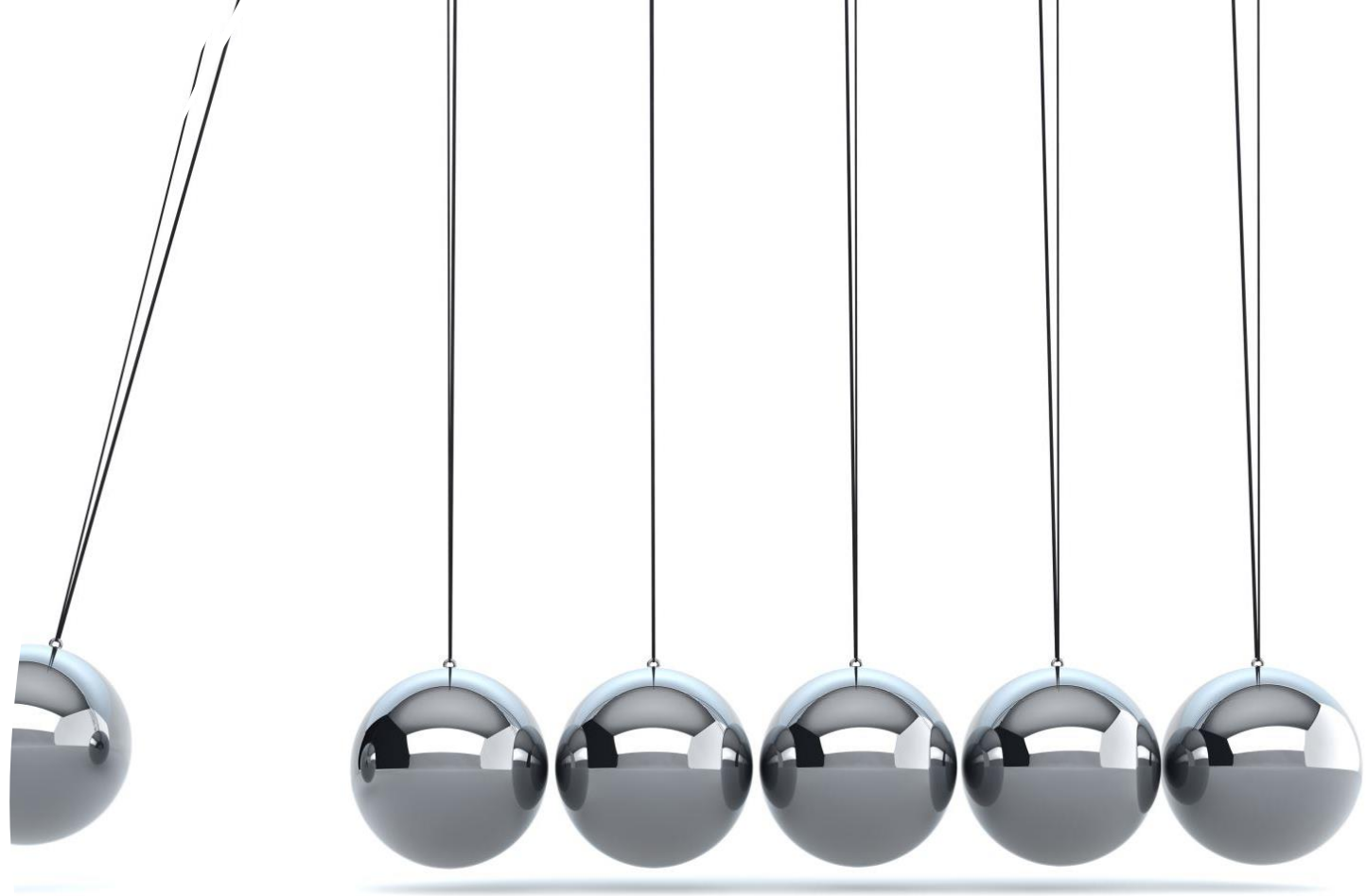# Case Study

Re-define network visibility for capacity planning & forecasting with Grafana

CASE-1: Objective/Goal

Integrate & Correlate Usage & Latency data sets from multiple Collectors/Probes for Capacity Planning & Performance Analytics

