Model-Driven Telemetry in an ISP Environment

Yannis Nikolopoulos
(yanodd@otenet.gr)

COSMOTE

“With model-driven telemetry (MDT), routers can stream out large amounts of operational data in a highly efficient, easily consumable way”

Current Model & Shortcomings

- Pull model to gather Operational stats
- Scaling (e.g. Syslog, SNMP)
- CPU load (e.g. Syslog, SNMP)
- Complex Structure (e.g SMI)
- In other words: **lack of efficiency**

Telemetry (Streaming)

- Data streamed from network elements (wide variety)
- near real-time monitoring
- data on-change functionality
- Get as much data off the box as quickly as possible
- Flexible data serialization (fit own tools and automation procedures)
- Cisco's MDT utilizes YANG to model data

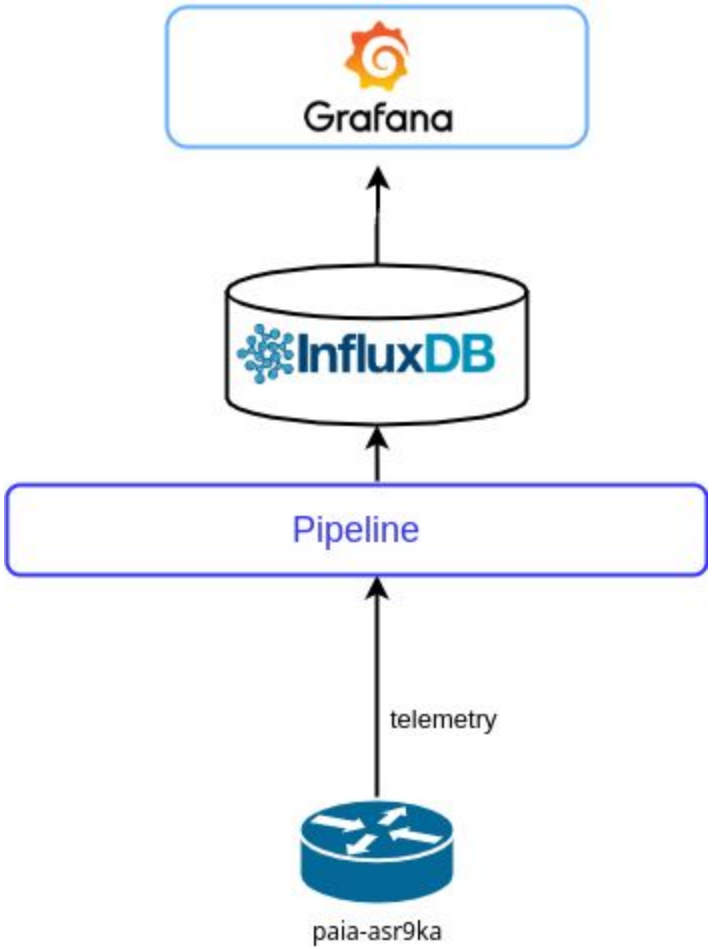
New Monitoring Requirements

- High-resolution data (even down to a few seconds)
- big data storage & analysis (store and analyze)
- Dial out model (no polling, CPU-friendly)
- Various consuming needs
- Event prediction & Automated actions (e.g. IP pool allocations)

Early PoCs

- Test Lab
  - local Cisco ASR9010 streaming IfStats, CPU/RAM, QoS
  - Pipeline to collect
    - Cisco developed, open-sourced, limited capabilities
    - JSON config file produced manually (a pain…)
  - InfluxDB as TSDB
    - straight forward
  - Grafana as frontend

