Grafana-Powered Traffic Dashboard for Customer Networks

Network engineers:
Munkhtulga.B
Khuslen.B

AGENDA

- About us
- Why we did this project
- Monitoring tools we use
- Finding the right solution
- Grafana
- Monitoring system design
- Building a unified dashboard
- Project challenges
- Conclusion

ABOUT US

We provide the following services for broadband, and corporate users:

- IPTV
- Internet
- VoIP
- OTT

Why we did this project

One of our customers requested a solution to monitor their bandwidth consumption, with a particular focus on distinguishing between local (Mongolian) and Internet traffic, as well as identifying their top bandwidth users.

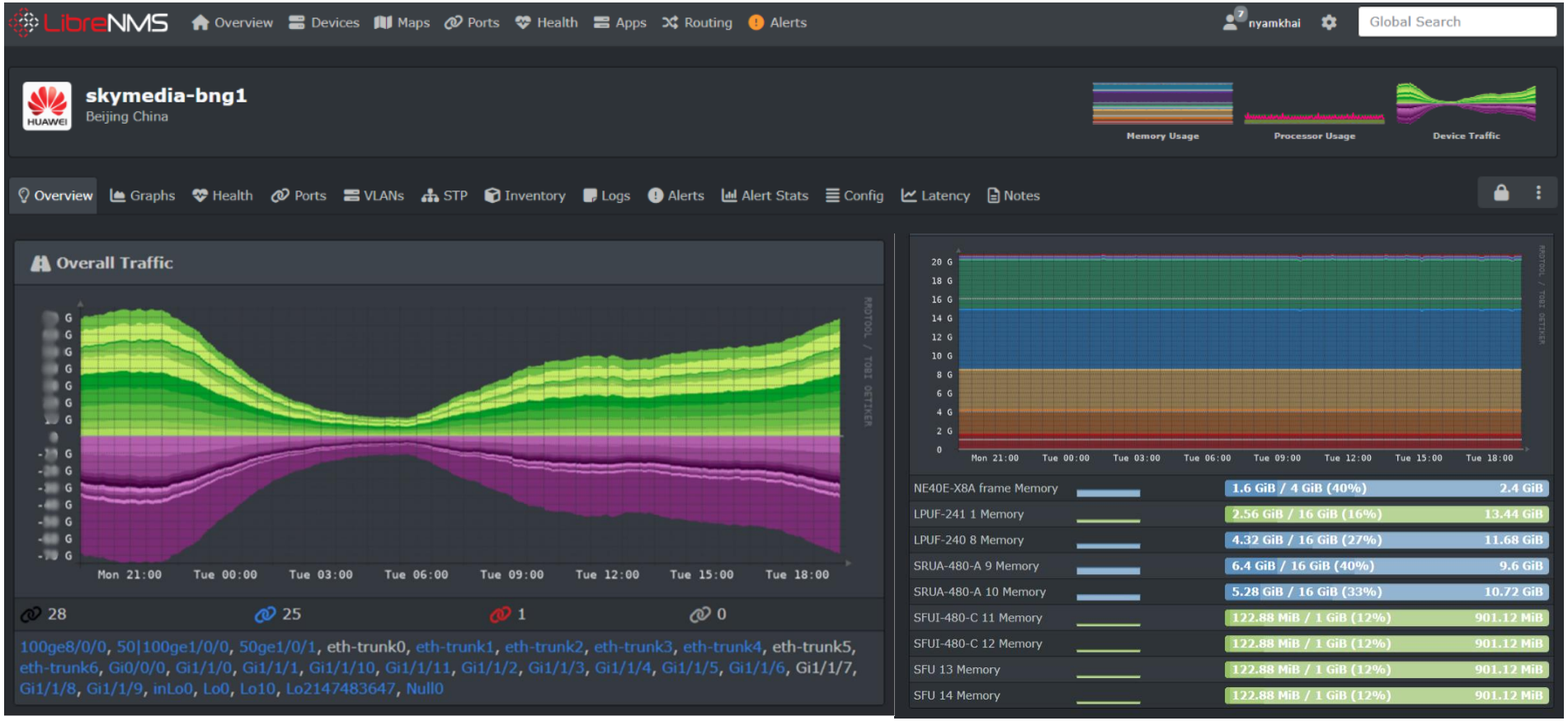
Bandwidth usage by:

- Total traffic
- Internet traffic
- Mongolian traffic - MIX, MISPA
- Top Source IPs
- Top Source ASs

Traffic for our corporate customers are monitored by Akvorado and LibreNMS.

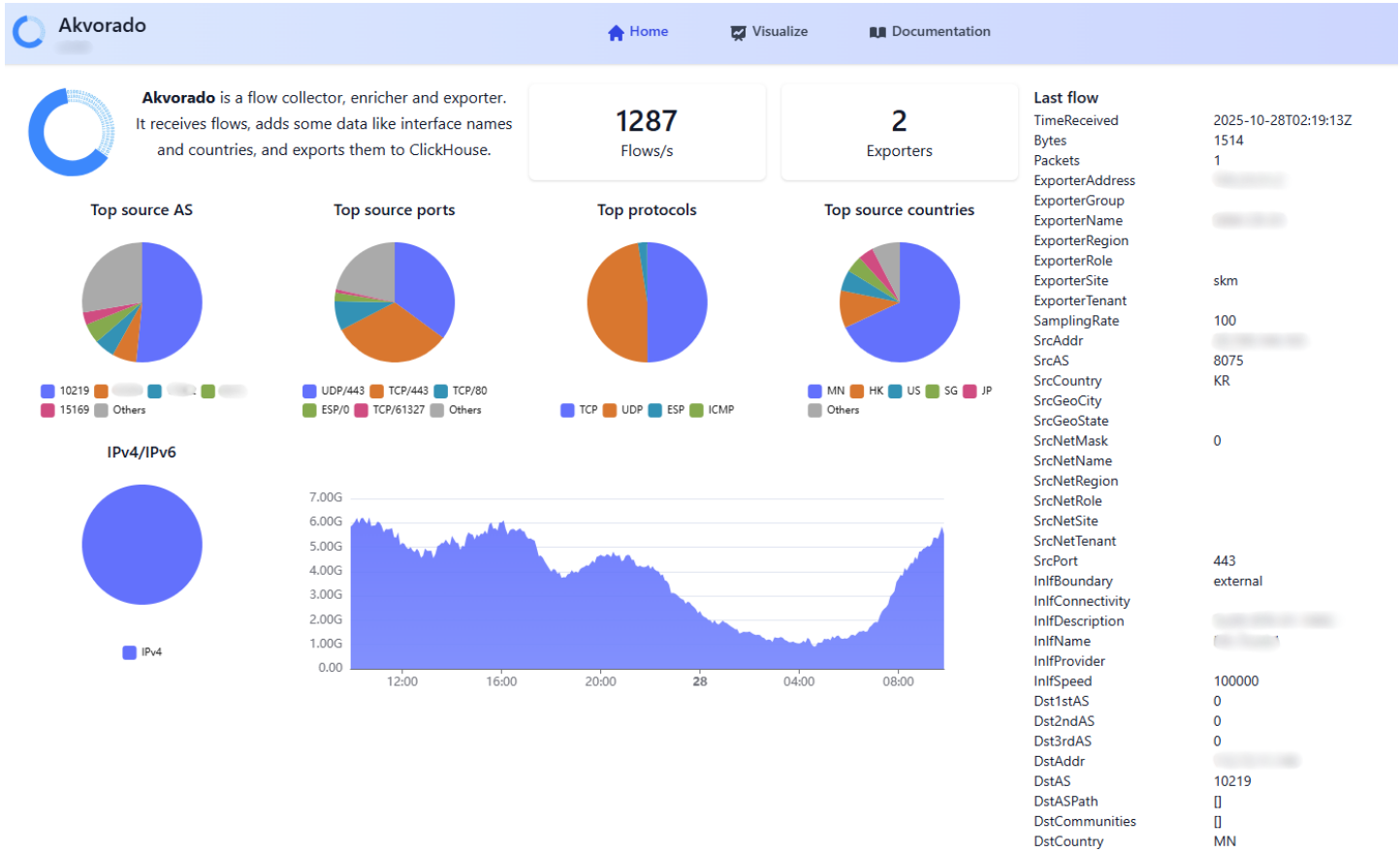
Monitoring tools we use

We use LibreNMS to monitor hardware status, including device health and availability, as well as to track and analyze interface traffic across the network.