Popular Monitoring Tools used in BD

Monitoring Software Type Comparison

**ICMP Based monitoring**

- Zabbix

- Solarwinds

- PRTG

- Smokeping

- Nagios

SNMP Based Monitoring

- Zabbix

- Solarwinds

- PRTG

- Cacti

- LibreNMS

- Observium

- Nagios

Operational Challenges

- Tracking & Logging Latency deviation from ISP prefixes
- Transit Provider Performance Analysis correlating ICMP & SNMP data sets
- Too much manual works also time consuming & error prone
- Timeline based aggregated comparison was not possible
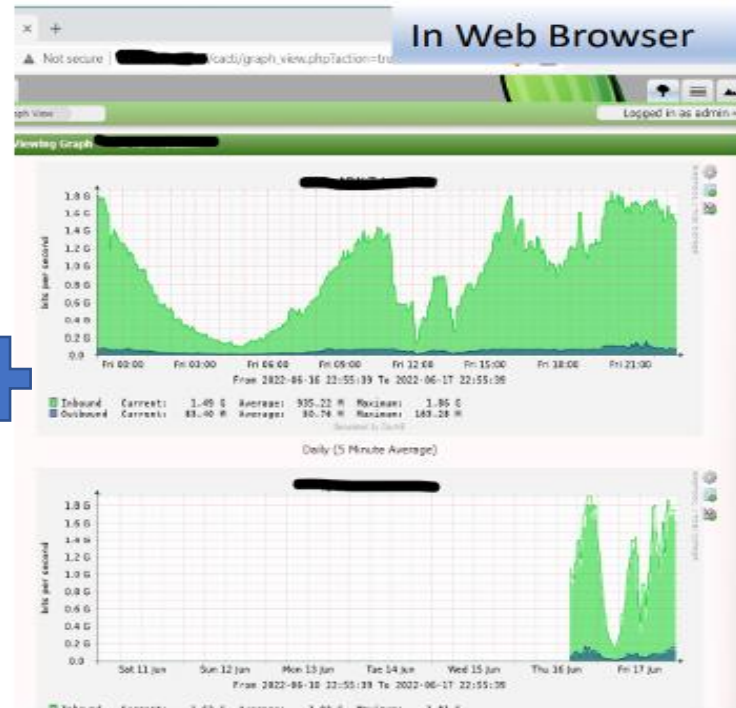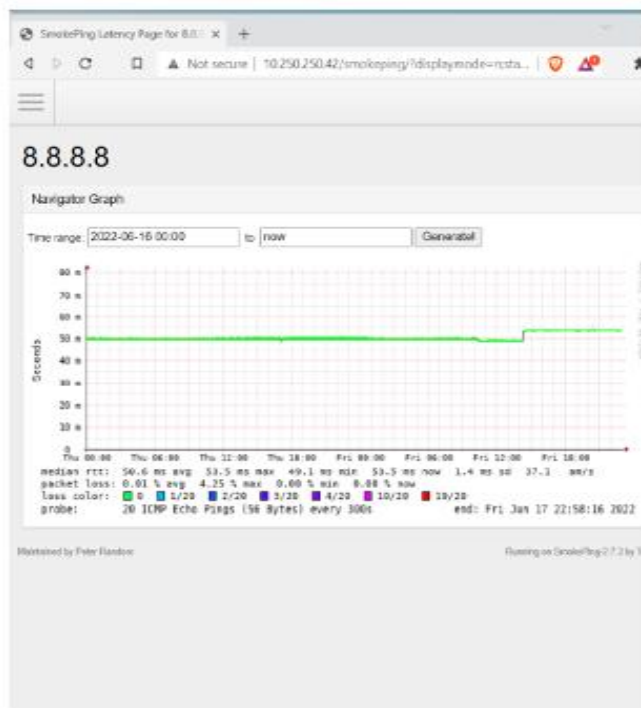
Operational Challenges (contd.)

- **Capacity planning & forecast**:
  - Threshold target was not statistically available
  - Aggregated Transit usage was needed for upgrade/downgrade plans
  - Management Dashboard was needed

- **Reporting:**
  - Manual records with multiple Screen Shots and Excel file logs for incidents

- **Ops Notes:**
  - Need to keep the existing platforms safe (data integrity) to avoid unforeseen risks
  - Need solution for cross platform data integration

Monitoring System Topology: Existing
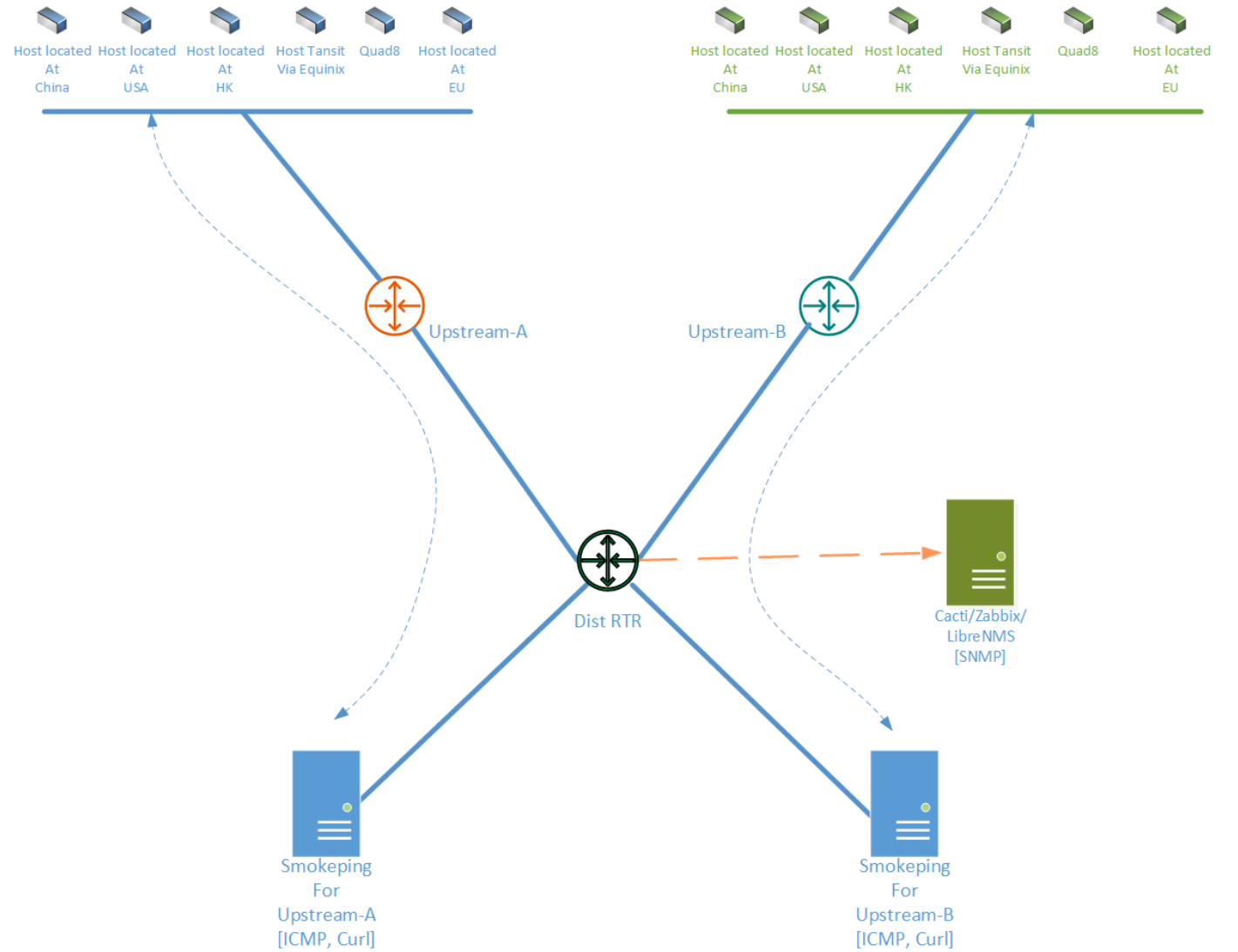
Selecting Grafana

