Infrastructure monitoring

Marian Rychtecký
& NIX.CZ Dev team

NIX·CZ

NX OS DME

What you get ?

- Speed
  - Requests takes milliseconds (full switch setup ~5s)
  - Individual requests (interface, VLAN, VNI, BGP settings) ~100ms
  - Reliability of the REST API

- **Operational parameters**

NX OS DME
(templates translating)

```
snmp-server contact email@domain.cz
snmp-server location Site 1, Prague, CZ
```

POST /api/mo/sys/snmp/inst.json

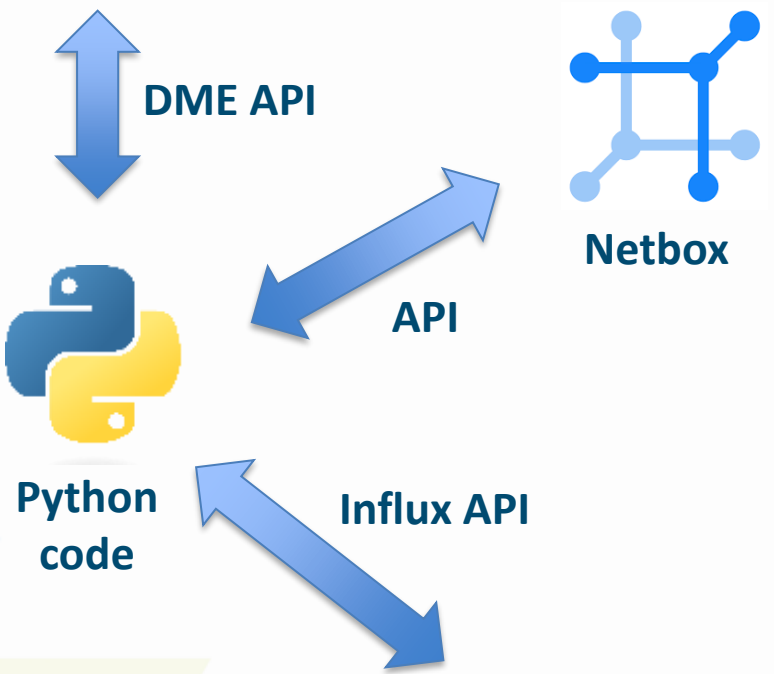
```
{
   "snmpInst": {
      "children": [
         {
            "snmpSysInfo": {
               "attributes": {
                  "sysContact": "email@domain.cz",
                  "sysLocation": "Site 1, Prague, CZ"
               }
            }
         }
      ]
   }
}
```

Collecting objects


```
"rmonIfIn": {
                "attributes": {
                    "broadcastPkts": "3",
                    "clearTs": "never",
                    "discards": "0",
                    "dn": "sys/intf/phys-[eth1/5]/dbgIfIn",
                    "errors": "0",
                    "modTs": "2023-07-11T09:44:04.692+00:00",
                    "multicastPkts": "1995153",
                    "nUcastPkts": "1995156",
                    "noBuffer": "0",
                    "octetRate": "4947614883",
                    "octets": "606732716440771 86",
                    "packetRate": "5404507",
                    "rateInterval": "300",
                    "ucastPkts": "63082941837541",
                    "unknownEtype": "0",
                    "unknownProtos": "0"
                }
            }
        }
```

Statistics in the real world

Network device (Cisco Nexus 9300)



flux™

Traffic weather map

External Graphs

Internal Graphs

influxdb open source™

Collecting objects

- Collecting data every 30s
- Data pre-processing (calculated items)
- Saving data to TSDB