Monitoring tools we use

We use Akvorado to monitor network traffic in detail, showing information about packets, bandwidth usage, and flow patterns. This helps us analyze network performance and to identify anomalies.

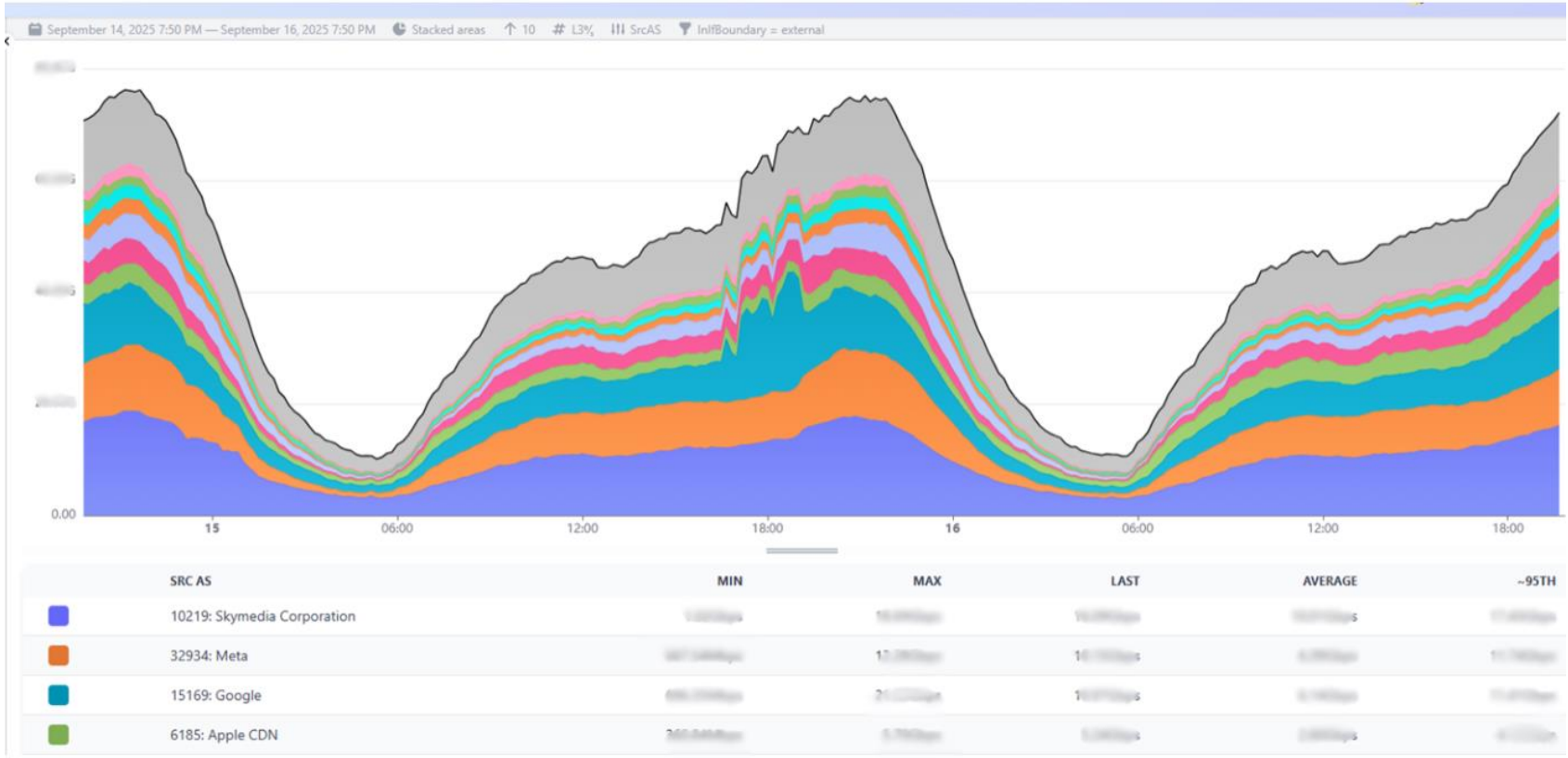


Last flow

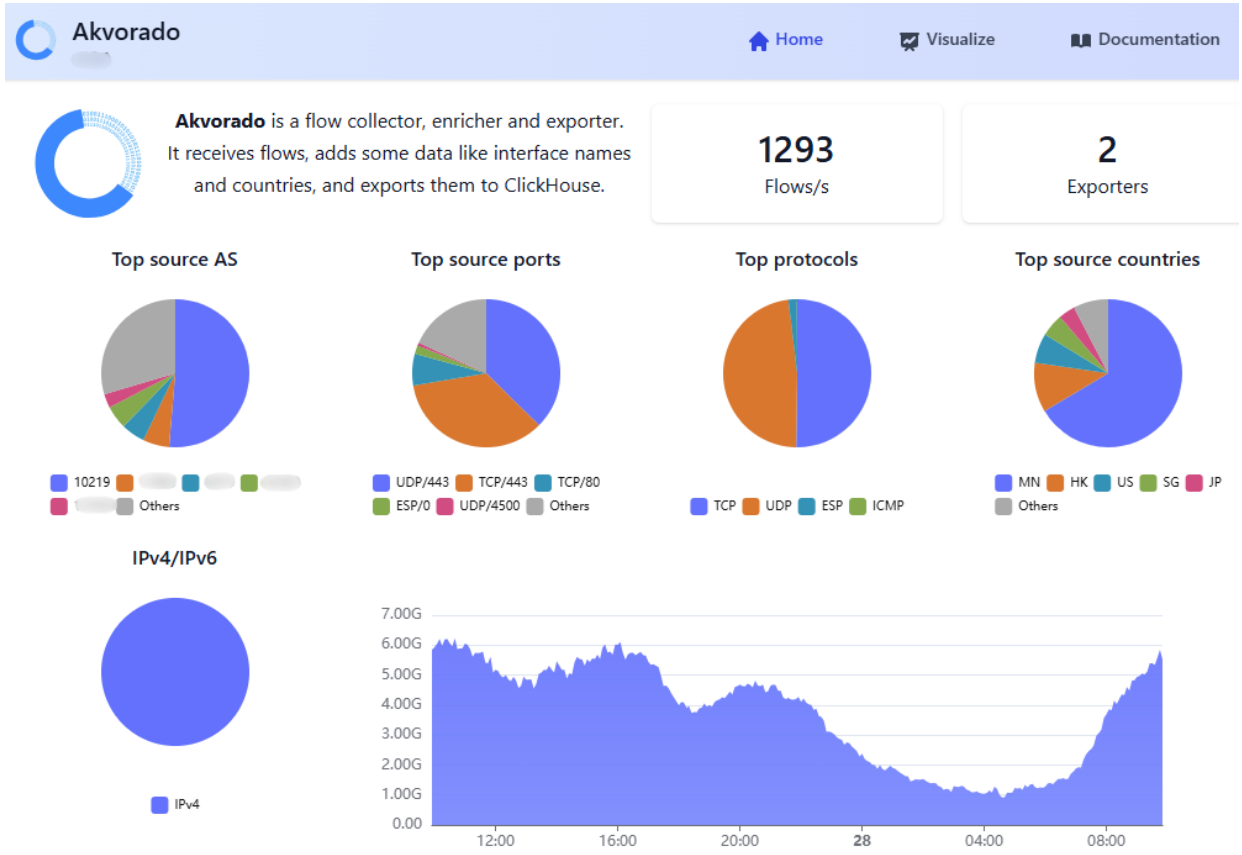
TimeReceived	2025-10-28T02:19:13Z
Bytes	1514
Packets	1
ExporterAddress	
ExporterGroup	
ExporterName	
ExporterRegion	
ExporterRole	
ExporterSite	skm
ExporterTenant	
SamplingRate	100
SrcAddr	
SrcAS	8075
SrcCountry	KR
SrcGeoCity	
SrcGeoState	
SrcNetMask	0
SrcNetName	
SrcNetRegion	
SrcNetRole	
SrcNetSite	
SrcNetTenant	
SrcPort	443
InIfBoundary	external
InIfConnectivity	
InIfDescription	
InIfName	
InIfProvider	
InIfSpeed	100000
Dst1stAS	0
Dst2ndAS	0
Dst3rdAS	0
DstAddr	
DstAS	10219
DstASPath	[]
DstCommunities	[]
DstCountry	MN