• Grafana is an open-source interactive data-visualization platform, developed by Grafana Labs, which allows users to see their data via charts and graphs that are unified into one dashboard (or multiple dashboards!) for easier interpretation and understanding. You can also query and set alerts on your information and metrics from wherever that information is stored.
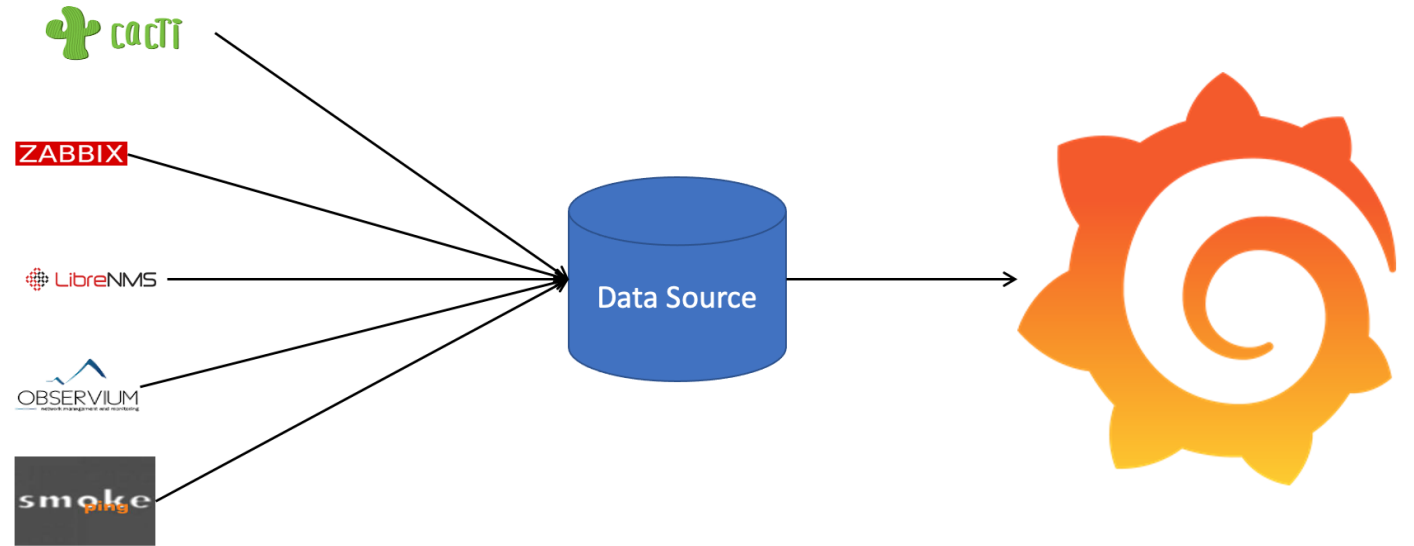
Monitoring System Topology: with Grafana
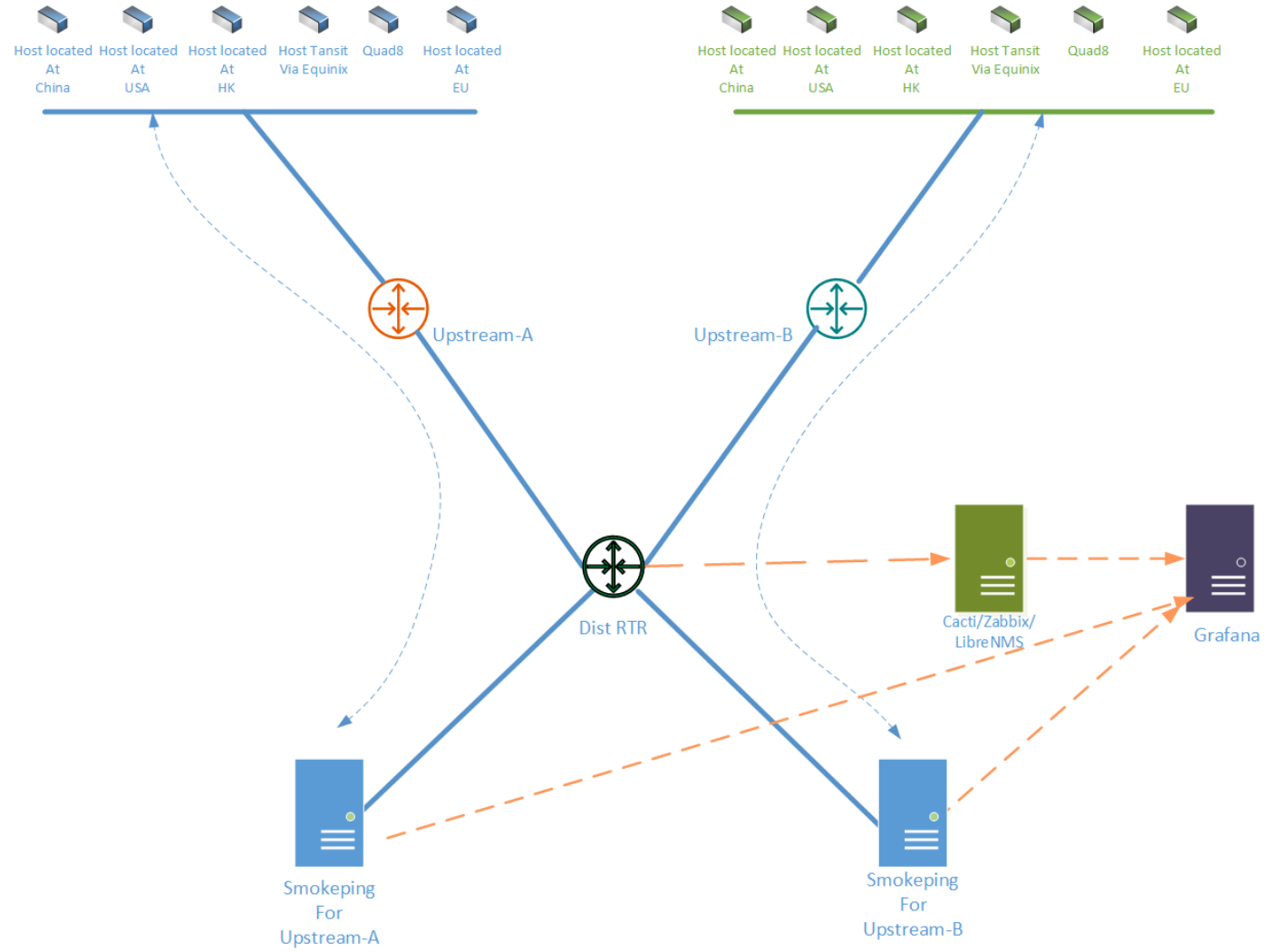
Data Sources for Grafana integration