Scale

We collect

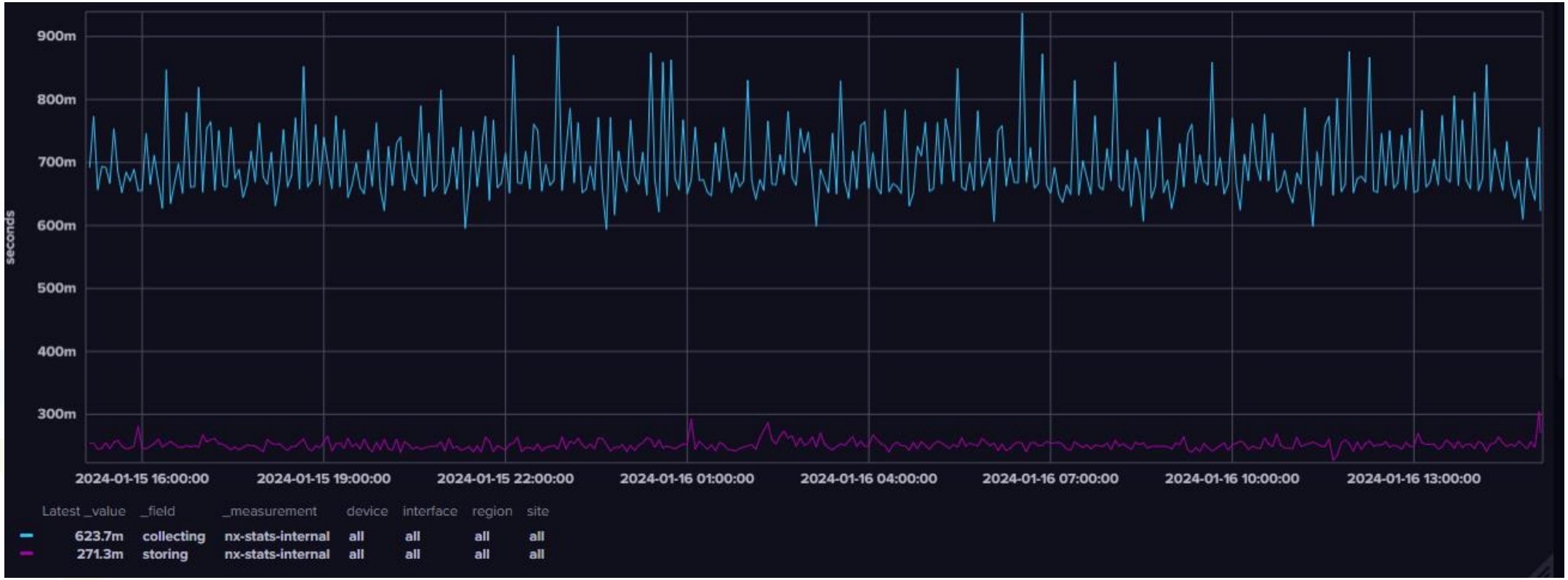
37 devices

2339 interfaces

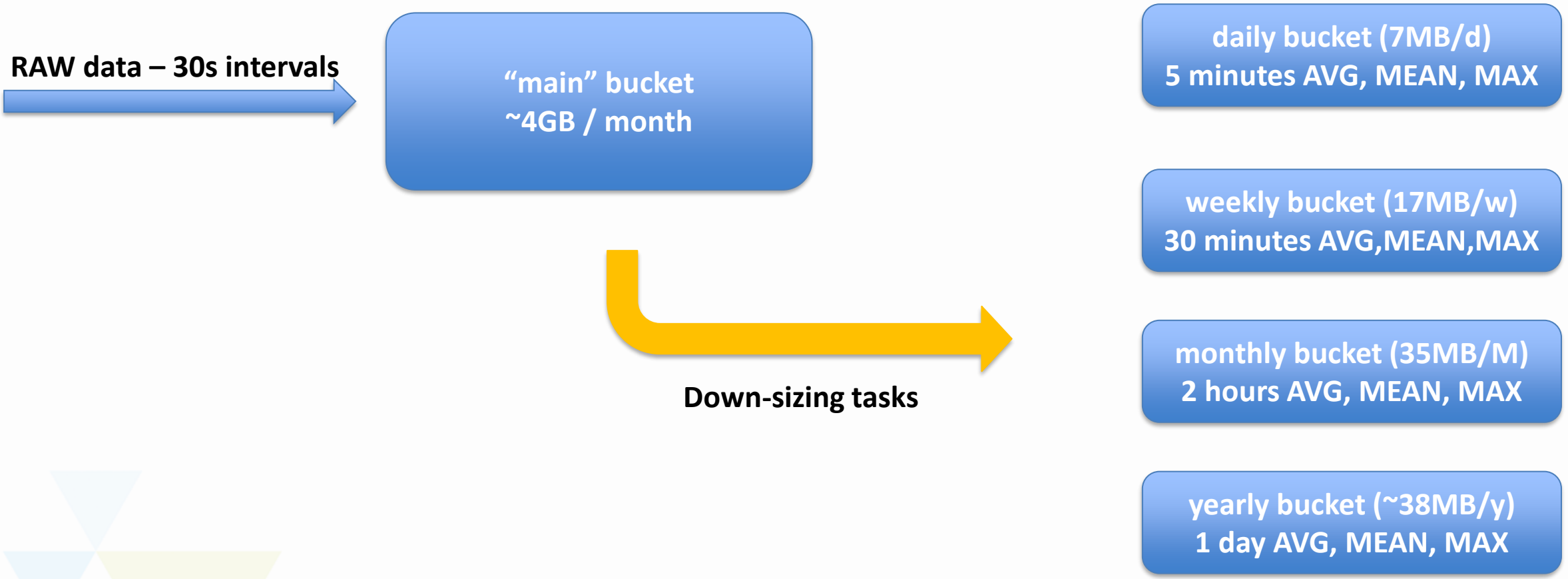
63080 metrics

Collection and storing

700ms to collect
250ms to store



Data handling – InfluxDB 2



Data handling – InfluxDB 2

Flux script example

```
from(bucket: “main")
    |> range(start: -task.every)
    |> filter(fn: (r) => r["_measurement"] == "nx-stats")
    |> aggregateWindow(every: 30s, fn: sum)
    |> aggregateWindow(every: 1d, fn: mean)
    |> set(key: "_measurement", value: "mean")
    |> toInt()
    |> to(bucket: “stat_year")
```