Monitoring tools we use

We use Akvorado to monitor network traffic in detail, showing information about packets, bandwidth usage, and flow patterns. This helps us analyze network performance and to identify anomalies.

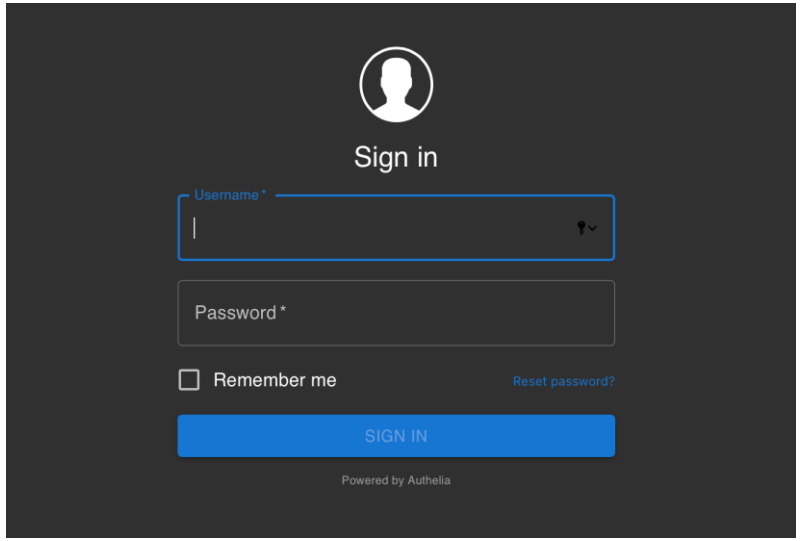


Finding the right solution



Problem:

Akvorado does not currently support RBAC (Role-Based Access Control). As a result, any user created would have access to view all flows, rather than being restricted to only their own.



Finding the right solution

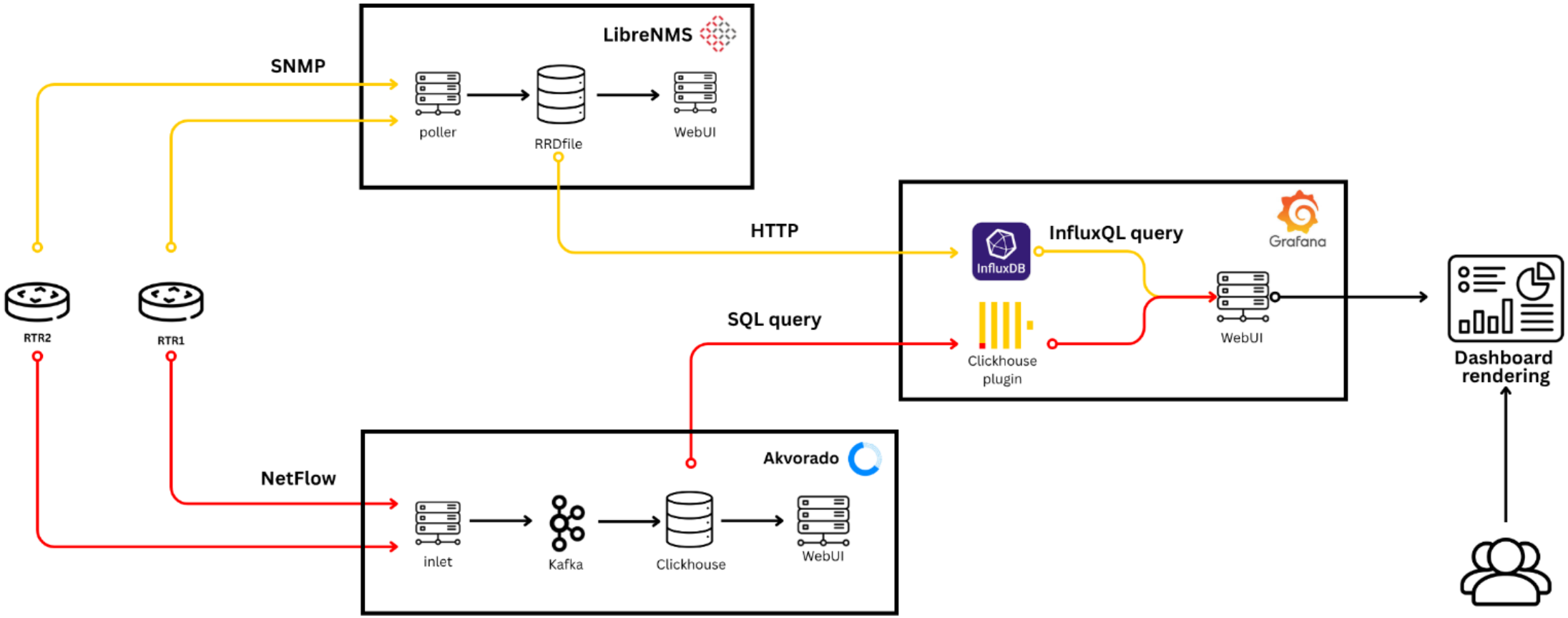
From the customers' perspective, accessing two separate monitoring tools can be complex, requiring additional effort such as writing filters to view their specific flows. Additionally, since Akvorado does not support RBAC, as users would potentially have unrestricted access to all network flows.

Therefore, we have chosen Grafana to create a unified dashboard that integrates information from both LibreNMS and Akvorado, offering a simplified, centralized, and secure monitoring solution.