- Cacti & Smokeping
  - Grafana SimpleJson datasource with Grafana-rrd-Server

- Zabbix
  - Zabbix App

- Librenms & Observium
  - Grafana mysql data source [Need to configure mysql database to allow Grafana Server IP for accessing database and run the query]
  - Grafana InfluxDB Data Source

Grafana Integration Steps

Installation Steps:

1.   Install Grafana

2.   Install Grafana-rrd-Server in Cacti & Smokeping server

3.   Add SimpleJson Data-Sources at Grafana server

4.   Add Zabbix App at Grafana server

Configuration Steps:

1.   Create a new Dashboard at Grafana

2.   Selected the *data-source* and run required query

3.   Select in-built graphing options for graph

4.   Start ICMP & SNMP based data correlation & performance analysis

Installation & Integration Challenges

Key notes on RRD:

- Most of the graphing softwares like ***cacti/smokeping/librenms*** store values to ***rrd files*** and use those rrd files to generate graphs
- These ***rrd files*** need to be readable via Grafana SimpleJSON data-collector

Importing rrd values:

- Install small software/script called **Grafana RRD Server** which works as a bridge between existing rrd based graphing softwares and Grafana
- Follow online installation process:

    Download/Install Link: https://github.com/doublemarket/grafana-rrd-server

- Allow server firewall to listen to port 9000 for Grafana

    grafana-rrd-server -p 9000 -r /path/to/rrds -s 300

Configuration Samples
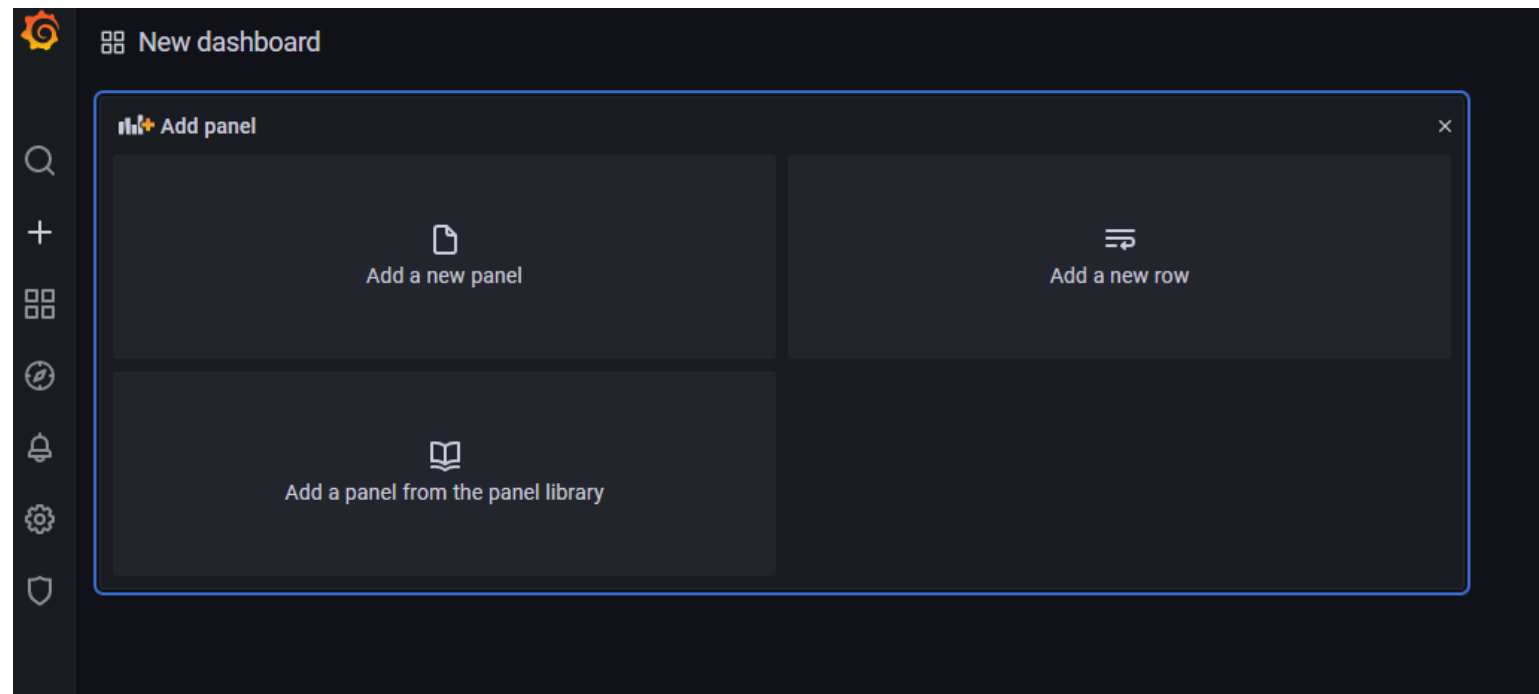
Adding Data Source

