



# Usage of time series databases in the Grafana platform for the NETIS service

Evgeny Alexandrov<sup>1</sup>, Mikel Eukeni Pozo Astigarraga<sup>2</sup>,  
Giuseppe Avolio<sup>2</sup>, on behalf of the ATLAS collaboration

<sup>1</sup>Joint Institute for Nuclear Research, Dubna, Russia.

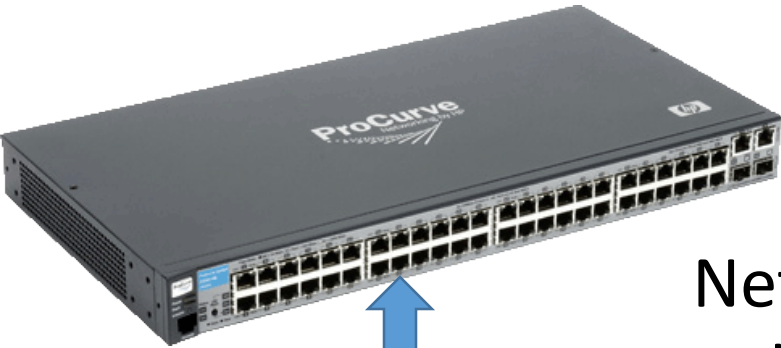
<sup>2</sup>European Laboratory for Particle Physics, CERN.

GRID Conference at JINR

08 July 2021



# Introduction



NetIs is a service used to monitor the Data Acquisition network of the ATLAS experiment.

P1 network:

285 Nodes

4020 Switches

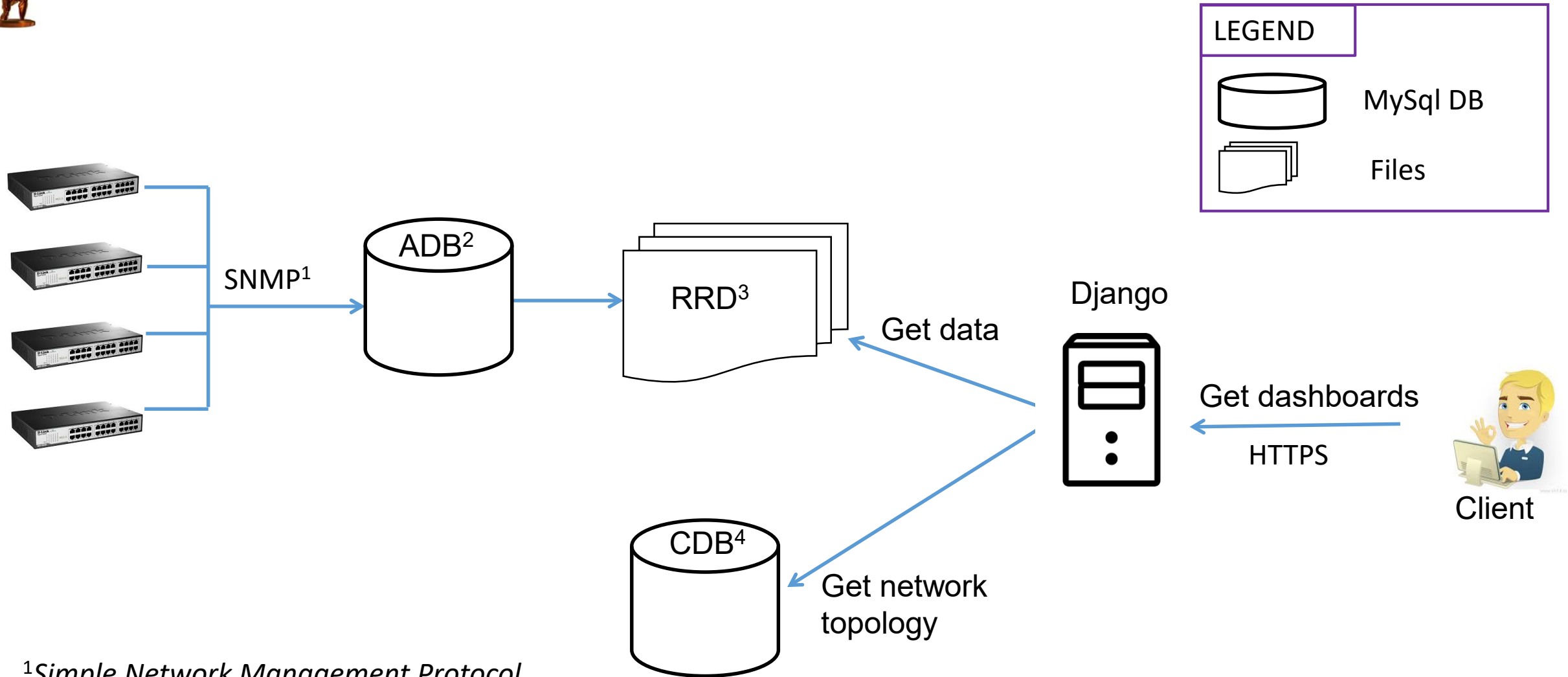
14778 Interfaces (Ports)

