04-ios-xr-telemetry-collection-stack-intro>

How to add proto files to the pipeline:

https://github.com/nleiva/pipeline/blob/master/add_proto.md

Pipeline with gNMI

Start today



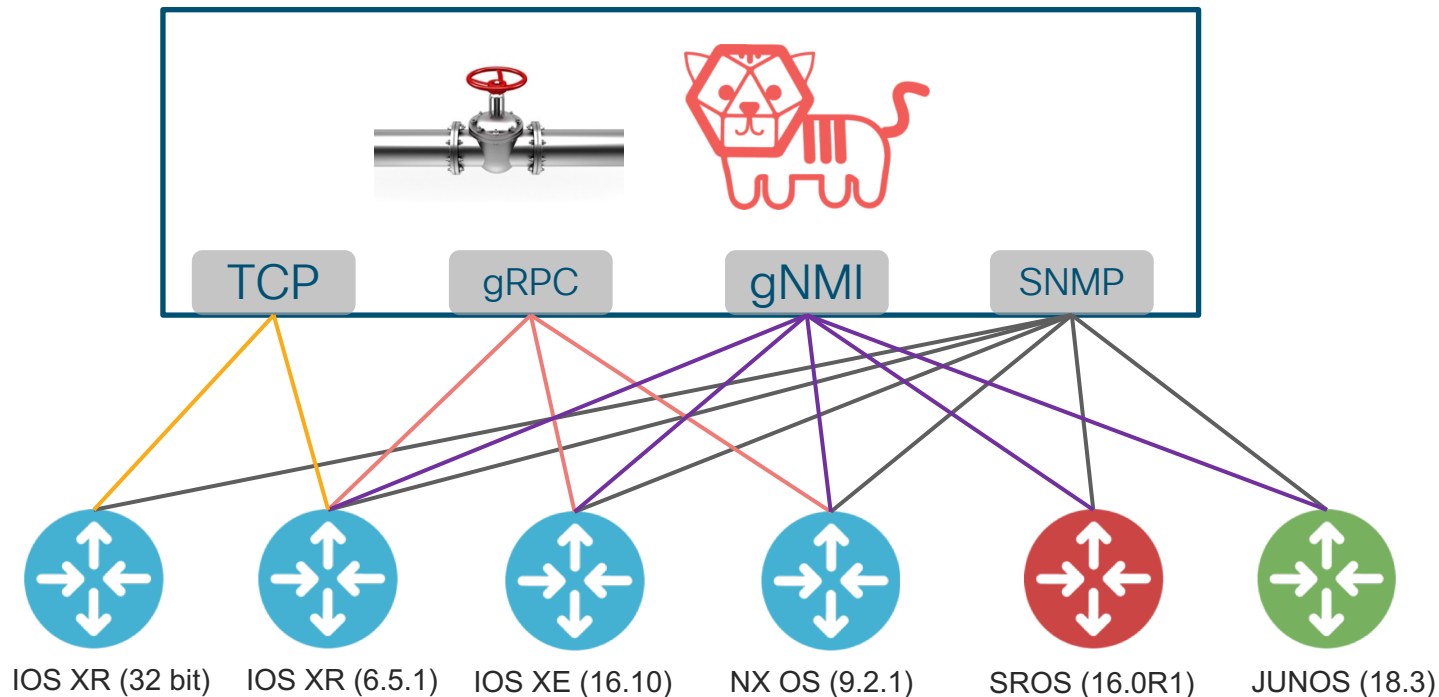
```
[mygnmirouter]
stage = xport_input
type = gnmi
server = 10.49.234.114:57777
```

```
# Sensor Path to subscribe to. No configuration on the device necessary
# Appending an @ with a parameter specifies subscription type:
# @x where x is a positive number indicates a fixed interval, e.g. @10 -> every 10 seconds
#
path1 = Cisco-IOS-XR-infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters@10
```

Telegraf + Pipeline = Universality

Try Telemetry With All Your Vendors

Available in
Telegraf 1.11



Telegraf + Pipeline Details

Telegraf Input Plugins

- CPU
- Disk
- DiskIO
- DNS Query
- Docker
- ElasticSearch
- Fluentd
- InfluxDB
- IPtables
- IPVS
- JTI OC Telemetry (Juniper)
- Kapacitor
- Kubernetes
- MySQL
- Net
- Netstat
- Network Response
- NGINX
- Nstat
- Procstat
- **SNMP**
- Syslog
- Telegraf
- TCP MDT (Cisco)
- gRPC Dial-IN w/ TLS (Cisco)
- gRPC Dial-IN w/o TLS (Cisco)
- gRPC Dial-OUT w/ TLS (Cisco)
- gRPC Dial-OUT w/o TLS (Cisco)
- gNMI (Cisco / other vendors)