Grafana

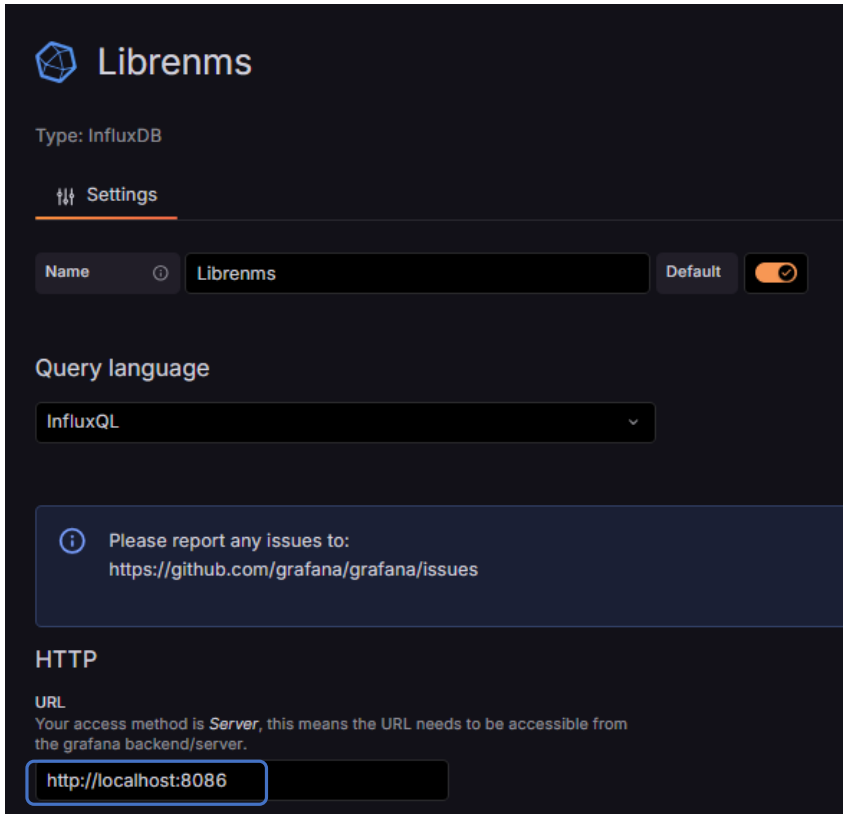
Grafana is an open-source monitoring platform that integrates with various data sources and provides customizable dashboards for real-time analysis of metrics and system performance.

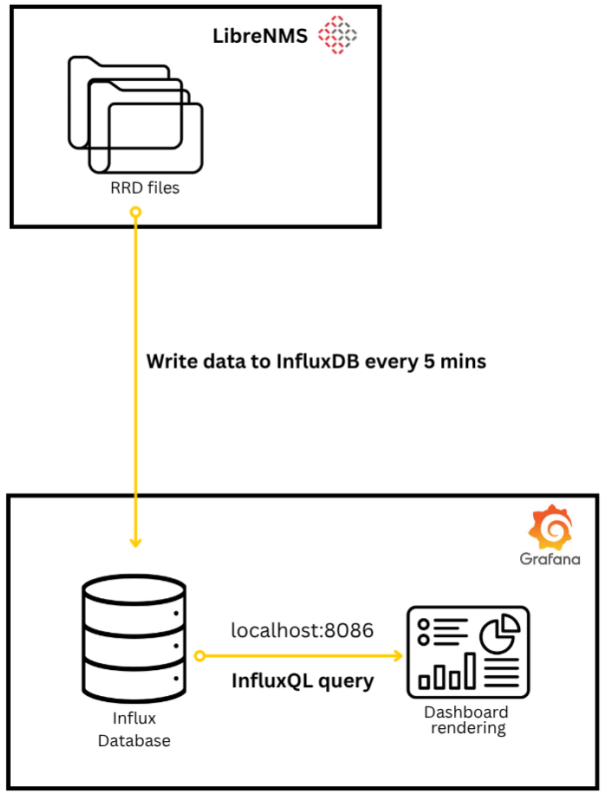
Monitoring System Design



The Grafana server uses two data sources for different types of graphs

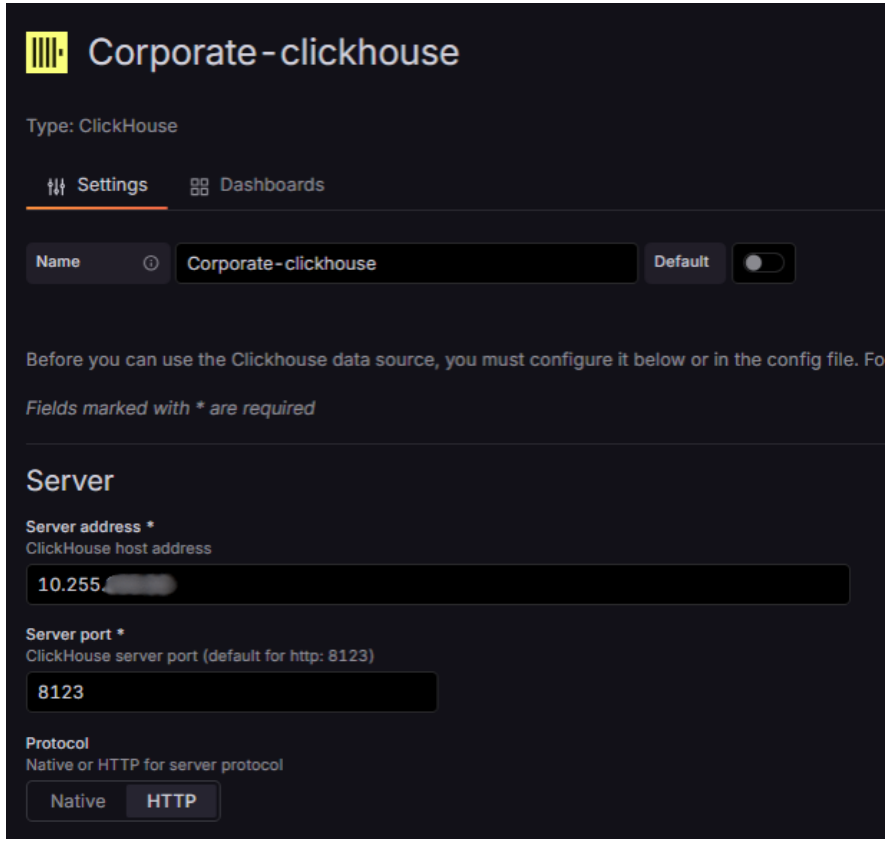
Monitoring System Design - LibreNMS

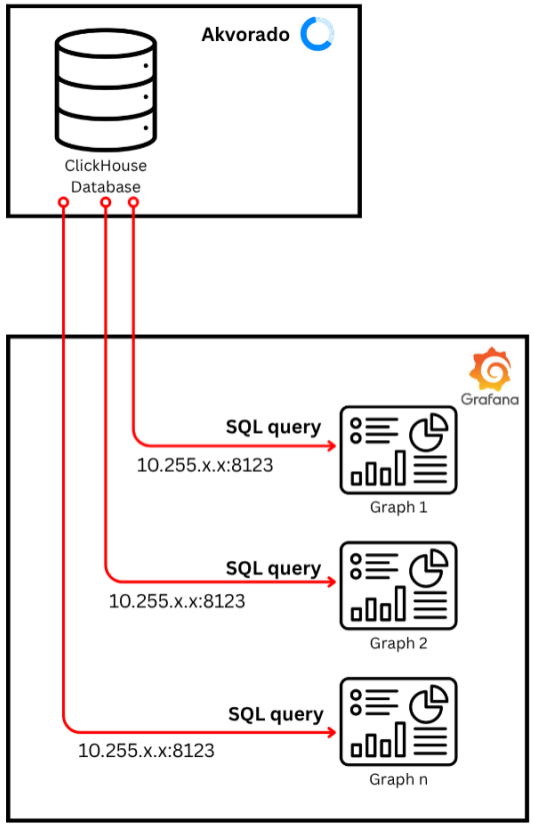




For device metrics, LibreNMS writes new data to InfluxDB every 5 minutes. Grafana fetches this data directly from InfluxDB on the same server, which ensures that rendering these graphs is fast and independent of the LibreNMS server.