The first version was developed at CERN in 2010.

Traffic (MB/s)  
Errors  
Discards  
Packets/s  
Speed  
Status  
30 sec precision



# Legacy Netis: design



<sup>1</sup>Simple Network Management Protocol

<sup>2</sup>Apoll DB

<sup>3</sup>Round-robin database

<sup>4</sup>Central Database



# Legacy Netis: View

Tree  
(JS library)

Zoom  
(JS library)

View type: Aggregates | Interval: 2016-07-05 12:48 till 2016-07-06 12:48 [3h|24h] | Filter: | Refresh | Stop | Search | Go to LR

**NETWORK::Traffic Aggregates** | Control core :: Ctrl ATCN SDX backbone

**Tree (JS library)**

- Data Flow Sections
  - Control core
    - Ctrl ATCN SDX backbone**
      - Ctrl backbone SDX-USA
      - Ctrl / ROS switches
      - Ctrl / TPU switches
      - Ctrl / Monitoring machines
      - Ctrl / Online machines
      - Ctrl / Uplinks
    - DataEdge switch
    - DataCollection switch
    - ROS subsystem
    - TPU computers
    - SFO computers
    - MLTSV computers
    - Calibration
    - Trunks
    - Important endnodes
    - Infrastructure
    - Starpoint SDX
    - Starpoint USA15
    - ACR PCoIP switches

**Settings**

Name: net.interface.pattern:UPLINK.\*(D3178).\*-sw-ctrl-core-.\*  
 Category: net / interface  
 Date start: 2016-07-05 12:48  
 Date end: 2016-07-06 12:48  
 Plot name / func: traffic / SUM  
 Filter:

**Aggregate**

Agg: SUM | Mode: Only aggregate plots | Scale: Default | Lower: | Upper: | Apply

**TRAFFIC**

AGGREGATE SUM UPLINK.\*(D3178).\*-sw-ctrl-core-.\*::TRAFFIC

net.interface.pattern:UPLINK.\*(D3178).\*-sw-ctrl-core-.\*  
 Tue 05 Jul 2016 12:48 - Wed 06 Jul 2016 12:48

|                 | Last   | Avg    | Min   | Max     |
|-----------------|--------|--------|-------|---------|
| Switch Input B  | 34.1 M | 43.6 M | 5.9 M | 268.4 M |
| Switch Output B | 66.7 M | 60.6 M | 7.2 M | 118.2 M |

2 plots found; 0 files missing; 0 ds missing;  
 (c) ATLAS Networking Team / 2016-07-06 12:48 [net\_interface]

**Miniplots**

Link aggregation: Trunks and trunk members | leftmost plot: Scale: Default | Lower: | Upper: | Apply

| #  | Device            | If                         | Peer | LOAD | ERRORS/DISCARDS | TRAFFIC |
|----|-------------------|----------------------------|------|------|-----------------|---------|
| 1. | sw-ctrl-core-0111 | UPLINK: d3178-2-rhp82-1.B1 |      |      |                 |         |
| 2. | sw-ctrl-core-0211 | UPLINK: d3178-2-rhp82-2.C1 |      |      |                 |         |