Data handling – InfluxDB 2

main bucket
30 sec AVG, MEAN, MAX

daily bucket
5 minutes AVG, MEAN, MAX

weekly bucket
30 minutes AVG,MEAN,MAX

monthly bucket
2 hours AVG, MEAN, MAX

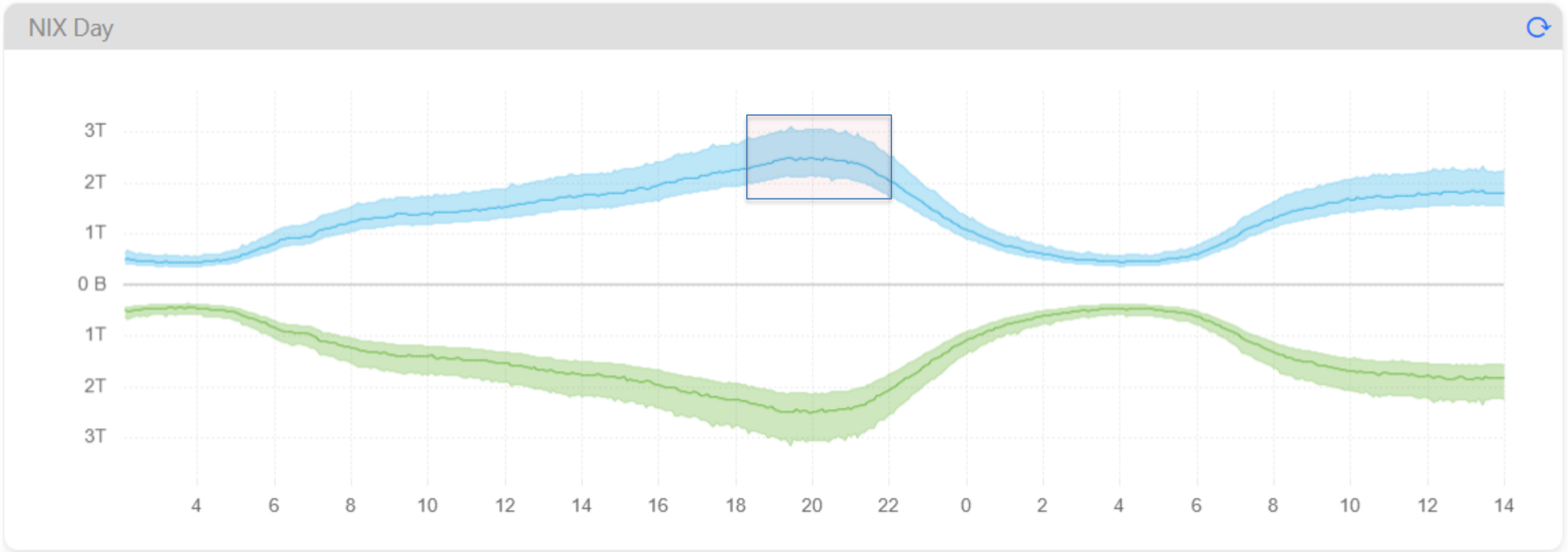
yearly bucket
1 day AVG, MEAN, MAX

Flux lang

Python
HTTP