Configuration Samples

- **Data-Source Panel:**

- Each Remote server [Cacti/Smokeping/Zabbix] needs a separate data-source
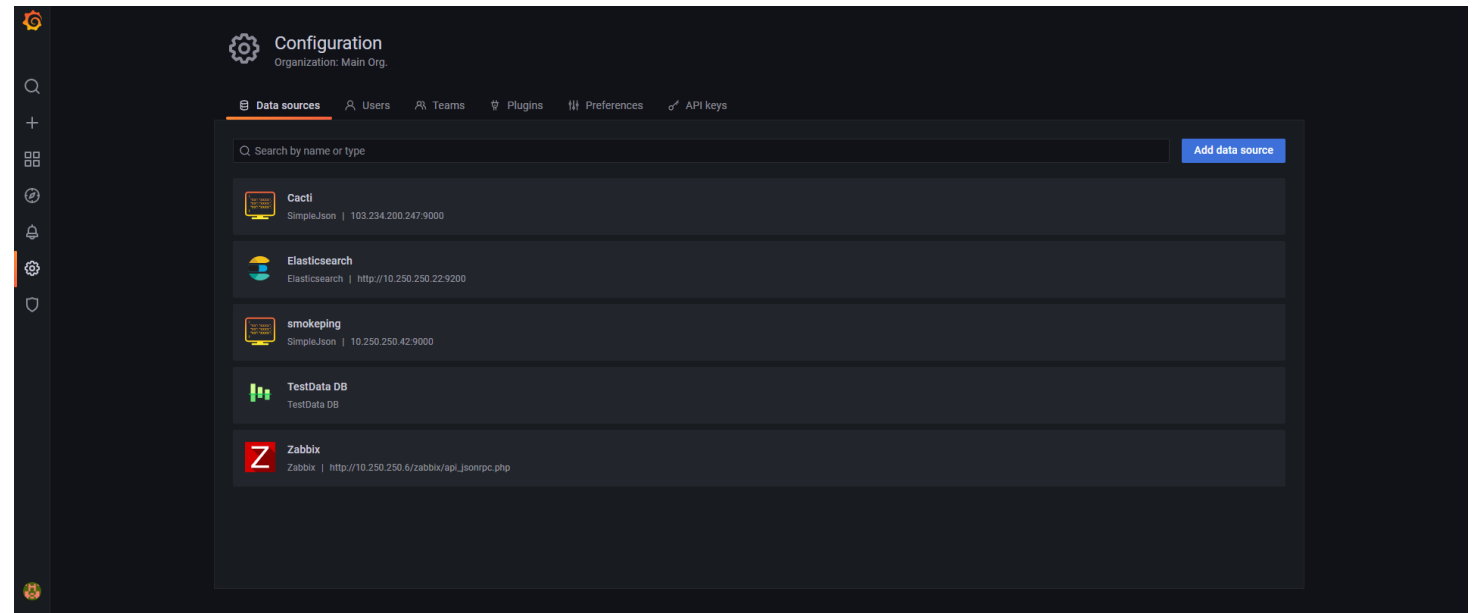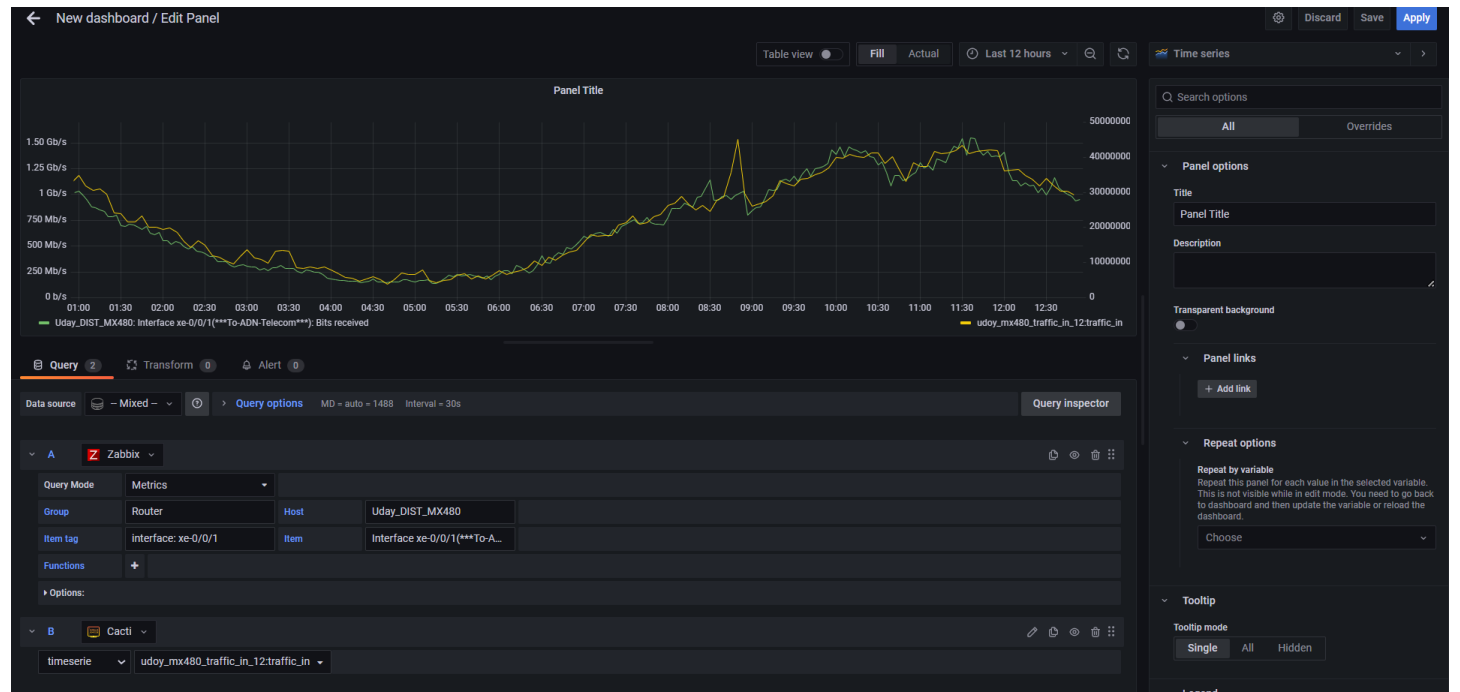
Configuration Samples