**Images**  
Is created by python Matplotlib

**DropDown menu**



# Motivation for changes

- It is difficult to maintain;
  - Requires knowledge on Django templating language;
  - Requires knowledge on Matplotlib library;
  - Requires knowledge on Web and JavaScript technologies (on the other hand this is also true for the system presented).
- Loss of data granularity and resolution
  - Stores data in the Round Robin Database (RRD);
- Static GUI
  - The graphs (images) are generated by server
  - Quite static though the GUI is familiar to many users.

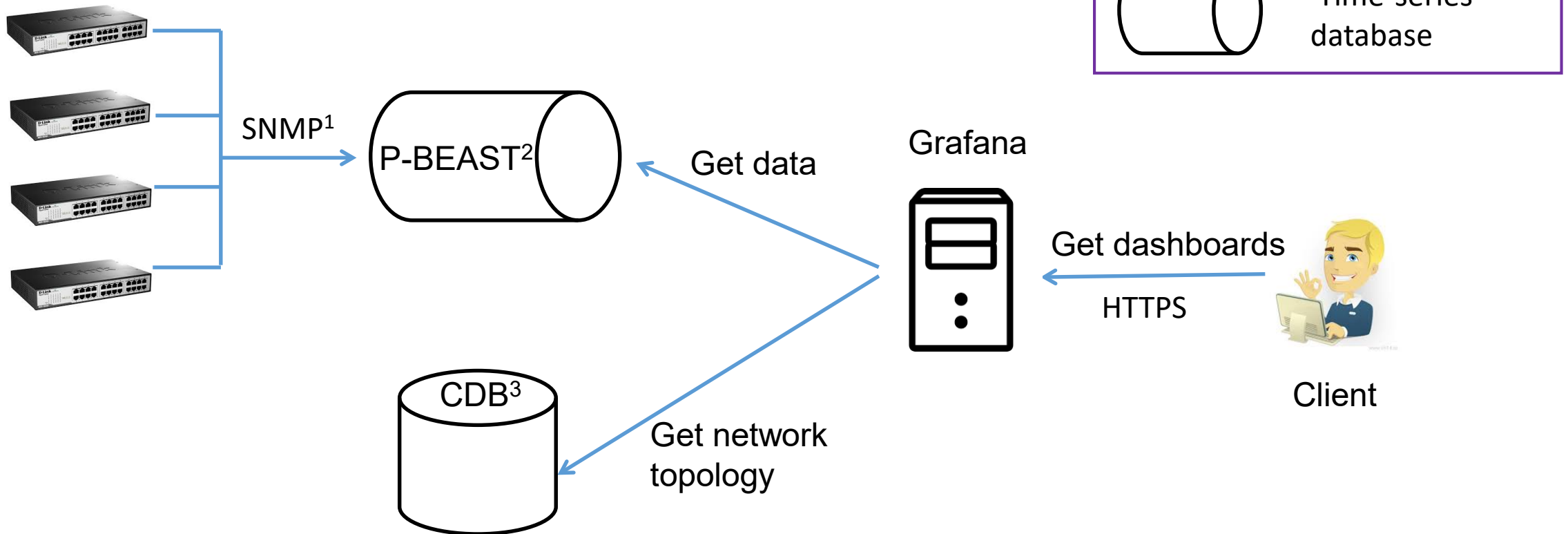


# Solution

- Loss of data granularity and resolution
  - Use the Persistent Back-End for the ATLAS Information System (P-BEAST) time-series database, developed in ATLAS for permanent storage of operational data
- Static GUI
  - Use open source tool Grafana so that data can be shown in a dynamic way
- It is difficult to maintain
  - Remove Django templating language and Matplotlib library
  - Requirement only knowledge on Web and JavaScript technologies



# Design




<sup>1</sup>Simple Network Management Protocol

<sup>2</sup>Persistent Back-End for the ATLAS Information System

<sup>3</sup>Central Database



# Problems: Grafana common panels




Pie Chart

Panel




Singlestat

Panel




Text

Panel




Graph

Panel




Singlestat

Panel



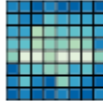
Table

Panel



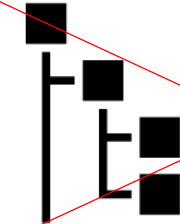
Dashboard list

Panel



Heatmap

Panel



Tree

PANEL NOT EXIST





# Problems: Grafana DropDown

Position of  
DropDown is always  
Up-Left corner.