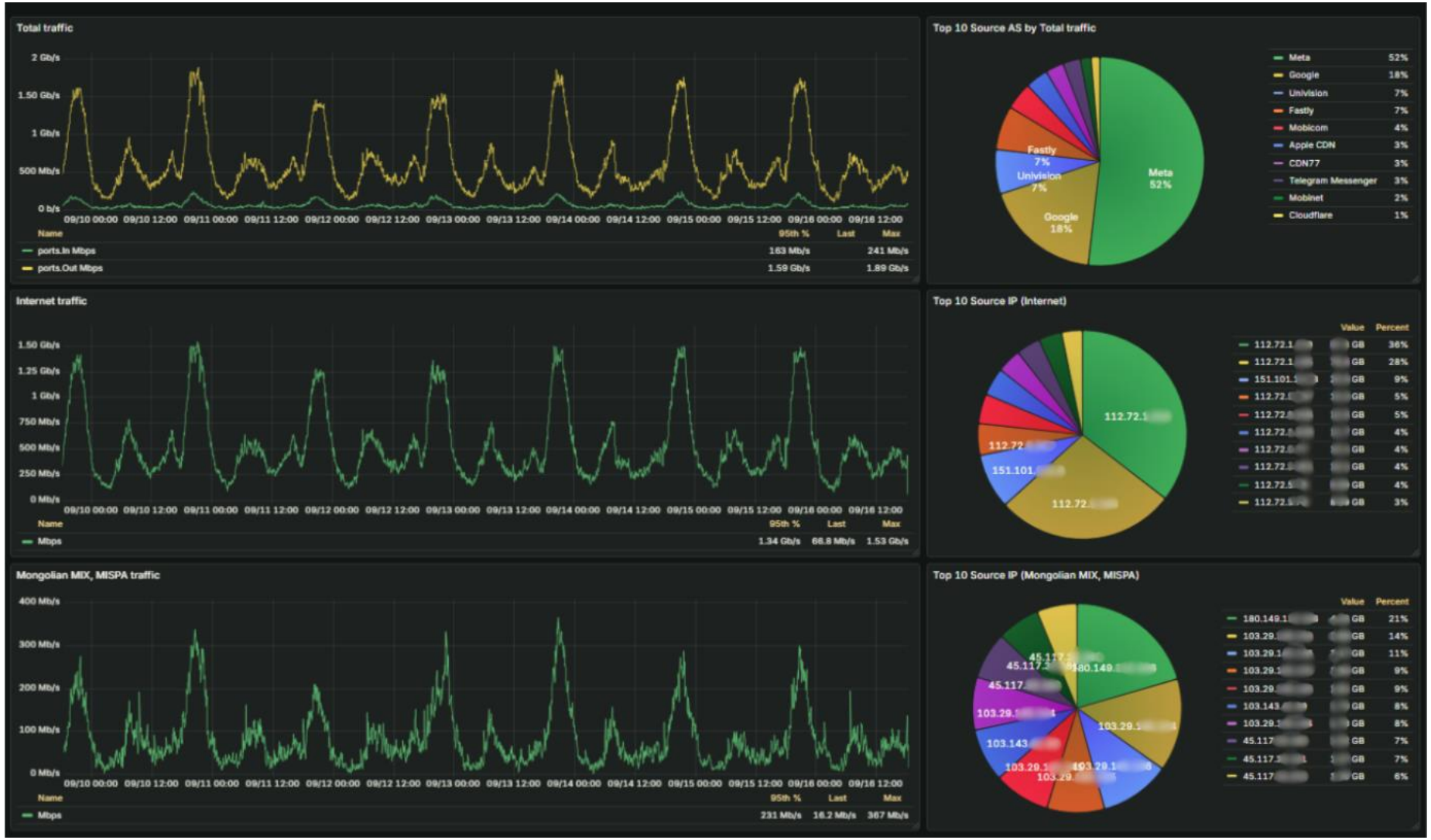
Monitoring System Design - Akvorado





For flow-related graphs, Grafana queries the ClickHouse database running on the Akvorado server. Each graph requires a separate query, so displaying multiple graphs results in multiple queries to Akvorado. If the Akvorado server goes down, most of the flow graphs on the Grafana dashboard will become unavailable.