Test Lab Diagram

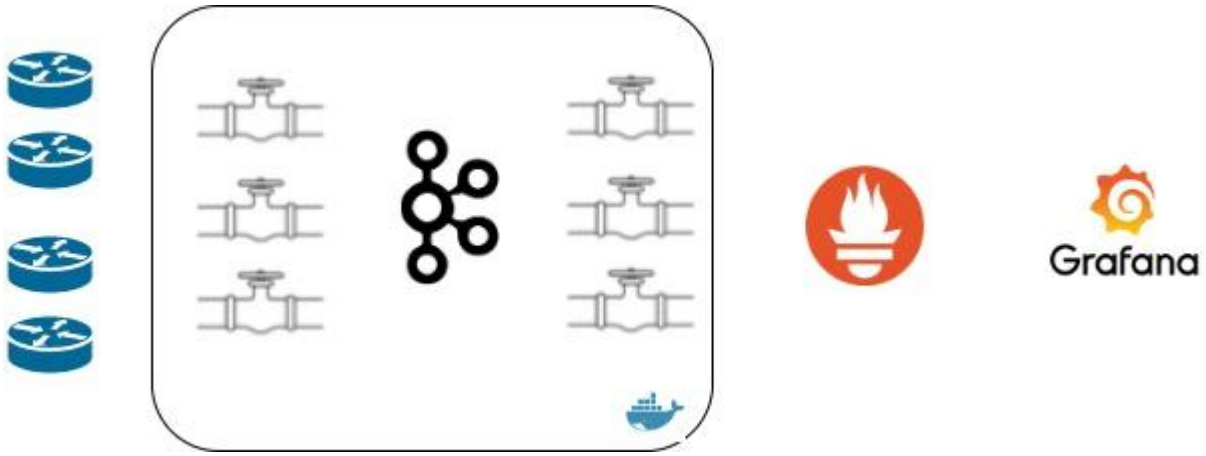


- single BNG streaming IfStats, CPU/MEM, QoS
- Lots of tweaking for metrics.json
- Pyang + yang models

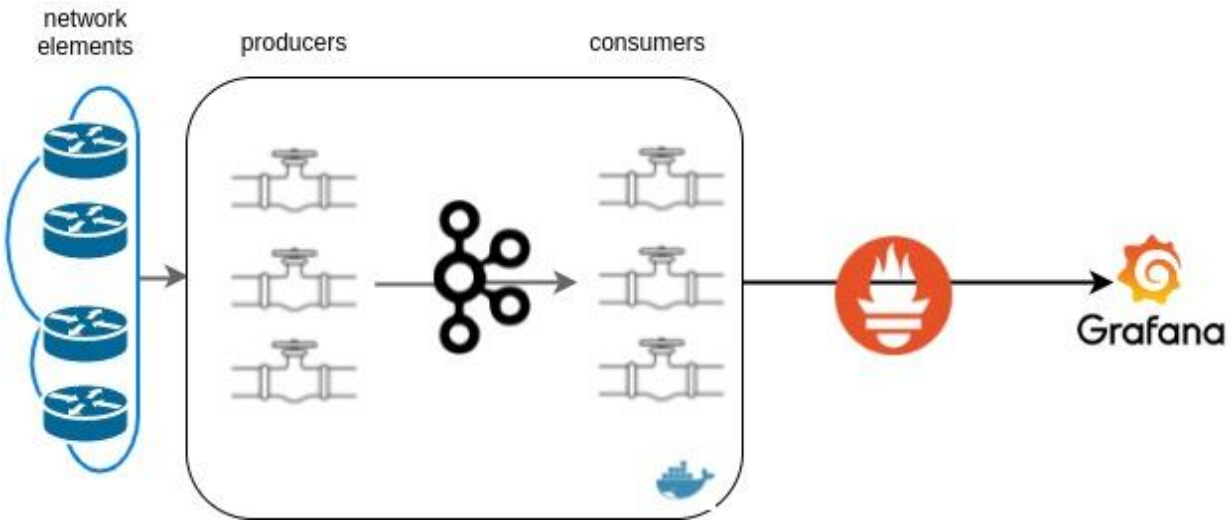
Telemetry - Phase I

- Multiple pipeline instances (Cisco limitation)
- Pipeline as producer and consumer (w/ Kafka bus)
- stack deployed as docker containers (via compose) over baremetal
- JSON config file (metrics.json) produced from cisco tool
- Complex and not so scalable (pipeline mainly)
- Prometheus high availability issue
- What about data storing/aggregation/consolidation?