Telegraf Output Plugins

- Amazon Cloud Watch
- Amazon Kinesis
- Amon
- AMQP
- **Apache Kafka**
- Crate DB
- Datalog
- Discard
- ElasticSearch
- File
- **Graphite**
- Graylog
- HTTP
- InfluxDB
- Instrumental
- Librato
- MSFT Azure Application Insights
- MSFT Azure Monitor
- MQTT Producer
- NATS Output
- NSQ
- **OpenTSDB**
- **Prometheus Client**
- Rieamann
- Socket Writer
- Stackdriver
- Wavefront

<https://docs.influxdata.com/telegraf/v1.9/plugins/inputs/>

<https://docs.influxdata.com/telegraf/v1.9/plugins/outputs/>

Conclusion



You make security **possible**

Summary and Key Messages

- ✓ It's easy to start with collector,
But hard to support
- ✓ Use gRPC or TCP, when it's possible
- ✓ Use KV-GPB. It's a good mid point between efficiency and convenience



Telemetry References

- <https://www.cisco.com/c/en/us/solutions/service-provider/cloud-scale-networking-solutions/model-driven-telemetry.html>
- <https://xrdocs.github.io/telemetry>
- <https://xrdocs.io/telemetry/tutorials/2018-06-04-ios-xr-telemetry-collection-stack-intro/>
- Start with Telemetry in 15 minutes. Automated bring up stack [already available](#).
- Reserve a DevNet Sandbox:
<https://devnetsandbox.cisco.com/RM/Diagram/Index/883f8ea6-54a1-453e-98f5-fc175a2a90de?diagramType=Topology>

References:

Breakouts:

- Day-2 Telemetry better – Network Insights for ACI/NX-OS – **BRKDCN-2712** by Karishma Gupta
- Application Hosting and Model Driven Telemetry on IOS XE – **BRKCRS-2004** by Jeremy Cohoe
- NX-OS Automation and Telemetry Made Simple, Powerful and Scalable with Open-Source Tools – **BRKDCN-2025** by Nicolas Delecroix
- Advanced Topics in Cisco OS Telemetry – **BRKSPG-2503** by Benoît Claise & Mike Korshunov

DevNet:

- Streaming telemetry: The value of "realtime" analytics for the network – **DEVNET-2062** by Stuart Clark
- Workshop – Real time telemetry with Go – **DEWKS-3000** by Luis Flores Kanter
- Workshop – Streaming Telemetry with NX-OS – **DEWKS-2624** by Gerard Sheehan
- Workshop – Use Docker to Orchestrate a Telemetry Analytics Solution – **DEWKS-1224** by Randy Zhang

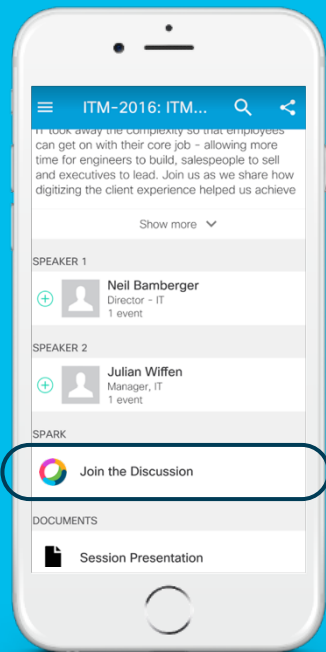


Thank you





You make **possible**



cs.co/ciscolivebot#DEVWKS-1980

Cisco Webex Teams

Questions?

Use Cisco Webex Teams (formerly Cisco Spark) to chat with the speaker after the session

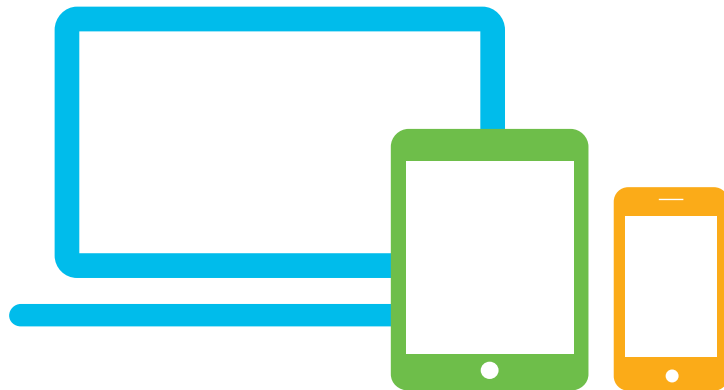
How

- 1 Find this session in the Cisco Events Mobile App
- 2 Click “Join the Discussion”
- 3 Install Webex Teams or go directly to the team space
- 4 Enter messages/questions in the team space

Complete your online session survey

- Please complete your Online Session Survey after each session
- Complete 4 Session Surveys & the Overall Conference Survey (available from Thursday) to receive your Cisco Live T-shirt
- All surveys can be completed via the Cisco Events Mobile App or the Communication Stations

Don't forget: Cisco Live sessions will be available for viewing on demand after the event at ciscolive.cisco.com



Continue Your Education