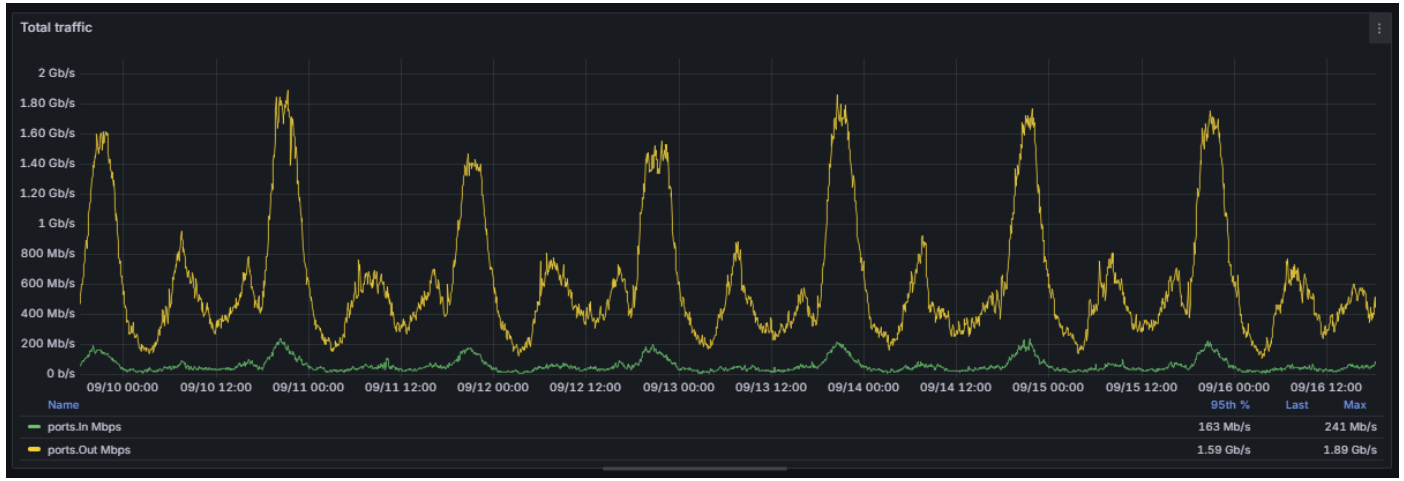
Building a Grafana dashboard



Building a Grafana dashboard - InfluxDB

InfluxQL query for visualizing Total traffic

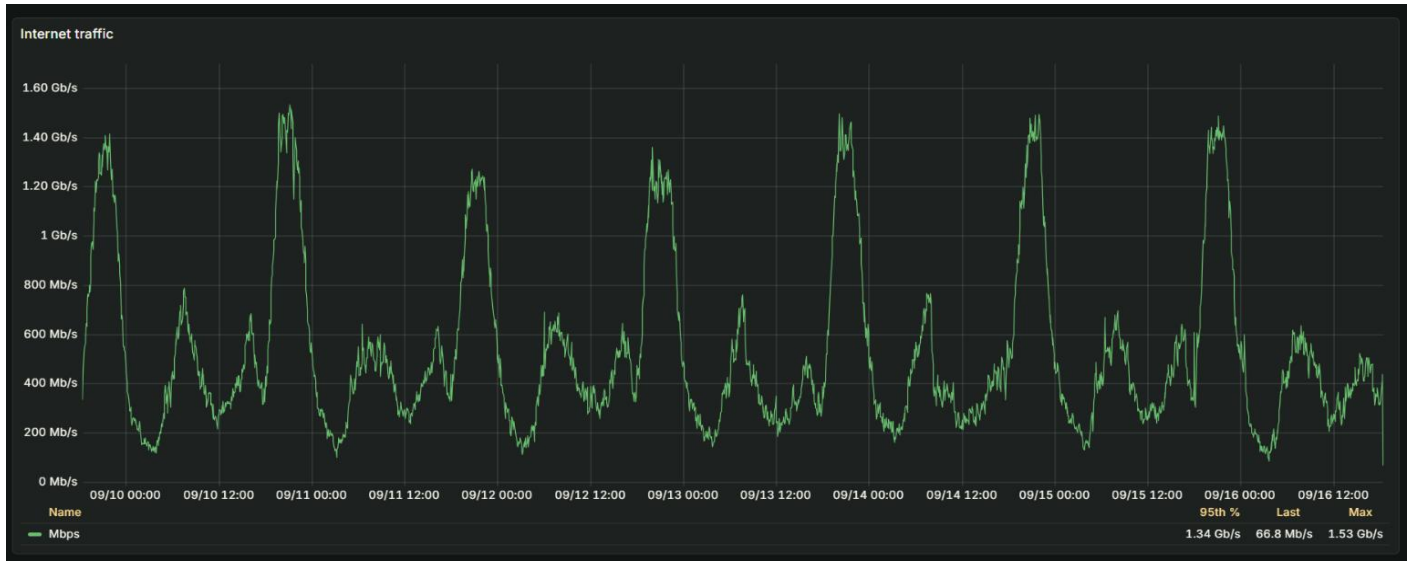
```
SELECT
  sum("in_val") AS "In Mbps",
  sum("out_val") AS "Out Mbps"
FROM (
  SELECT mean("ifInBits_rate") AS "in_val",
                mean("ifOutBits_rate") AS "out_val"
  FROM "ports"
  WHERE ("hostname" = '[redacted]' OR "hostname" = '[redacted]')
    AND "ifName" = 'Eth-Trunk-Y'
    AND $timeFilter
  GROUP BY time(4m), "hostname" fill(null)
)
GROUP BY time(4m) fill(null)
```



Building a Grafana dashboard - ClickHouse plugin

SQL query for visualizing Internet traffic

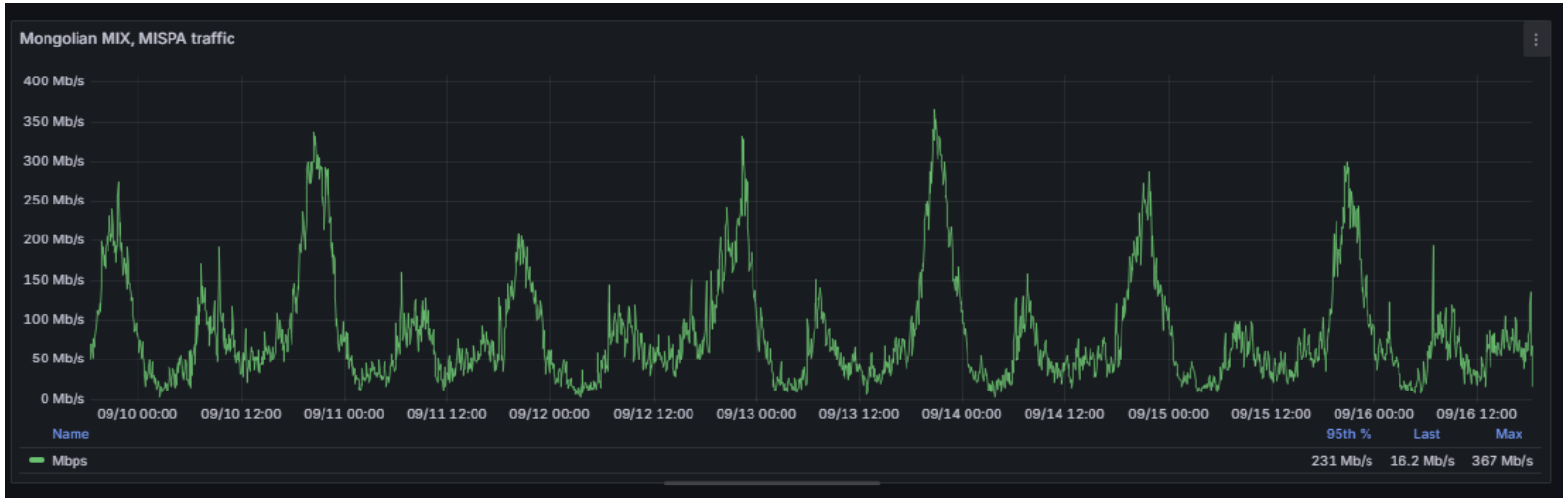
```sql
SELECT
  toStartOfInterval(TimeReceived, INTERVAL 4 minute) AS ts,
  round(sum(Bytes) * 8 / 1000/ 1000/ 1000/ 240, 2) AS Mbps
FROM flows_1m0s
WHERE SrcAS NOT IN (Mongolian_AS_number1, Mongolian_AS_number2,...)
  AND DstAddr BETWEEN toIPv6('::ffff:x.x.x.x') AND toIPv6('::ffff:x.x.x.x')
  AND $__timeFilter(TimeReceived)
GROUP BY ts
ORDER BY ts;
```



Building a Grafana dashboard - ClickHouse plugin

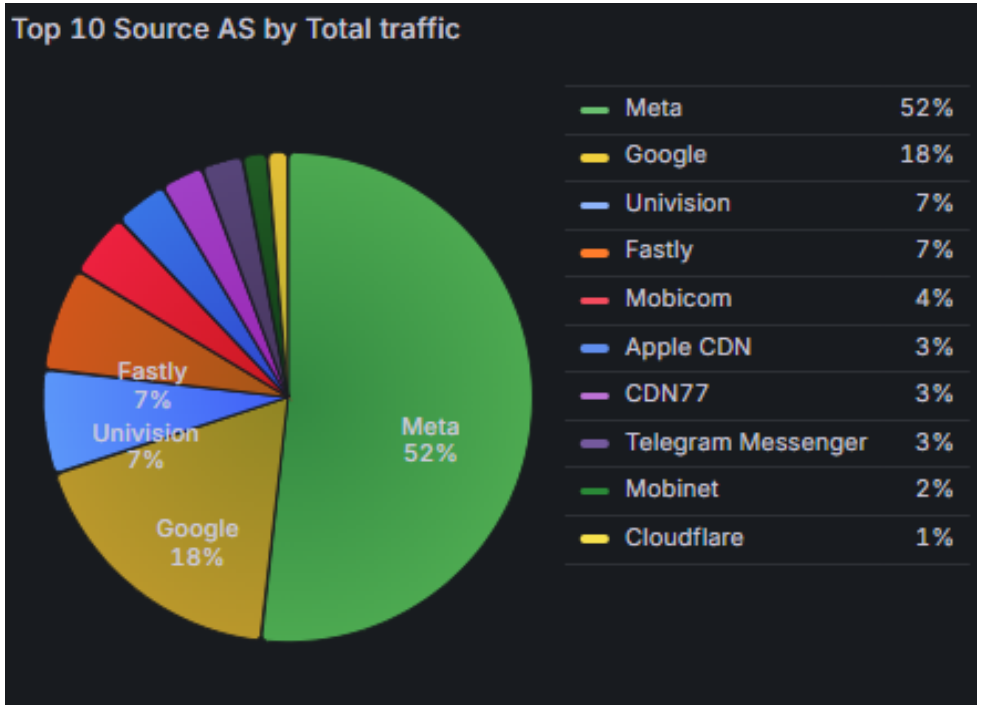
SQL query for visualizing Local traffic

```
SELECT
  toStartOfInterval(TimeReceived, INTERVAL 4 minute) AS ts,
  round(sum(Bytes) * 8 / 1000/ 1000/ 1000/ 240, 2) AS Mbps
FROM flows WHERE SrcAS IN (Mongolian_AS_number1, Mongolian_AS_number2,...)
  AND DstAddr BETWEEN toIPv6('::ffff:x.x.x.x') AND toIPv6('::ffff:x.x.x.x')
  AND $__timeFilter(TimeReceived)
GROUP BY ts
ORDER BY ts;
```