API

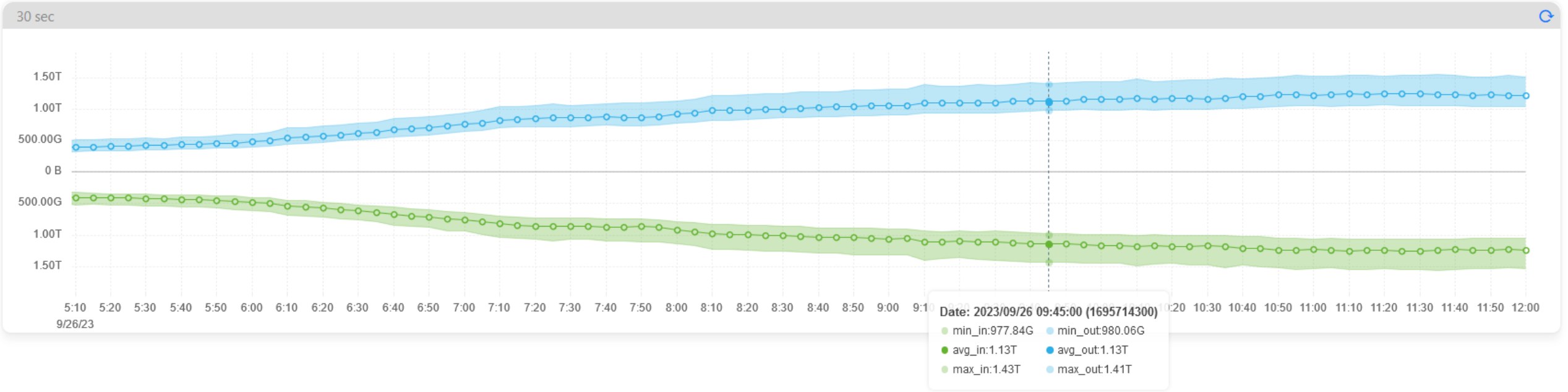
Data handling – InfluxDB 2

Visualization using uPlot



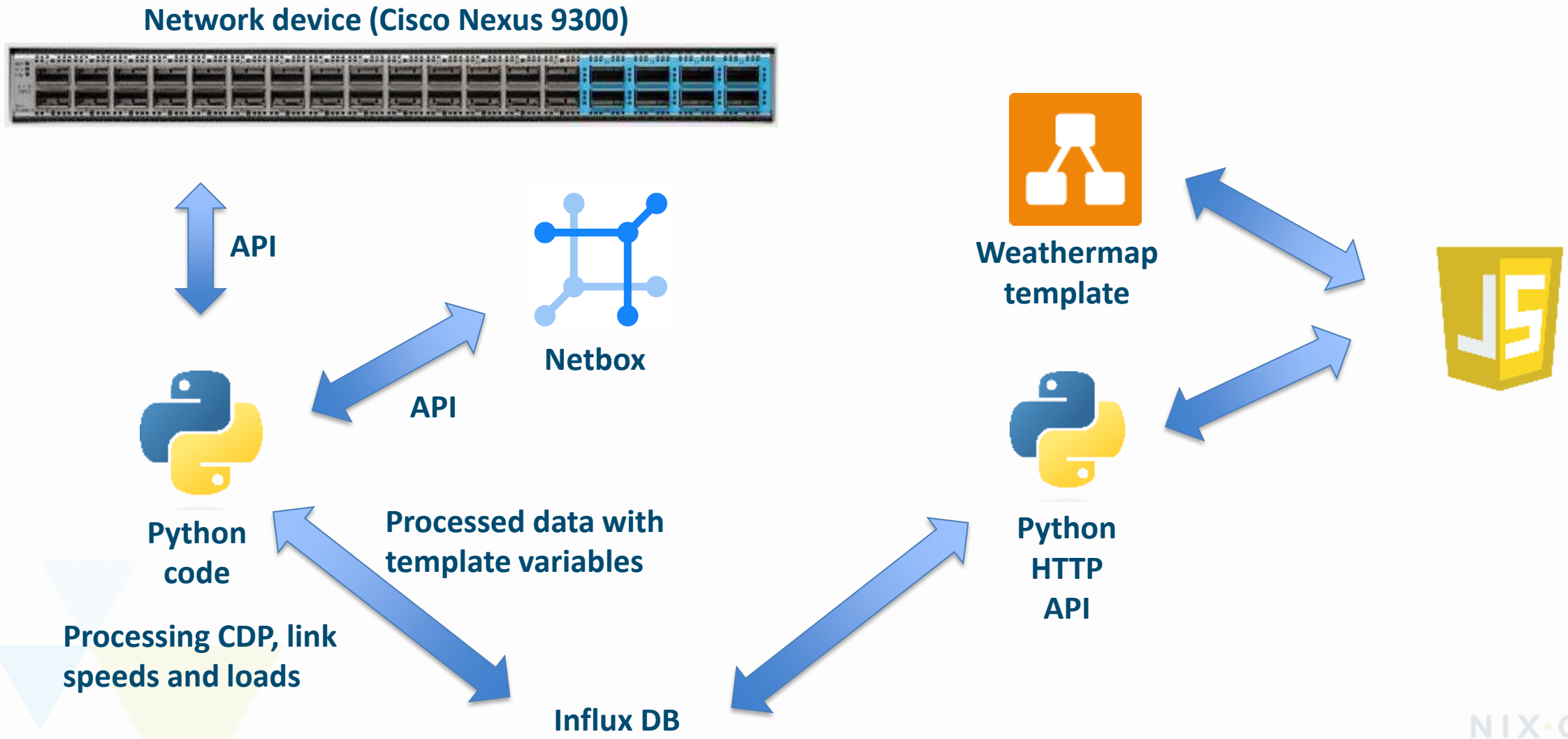
Data handling – InfluxDB 2

Visualisation using uPlot

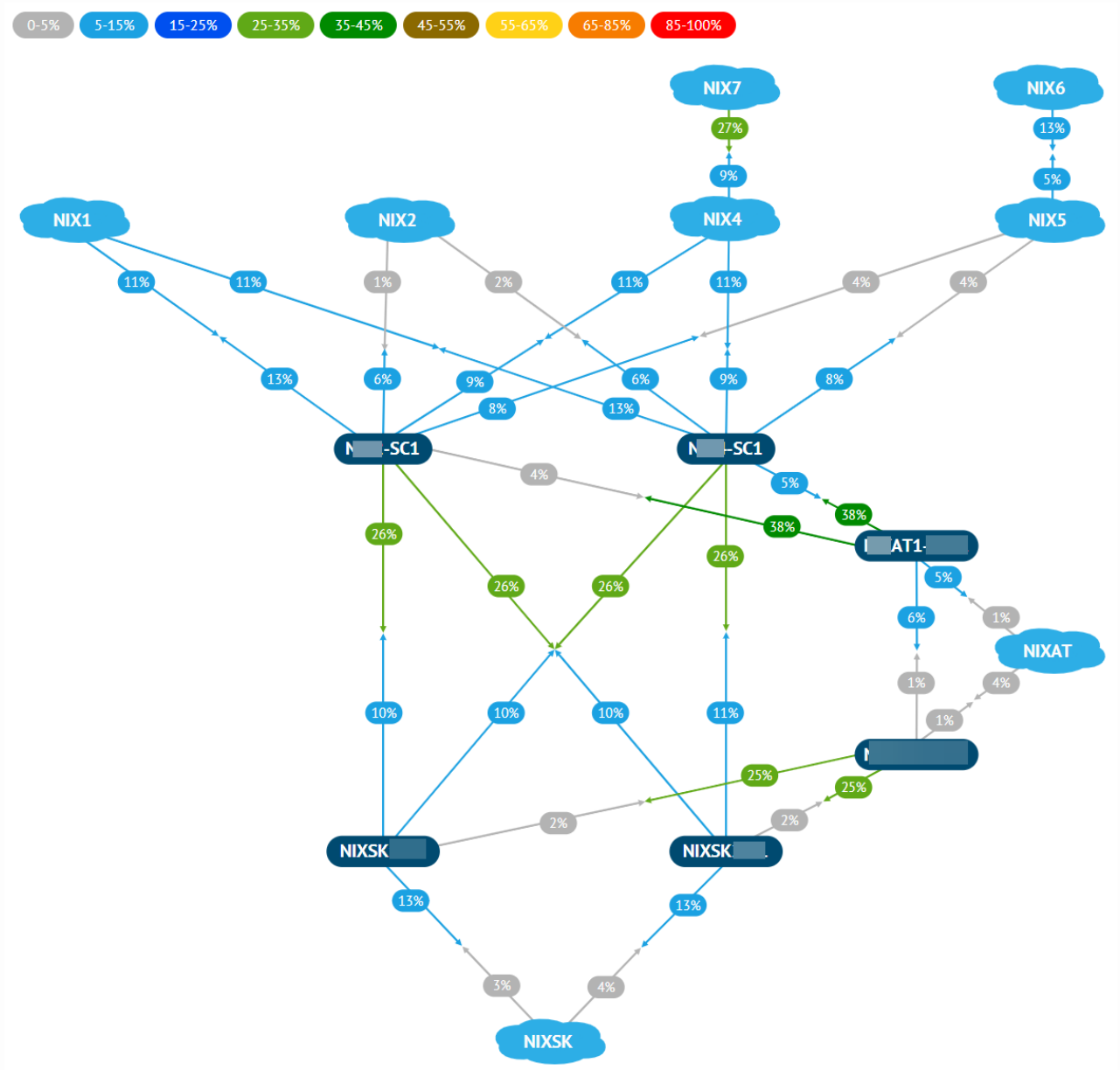


Data handling – InfluxDB 2