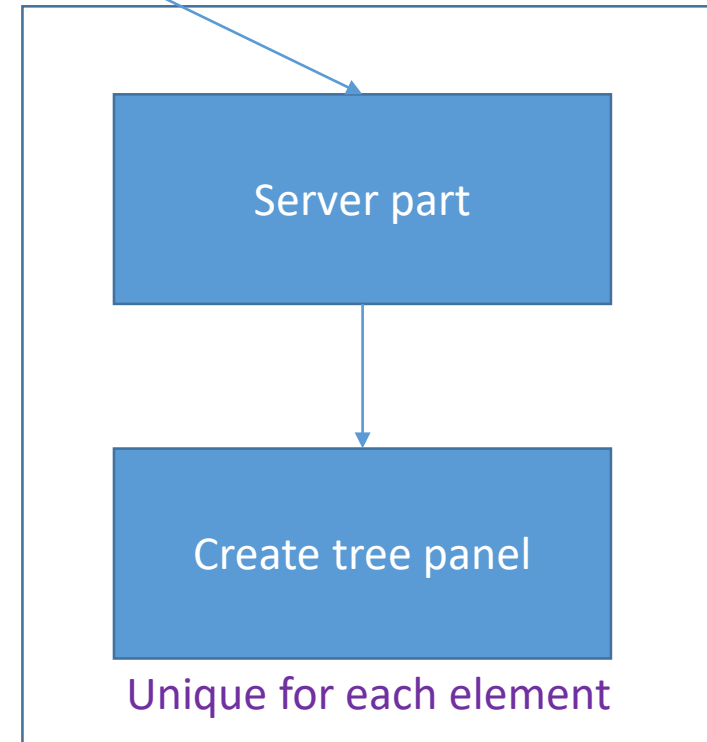
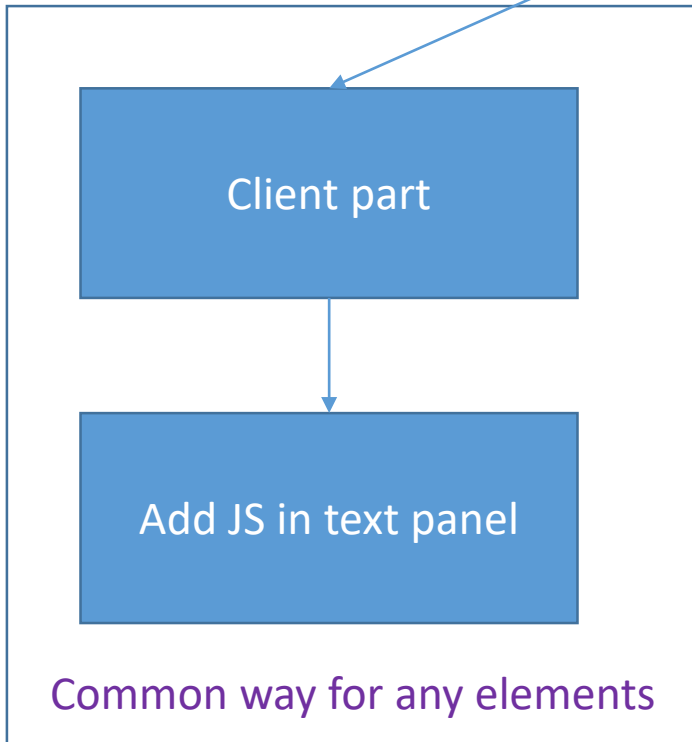