Creating Dashboard

Configuration Samples

Creating Dashboard – contd.

Outcome – Cross Platform Data Correlation

Multiple data-sources are combined into one dashboard

# Results – *we need a end point!*

Challenges

- **Capacity planning & forecast**:
    - Threshold target was not statistically available
    - Aggregated Transit usage was needed for upgrade/downgrade plans
    - Management Dashboard was needed

- **Reporting:**
    - Manual records with multiple Screen Shots and Excel file logs for incidents

- **Ops Notes:**
    - Need to keep the existing platforms safe (data integrity) to avoid unforeseen risks
    - Need solution for cross platform data integration

Results

- **Capacity planning & forecast**:
    - Link Threshold found in between 89-91% utilization
    - Timeline based aggregated Transit usage was recorded in 5 mins granular interval with no data loss.
    - Management Dashboard was made possible

- **Reporting:**
    - Accurately report weekly peak usage and plan for necessary upgrade/downgrade
    - Live Graph comments were available for more readily available reports

- **Ops Notes:**
    - Avoided the risk of re-tuning all existing platforms
    - Achieved Cross-Platform Data integration for correlated analytics

Still not done!!! ☺

- Exporting data (CSV):

- 5 min interval Full/Partial records are available in CSV format in a couple of clicks.

- This can later be used for more extensive reporting!
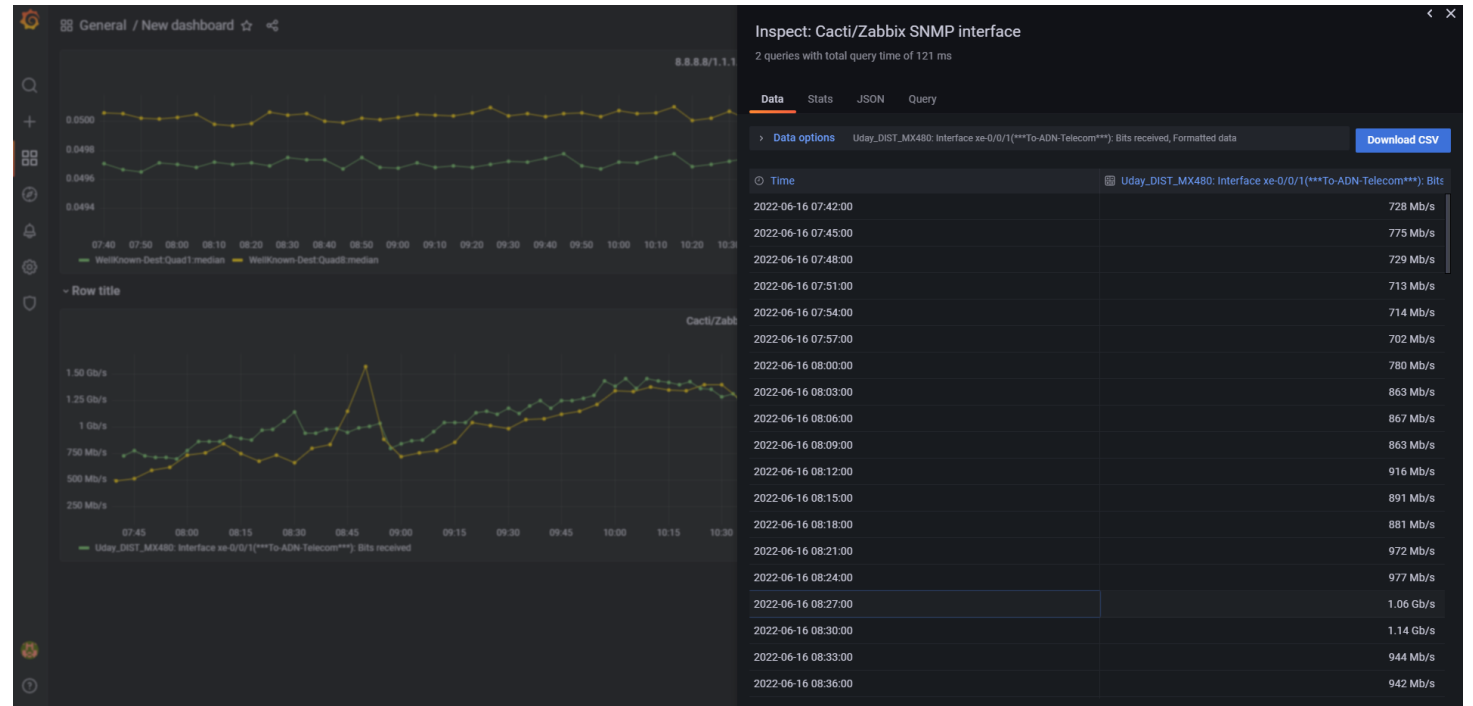
Little more!!!