Weathermap using InfluxDB

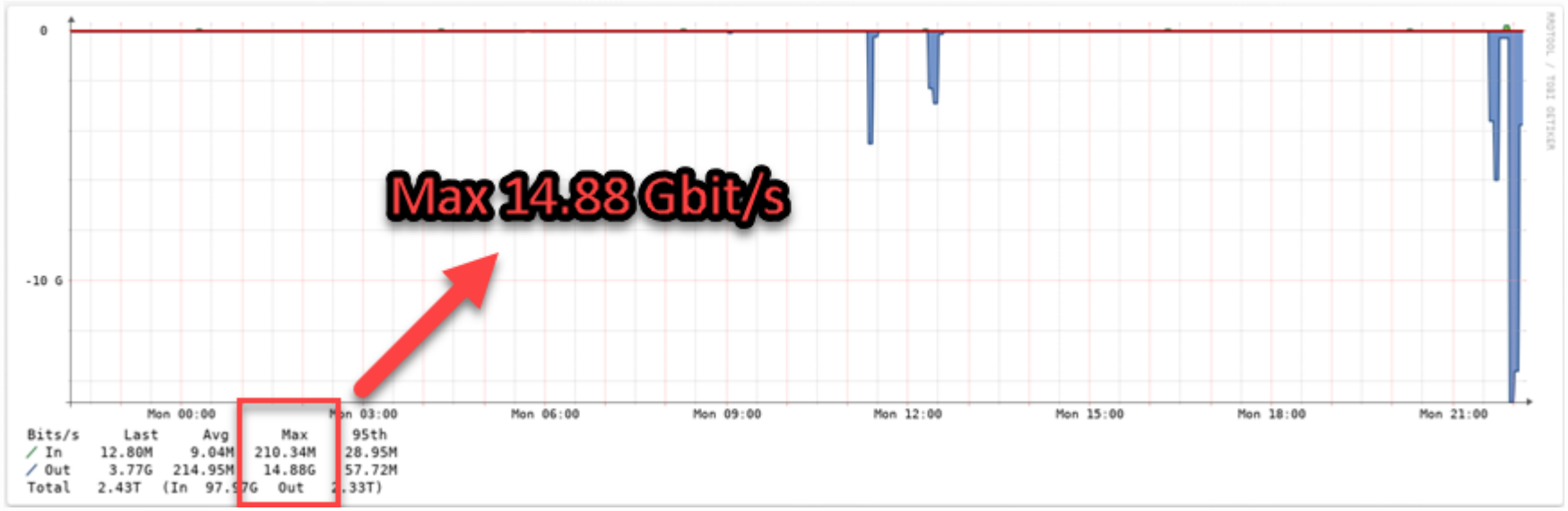


Weathermap using InfluxDB



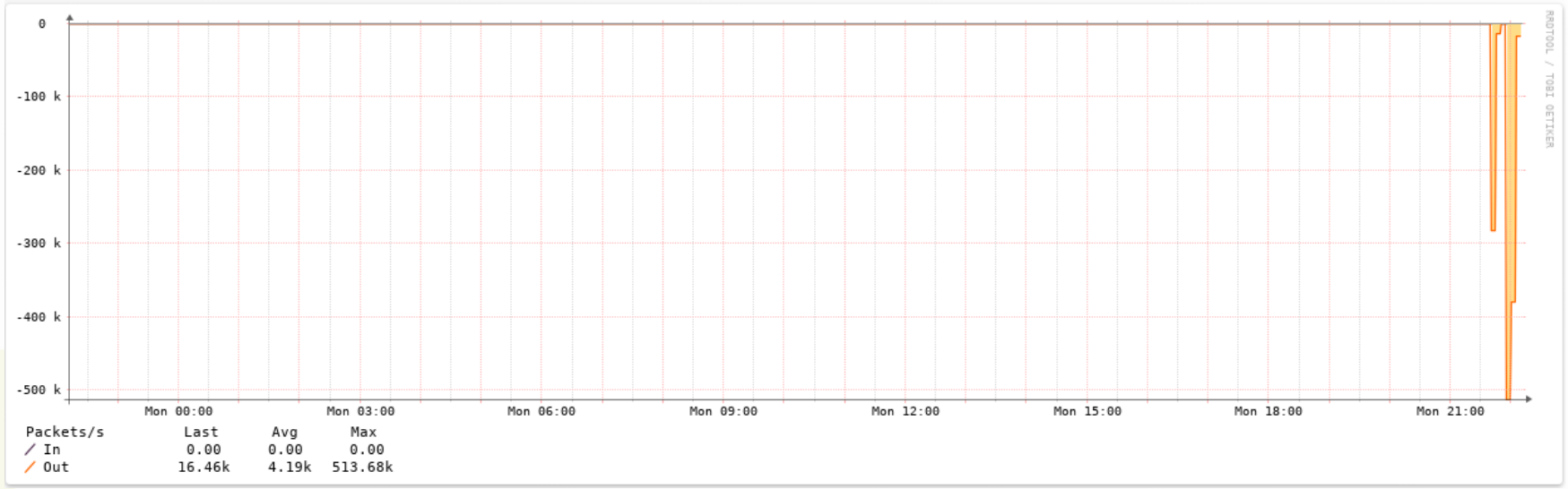
Real life scenario - DDoS

Traffic bit/s



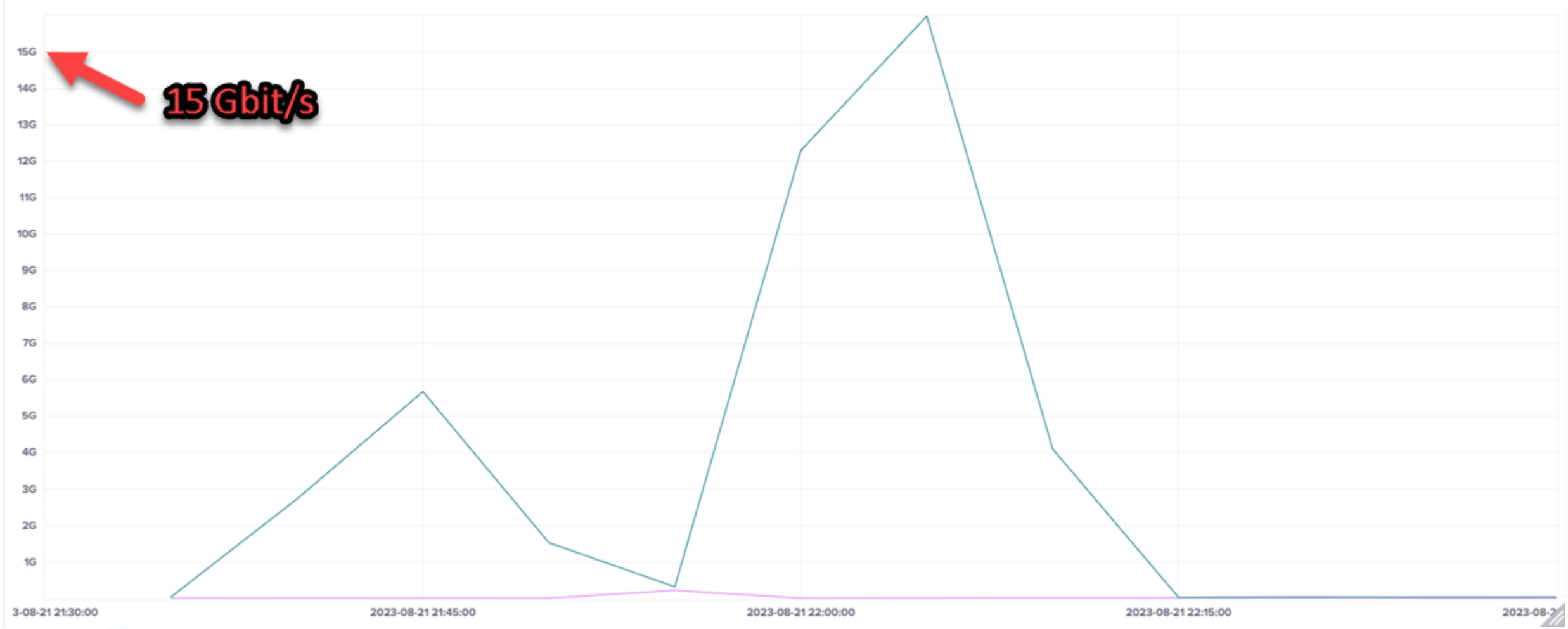
Max 14.88G

Discards p/s



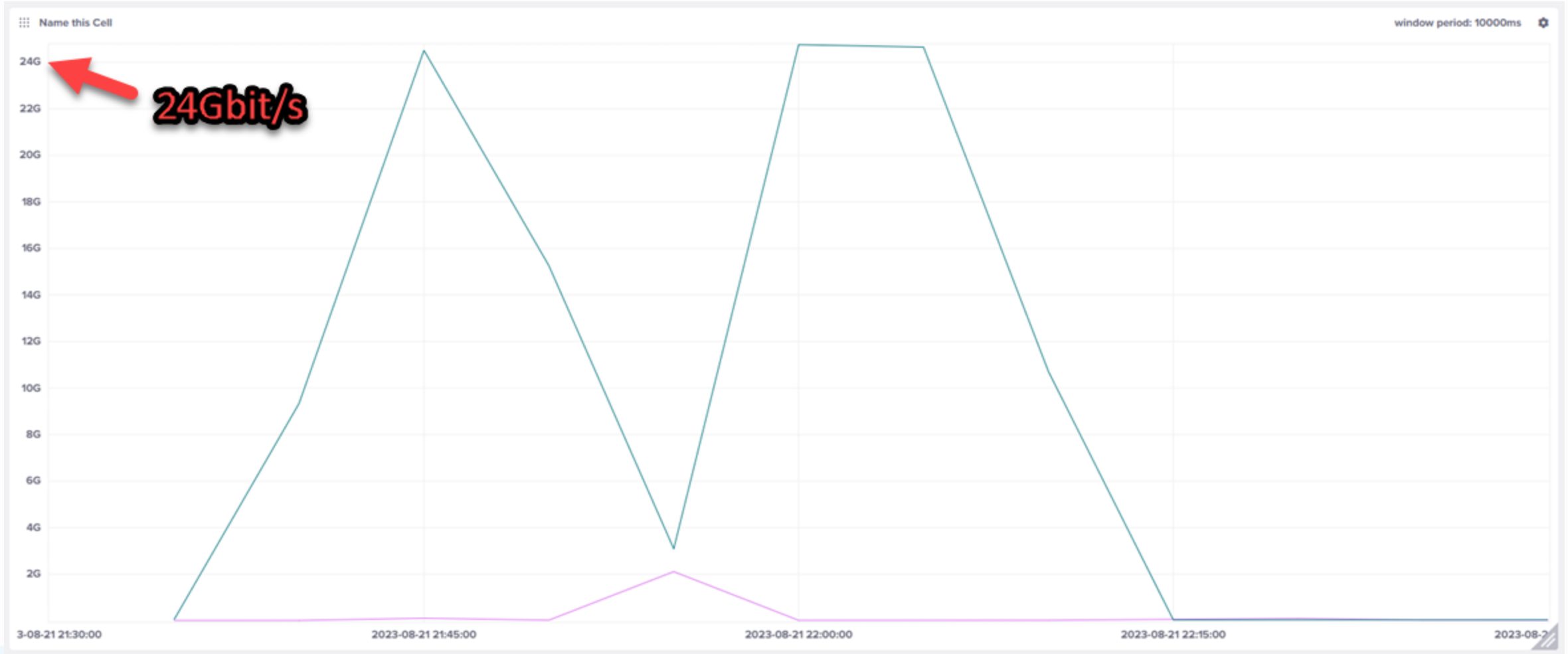