# Grafana tree



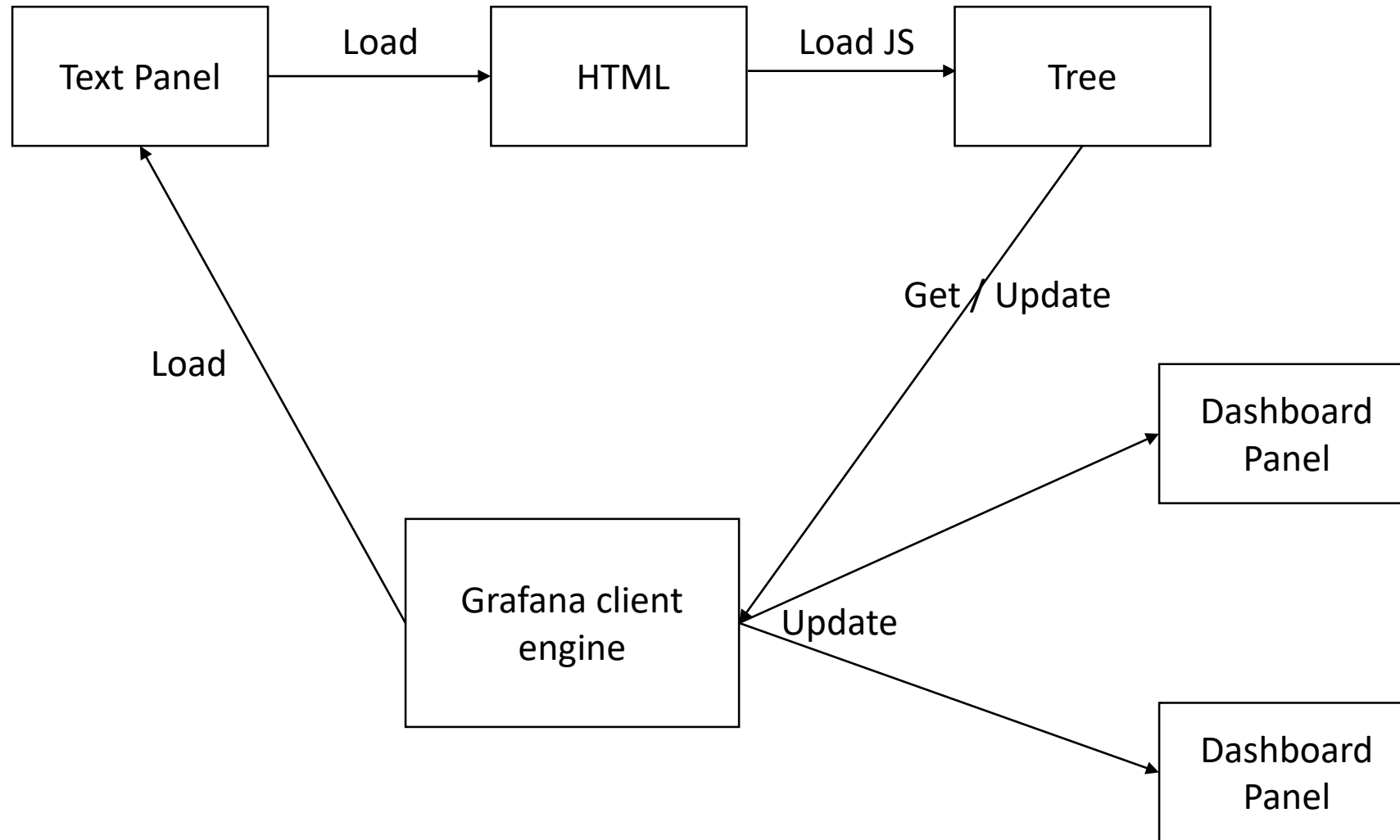
Two ways to add a tree to Grafana



We decide  
implement this  
way



# Tree: Integration & Interaction





# Tree

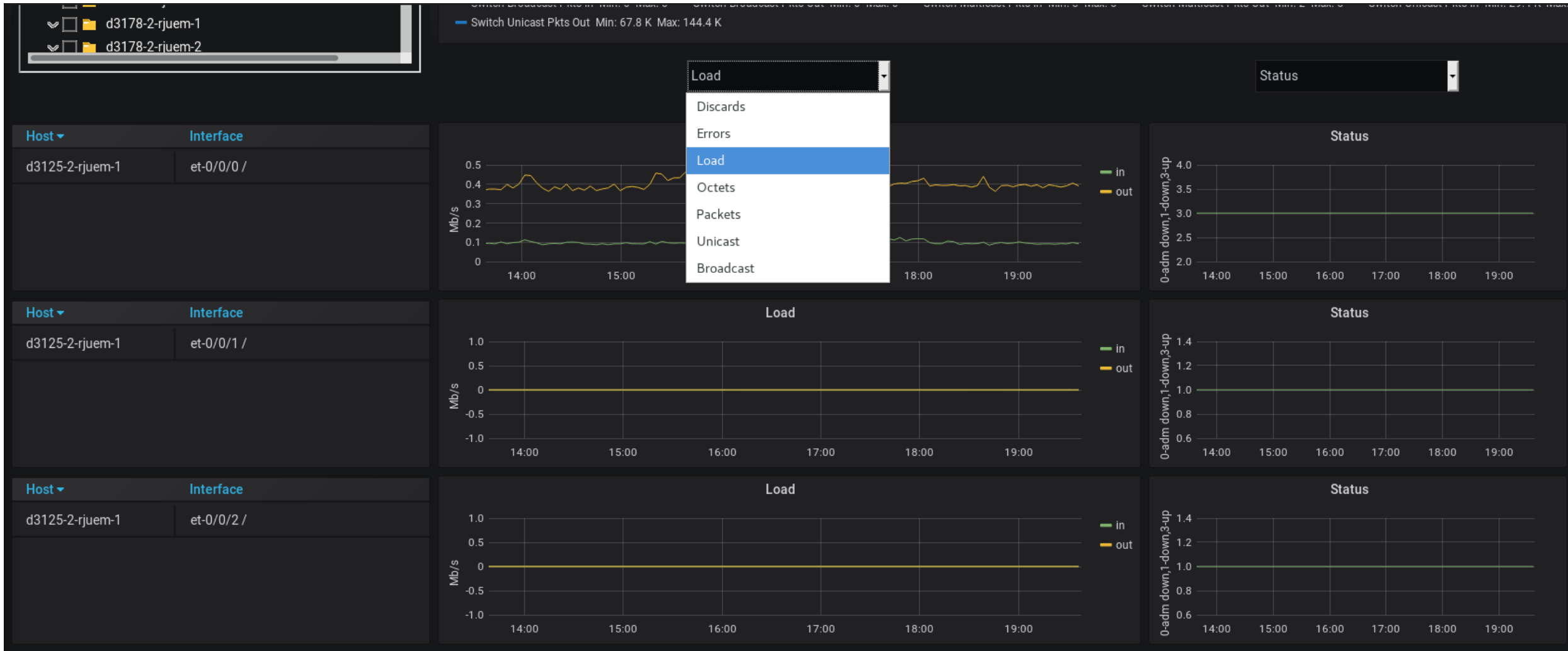
- Implementation is free version of JS Library dhtmlxtree\*
- It is the navigation tool
  - Exists on all pages
  - Controls the structure of the HTML page, Grafana templates and dashboards
- Source data
  - Source data from MySQL is stored in the template in the special format:  
*Function::Device::Linecard::Interface*
- The node Id contains information about the level of the node in the tree and its parents. Id has following format:
  - Function: *\_RR\_FunctionName*
  - Device: *Deviceld*
  - Linecard: *\_LL\_Deviceld:LinecardId*
  - Interface: *\_II\_Deviceld:LinecardId:Interfaceld*