Not only Graphs but Features can also be combined in a single dashboard

CASE-2: Objective/ Goal

An alternative approach to view Mikrotik PPPoE interface historical usage using Zabbix & Grafana

Hardware & Software used for Retail network

Mikrotik Routers [CCR Series Mostly]

In-house developed OSS/BSS software billing & client management

Zabbix Server

Grafana Server

Operation Challenges

Widely used features like these

1. PPPoE users' session status [Connection/Disconnection/Failed]
2. Usage history

Are missing

To resolve Problem-2 [Usage history]

1. Enabled SNMP at Mikrotik NAS/BRAS

2. Installed Zabbix Server and Start probing Mikrotik interfaces using built-in Mikrotik templates

Configuration challenges

As PPPoE interfaces are created dynamically, their SNMP index values keep changing every time session starts. So its impossible to generate static graphs using rrd number.

Interface <pppoe-84282>(): Bits received ?	11h 51m 3s	24 bps	-1.79 Kbps	component: network	description	interface: <pppoe-84282>	Graph	i
Interface <pppoe-84282>(): Bits received ?	59s	34.64 Kbps	-30.47 Kbps	component: network	description	interface: <pppoe-84282>	Graph	
Interface <pppoe-84282>(): Bits sent ?	11h 50m 58s	48 bps	-608 bps	component: network	description	interface: <pppoe-84282>	Graph	i
Interface <pppoe-84282>(): Bits sent ?	55s	576.46 Kbps	-1.01 Mbps	component: network	description	interface: <pppoe-84282>	Graph	
Interface <pppoe-84282>(): Interface type ?	23h 30m 34s	ppp (23)		component: network	description	interface: <pppoe-84282>	Graph	i
Interface <pppoe-84282>(): Interface type ?	10h 30m 57s	ppp (23)		component: network	description	interface: <pppoe-84282>	Graph	
Interface <pppoe-84282>(): Operational status ?	11h 51m 3s	up (1)		component: network	description	interface: <pppoe-84282>	Graph	i
Interface <pppoe-84282>(): Operational status ?	1m 4s	up (1)		component: network	description	interface: <pppoe-84282>	Graph	

Workaround

At this point we found Grafana provides excellent way to generate the time series graph which we are looking for
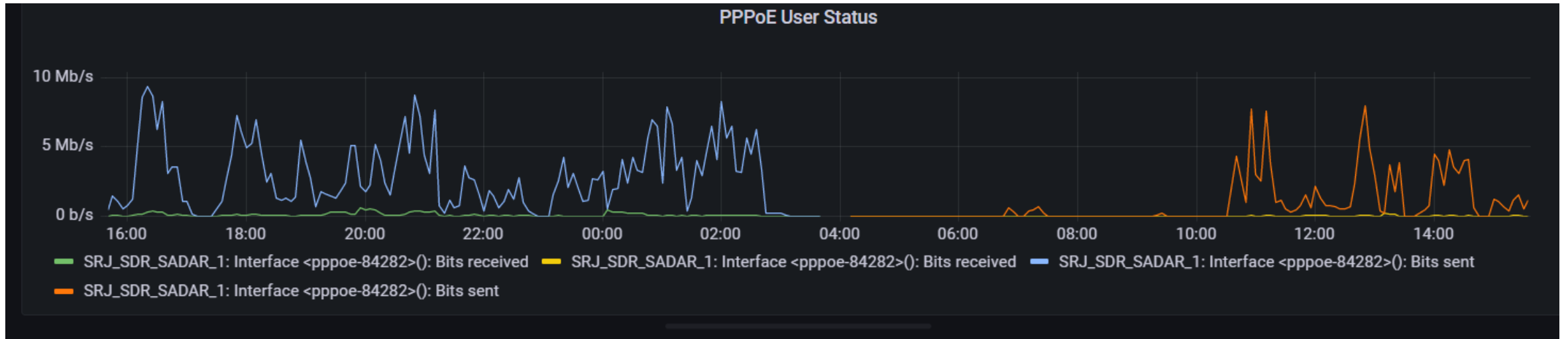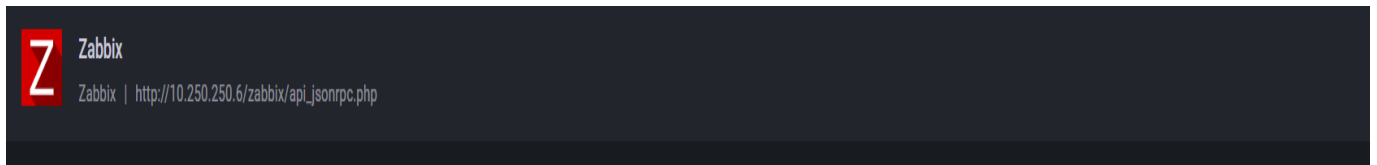
Setup scenario
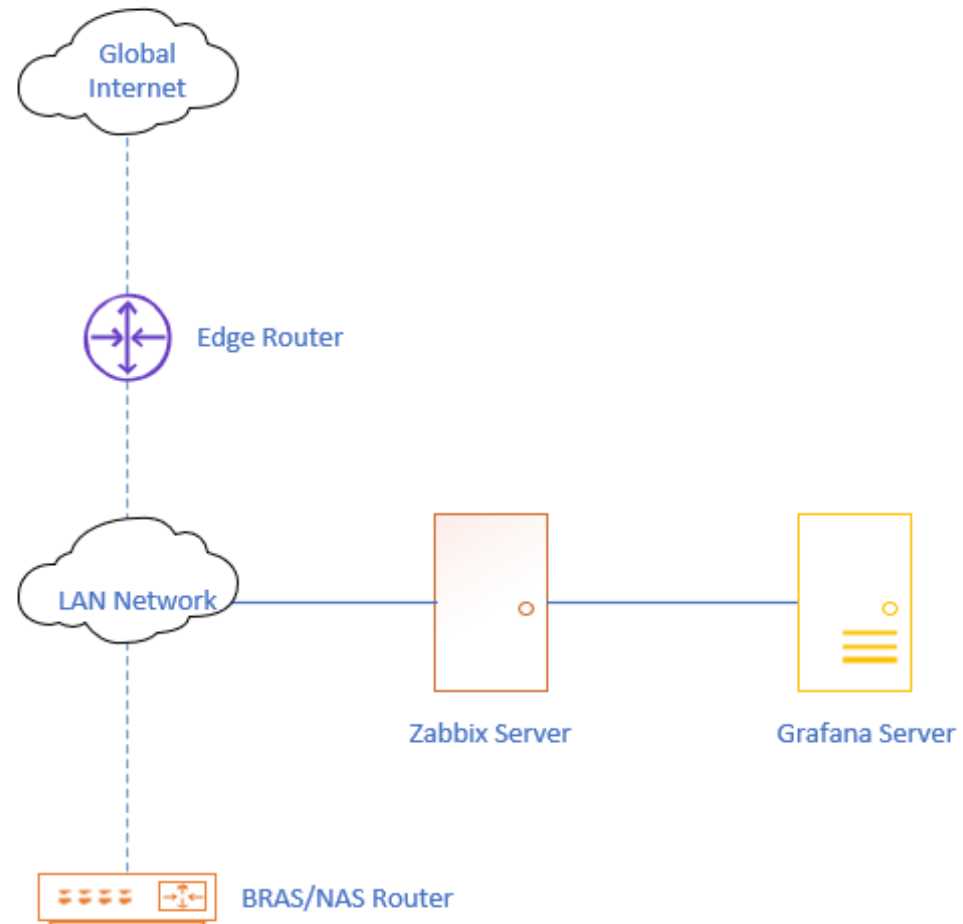
Data Source & Template used for this