Telemetry - Phase I



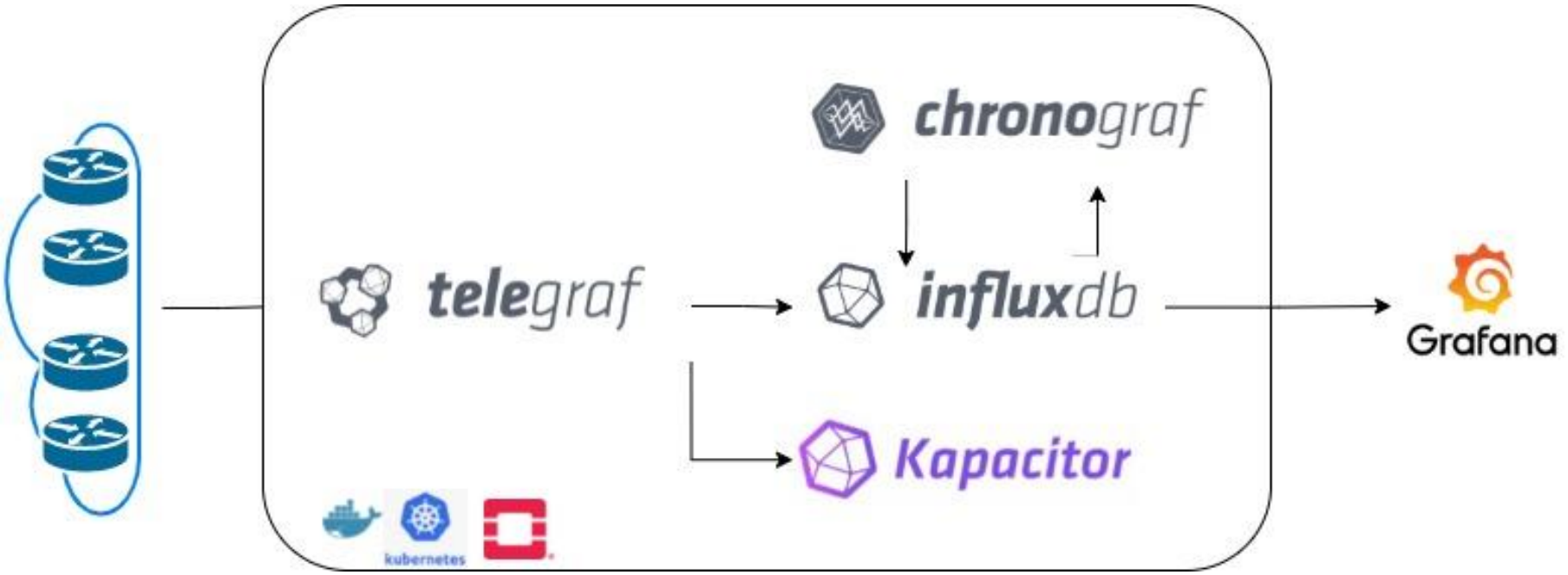
Telemetry - Phase I



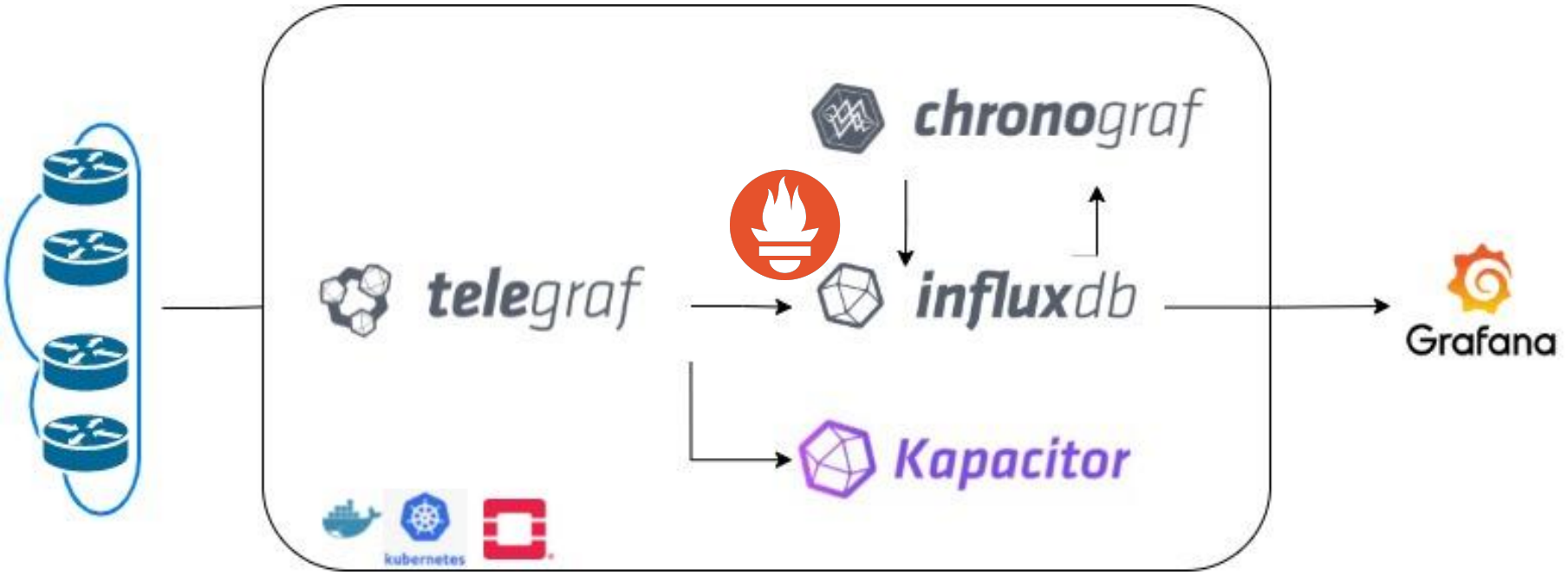
Telemetry - Phase II

- Pipeline ~~to be~~ replaced by telegraf
- Prometheus to be complemented by InfluxDB
  - Influx supports clustering (paid plan)
- Kapacitor and chronograph to be added (alerting, querying etc)
- What about data storing/aggregation/consolidation?
  - considering various options
  - e.g. Prometheus high-res data, Influx historical (lo-res)
- TICK stack deployed as docker containers via compose (for now)
  - To be replaced by X containers on k8s over OpenStack