NIX·CZ

Real life scenario - DDoS



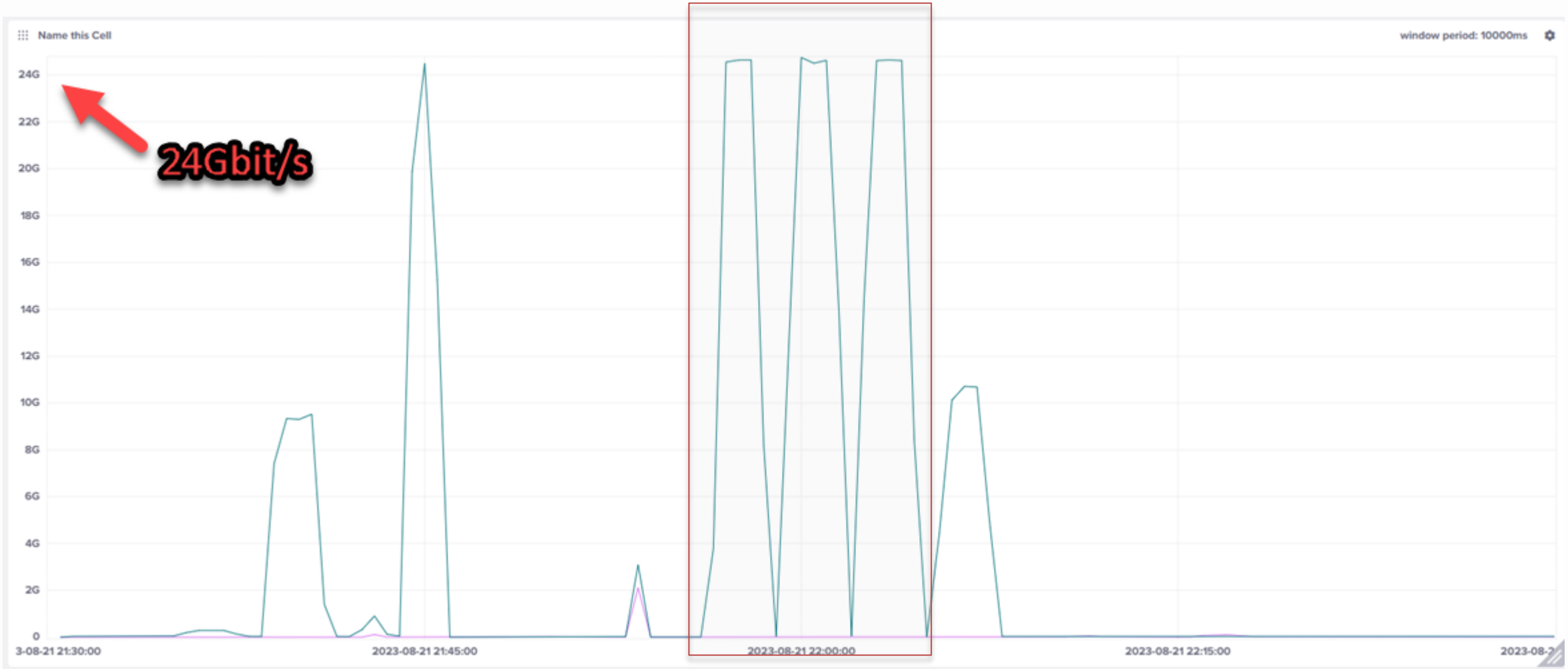
30s intervals averaged in 300s window

Real life scenario - DDoS



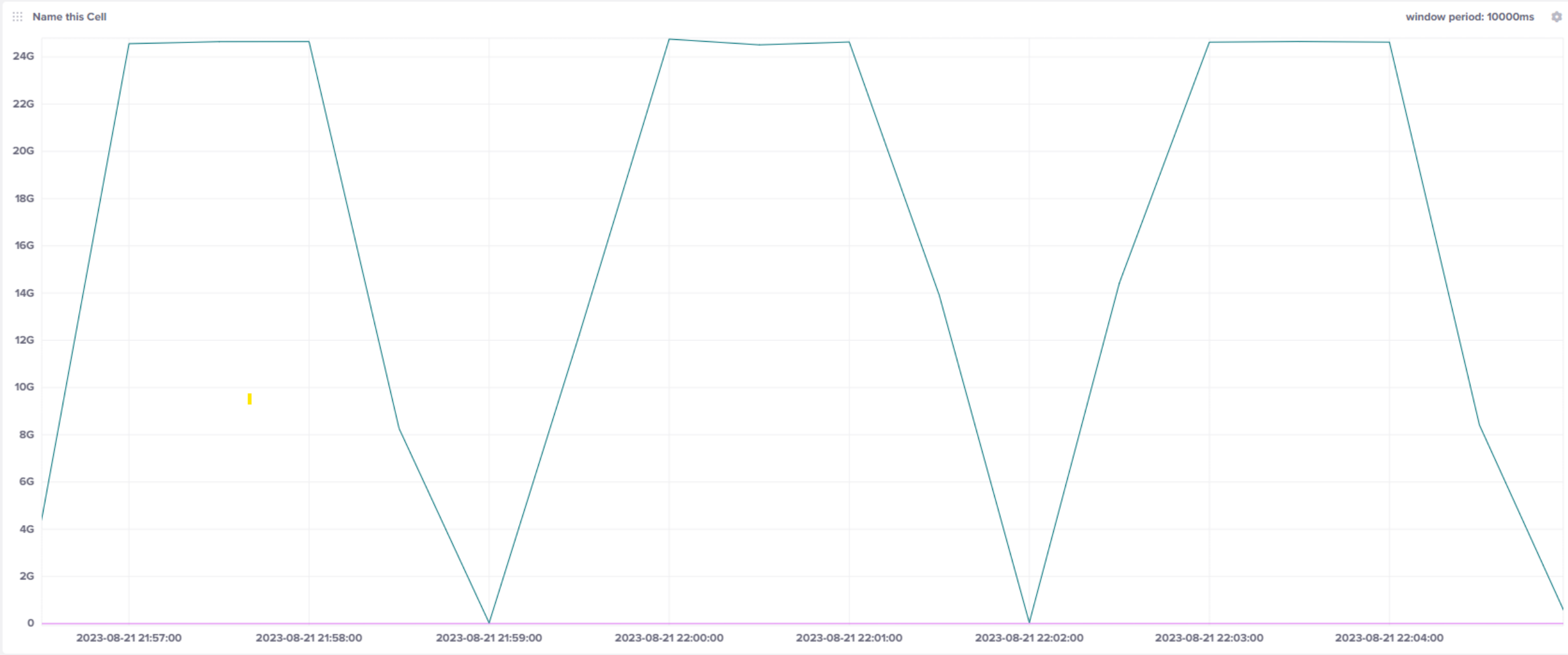
30s intervals max in 300s window

Real life scenario - DDoS



30s intervals maximums