Building a Grafana dashboard - ClickHouse plugin

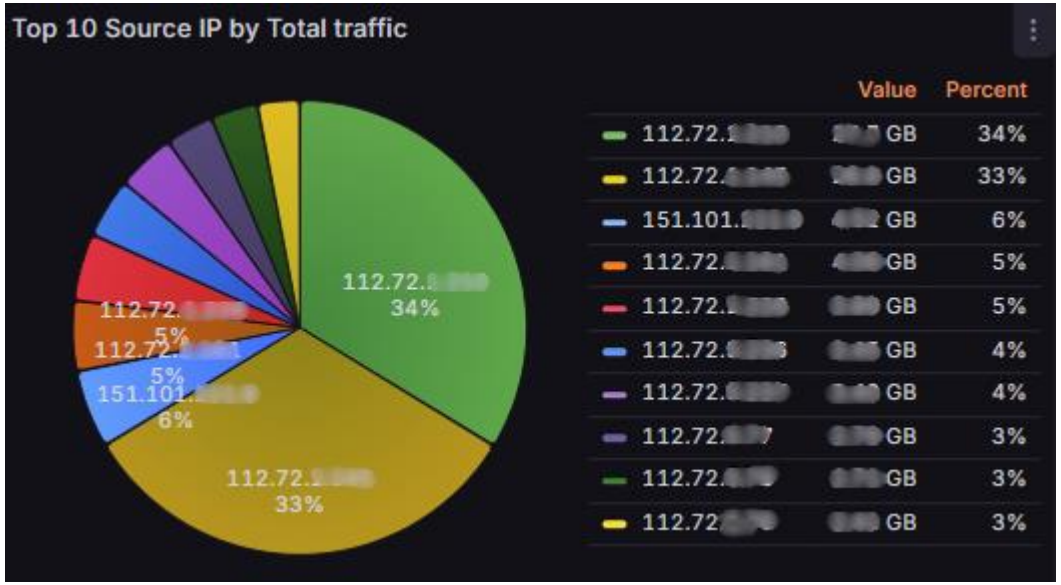
SQL query for visualizing Top 10 Source AS




```
SELECT
  CASE
    WHEN SrcAddr >= '[redacted]' AND SrcAddr <= '[redacted]' THEN 'Google'
    WHEN SrcAddr >= '[redacted]' AND SrcAddr <= '[redacted]' THEN 'Google'
    WHEN SrcAddr >= '[redacted]' AND SrcAddr <= '[redacted]' THEN 'Akamai'
    WHEN SrcAddr >= '[redacted]' AND SrcAddr <= '[redacted]' THEN 'Meta'
    WHEN SrcAddr >= '[redacted]' AND SrcAddr <= '[redacted]' THEN 'Oracle'
    WHEN SrcAS = 32934 THEN 'Meta'
    WHEN SrcAS = 15169 THEN 'Google'
    WHEN SrcAS = 8075 THEN 'Microsoft'
    WHEN SrcAS = 6185 THEN 'Apple CDN'
    WHEN SrcAS = 54113 THEN 'Fastly'
    WHEN SrcAS = 17882 THEN 'Univision'
    WHEN SrcAS = 62014 THEN 'Telegram Messenger'
    WHEN SrcAS = 55805 THEN 'Mobicom'
    WHEN SrcAS = 9484 THEN 'Mobinet'
    WHEN SrcAS = 13335 THEN 'Cloudflare'
    WHEN SrcAS = 60068 THEN 'CDN77'
    WHEN SrcAS = 396982 THEN 'Google Cloud'
    ELSE CONCAT('Other-', SrcAddr)
  END AS service,
  SUM(Bytes) / 1000 / 1000 / 1000 AS Traffic_GB
FROM flows
WHERE DstAddr BETWEEN toIPv6('::ffff:[redacted]') AND toIPv6('::ffff:[redacted]')
  AND $__timeFilter(TimeReceived)
GROUP BY service
ORDER BY Traffic_GB DESC
LIMIT 10;
```

Building a Grafana dashboard - ClickHouse plugin

SQL query for visualizing Top 10 Source IP




```sql
SELECT
  SrcAddr AS src_ip,
  sum(Bytes) / 1000 / 1000 / 1000 AS Traffic_MB
FROM flows
WHERE SrcAS NOT IN (Mongolian_AS_number1, Mongolian_AS_number2,..)
  AND DstAddr BETWEEN toIPv6('::ffff:x.x.x.x') AND toIPv6('::ffff:x.x.x.x')
  AND $__timeFilter(TimeReceived)
GROUP BY src_ip
ORDER BY Traffic_MB DESC
LIMIT 10;
```

Project challenges - 1

Akvorado's setup stored raw data for too long, causing slow loading of long-term graphs. When we tried to adjust the retention settings, the service stopped working. To resolve this, we deployed a new server with the same hardware specifications and installed Akvorado using configurations that fit our needs. This resulted in noticeably better performance.