- Zabbix:
  - Buiiltin Mikrotik Templates

- Grafana:
  - Zabbix App

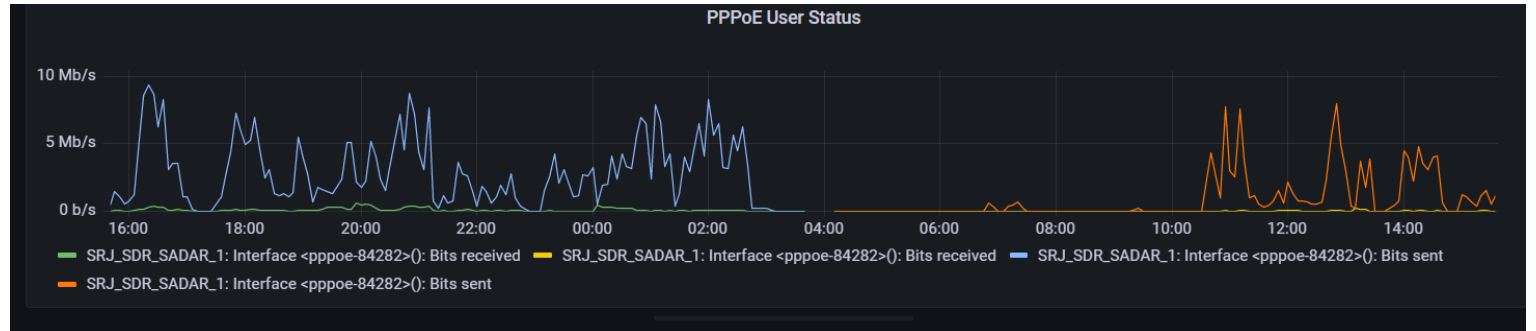# Grafana Query

Inspect: PPPoE User Status
2 queries with total query time of 1 s

Data   Stats   JSON   **Query**

▼request: Object
  url: "api/ds/query"
  method: "POST"
  ▼data: Object
    ▼queries: Array[2]
      ▼0: Object
        ►application: Object
        ►datasource: Object
        ►functions: Array[0]
        ►group: Object
        hide: false
        ▼host: Object
          filter: "SRJ_SDR_SADAR_1"
        ▼item: Object
          filter: "Interface <pppoe-84282>(): Bits received"
        ▼itemTag: Object
          filter: "interface: <pppoe-84282>"
        ►options: Object
        ►proxy: Object
        queryType: "0"
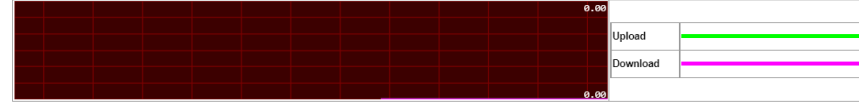   refId: "A"
        resultFormat: "time_series"
        ►table: Object

    ▼1: Object
      ►application: Object
      ►datasource: Object
      ►functions: Array[0]
      ►group: Object
      hide: false
      ▼host: Object
        filter: "SRJ_SDR_SADAR_1"
      ▼item: Object
        filter: "Interface <pppoe-84282>(): Bits sent"
      ▼itemTag: Object
        filter: "interface: <pppoe-84282>"
      ►options: Object
      ►proxy: Object
      queryType: "0"
      refId: "B"
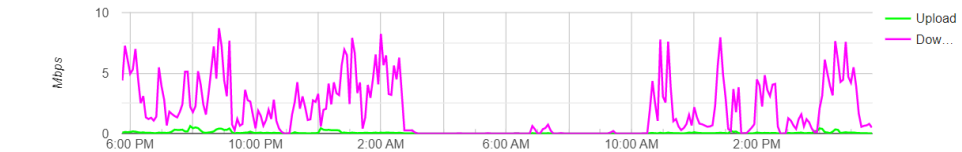   resultFormat: "time_series"

Outcome/Result at Grafana Dashboard

Result for Udoy OSS/BSS

Questions & Answers

Thank You