Telemetry - Phase II



Telemetry - Phase II



Useful References

https://www.tail-f.com/what-is-yang/

https://github.com/YangModels/yang

https://github.com/cisco/bigmuddy-network-telemetry-pipeline

https://www.influxdata.com/products/influxdb-overview/

https://docs.influxdata.com/telegraf/v1.18/plugins/#cisco_telemetry_mdt

https://grafana.com/

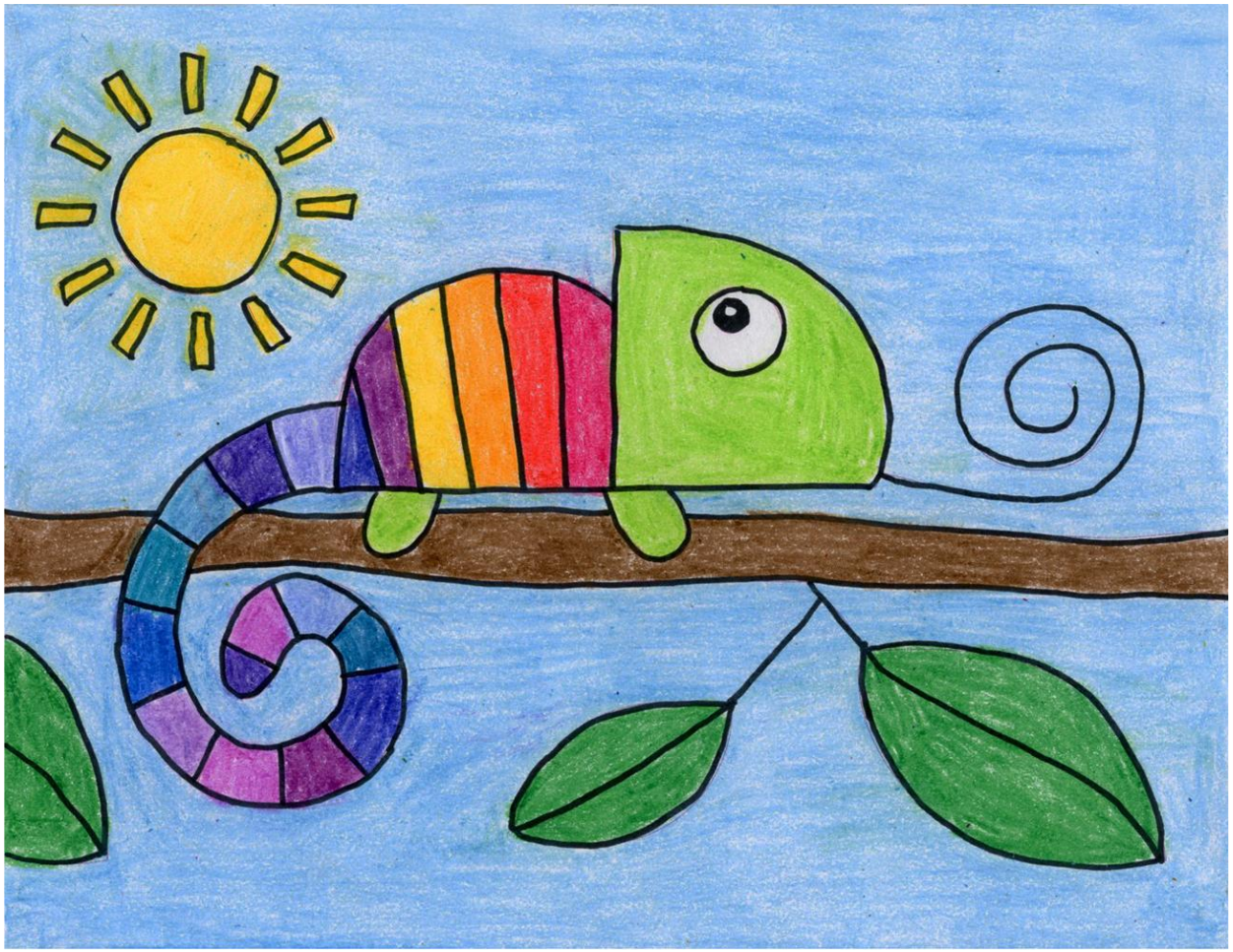
https://xrdocs.io/telemetry/

https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-5/telemetry/configuration/guide/b-telemetry-cg-asr9000-65x/b-telemetry-cg-asr9000-65x_chapter_010.html

ACK

Theodore Kyriakidis (Cisco)
Takis Samanis (Cisco)
Nia Provia (OTE)
Dimitris Stamatiadis (OTE)
Marinos Chondrogiannoglou (OTE)

a Short Demo with Pretty Pictures



Backup Slides

Notes on Configuration (Cisco IOS-XR)

```
telemetry model-driven
 destination-group troy
  address-family ipv4 83.235.79.70 port 5432
   encoding self-describing-gpb
   protocol tcp
  !
 !
 sensor-group health-sg
  sensor-path Cisco-IOS-XR-shellutil-oper:system-time/uptime
  sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
 !

 sensor-group subscriber-sg
  sensor-path Cisco-IOS-XR-ip-daps-oper:address-pool-service/nodes/node/pools
 !
 subscription health
  sensor-group-id health-sg strict-timer
  sensor-group-id health-sg sample-interval 30000
  destination-id troy
  destination-id mdt_pipeline2
```

Where to stream the info (can have multiple destination-groups i.e.collectors)

What info to stream (can have multiple sensor-groups)

A subscription can consist of multiple Sensor-groups and destinations

Notes on Configuration (IOS-XR)

```
telemetry model-driven
 destination-group troy
  address-family ipv4 83.235.79.70 port 5432
   encoding self-describing-gpb
   protocol tcp
  !
 !
 sensor-group health-sg
  sensor-path Cisco-IOS-XR-shellutil-oper:system-time/uptime
  sensor-path Cisco-IOS-XR-wdsysmon-fd-oper:system-monitoring/cpu-utilization
 !

 sensor-group subscriber-sg
  sensor-path Cisco-IOS-XR-ip-daps-oper:address-pool-service/nodes/node/pools
 !
 subscription health
  sensor-group-id health-sg strict-timer
  sensor-group-id health-sg sample-interval 30000
  destination-id troy
  destination-id mdt_pipeline2
```

https://developers.google.com/protocol-buffers

YANG model

Path of metric