\*<https://dhtmlx.com/docs/products/dhtmlxTree/>

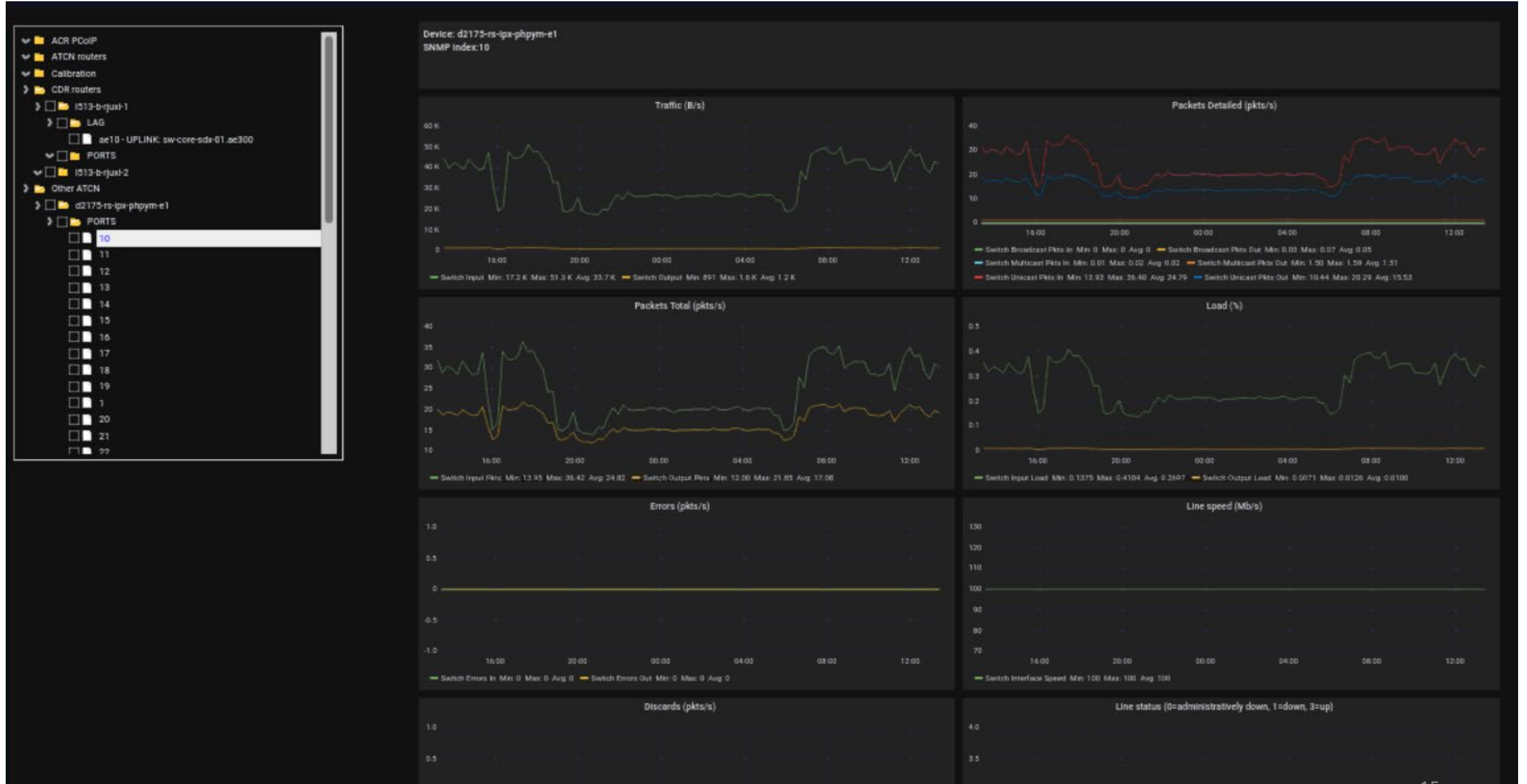
# New version: aggregate page



# New version: miniplots & DropDown



# New version: interface page





# Conclusion

- The new service requires less knowledge to support
- Using the P-BEAST time-series database allowed not to lose detail of the data and not degrade the resolution of the dashboards
- It use Grafana platform so that data can be shown in a dynamic way
- The new service added to ATLAS TDAQ Network software and is actively used to monitor the system during ATLAS commissioning tests.