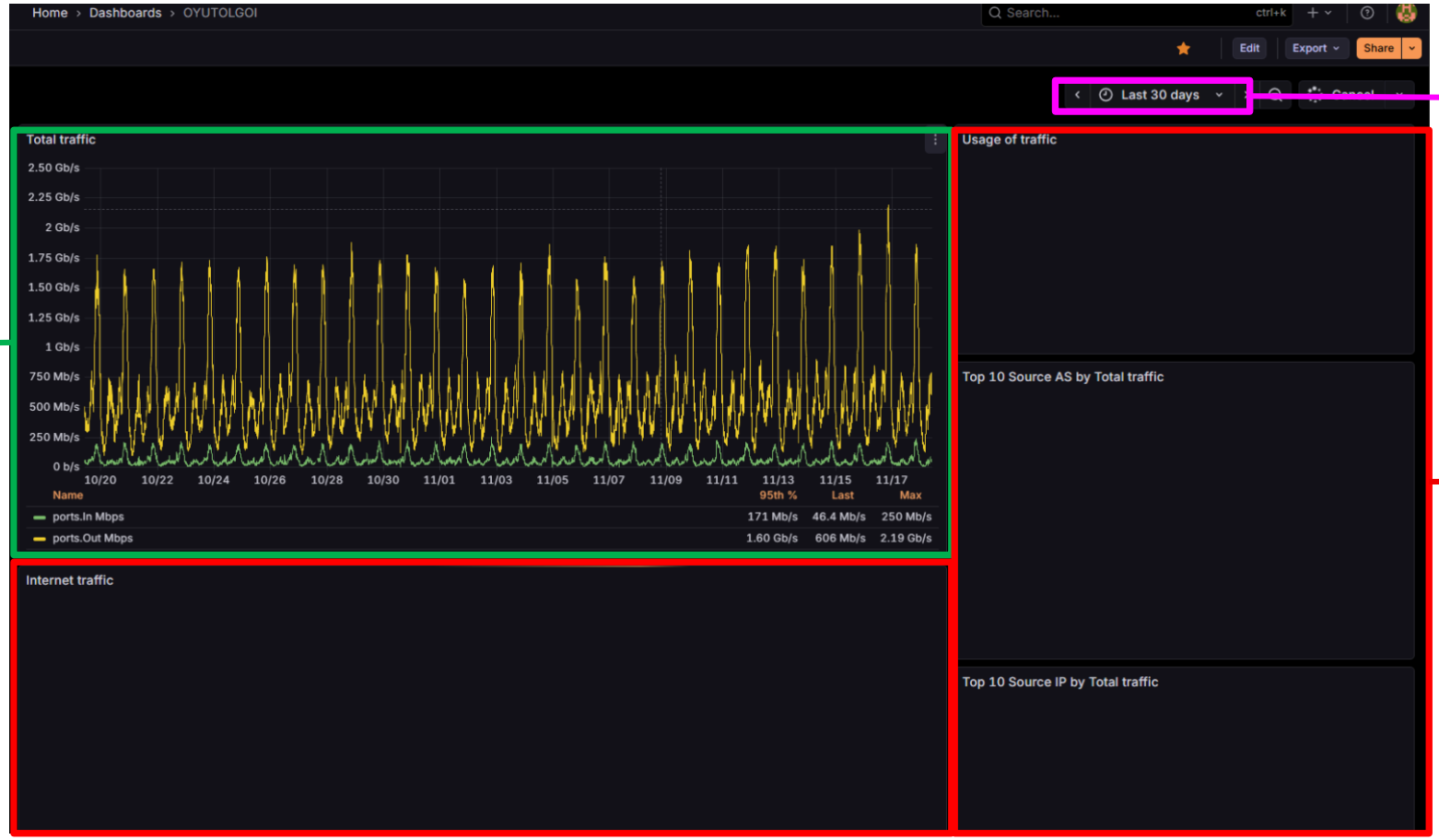
	Previous akvorado	Current akvorado
kafka brokers	1	3
kafka topic config /num partition/	8	12
core workers	6	24
flow input workers	6	6
clickhouse - kafka consumers	4	8

Table 1. Akvorado config

Resolution step	Previous akvorado	Current akvorado
raw data	31 days	1 day
1 min	none	35 days
5 min	90 days	90 days
1 hour	1 year	1 year

Table 2. Data retention config of Clickhouse

Project challenges - 2



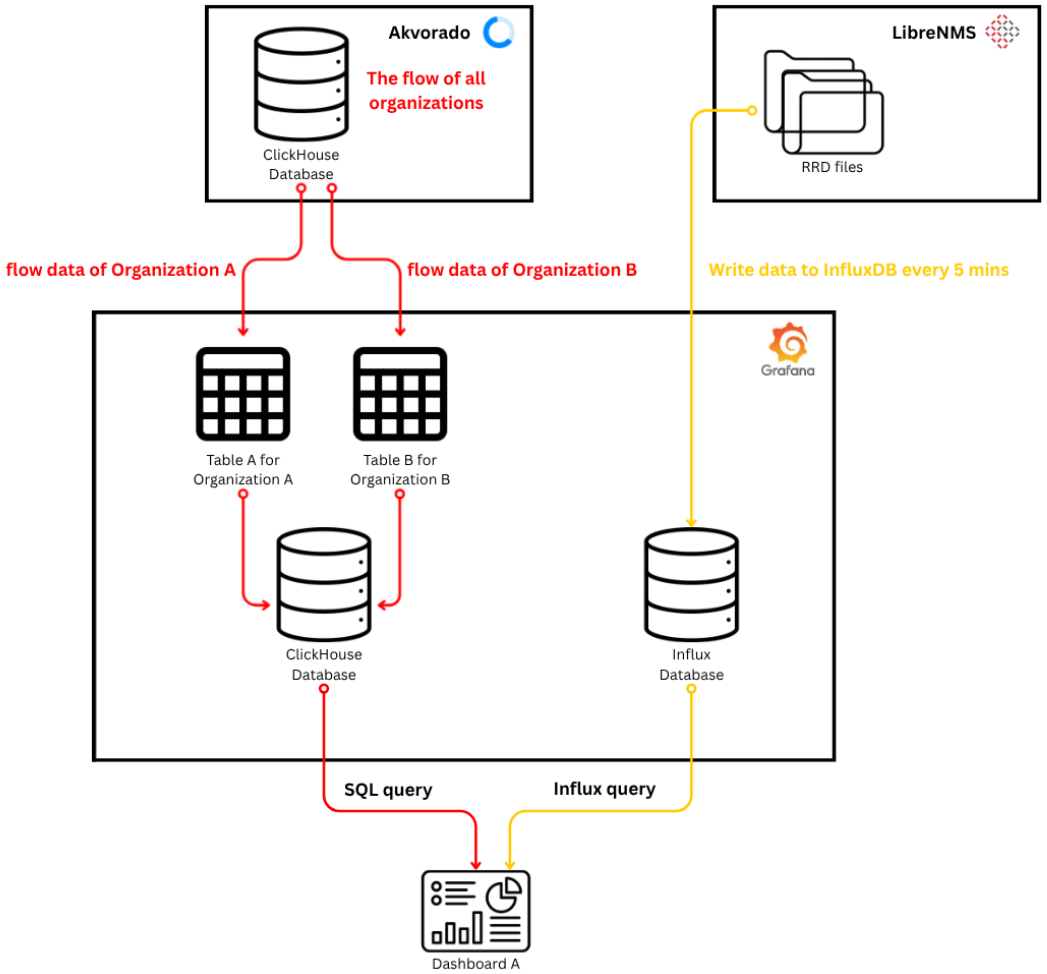
For the organization dashboard, there are seven graphs, each requiring its own query. One of the queries comes from LibreNMS, while the others are from Akvorado. This clearly demonstrates the load difference between a table-based database and a time-series database.

Project challenges - 2

Since ClickHouse is a table-based database and InfluxDB is a time-series database, and because time-series databases generally operate faster and with lower system load, we determined that using InfluxDB would be the better approach.

For this reason, we decided to export Akvorado flow data from ClickHouse to InfluxDB using Telegraf. Although InfluxDB successfully received the data, some fields were not exported in the correct format, which caused several Grafana queries to fail.

Project challenges - 2



If we were to set this up again, we would create a second ClickHouse database specifically to store data for organizations that want to monitor their bandwidth usage. By separating this data, we could reduce the overall storage requirements and ensure that the graphs load much faster, providing a smoother and more efficient experience for users

Conclusion

In conclusion, by implementing Grafana, we have successfully consolidated the monitoring of LibreNMS and Akvorado into a single, unified dashboard. Even though we still have some improvements to make, this solution simplifies access for customers, enhances security by mitigating unrestricted access issues, and provides a centralized platform for efficient monitoring of network flows and system performance.