Real life scenario - DDoS



30s intervals maximums - zoomed

(s)Flow analysis

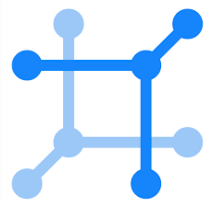
Network device (Cisco Nexus 9300)

sFlow

FTAS

Python code

API



Netbox

Influx DB

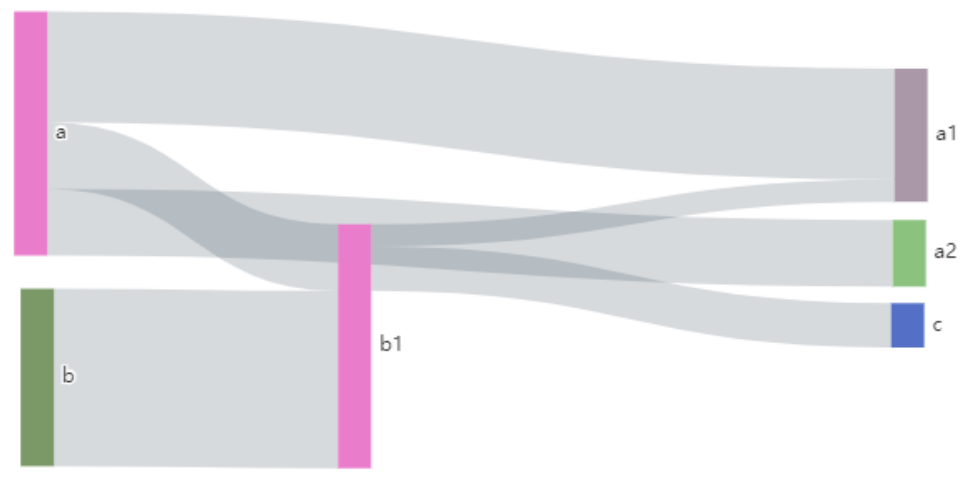
Weathermap template

Python HTTP API

(s)Flow processing

- ~150 Mbit/s of sFlow samples

- downsizing tasks
  - 5 min statistics for day - 700MB/week
  - 30 minutes statistics for week
  - 2 hours statistics for month
  - 1 day statistics for year



Sankey diagram

Thank you for your attention.

mr@nix.cz