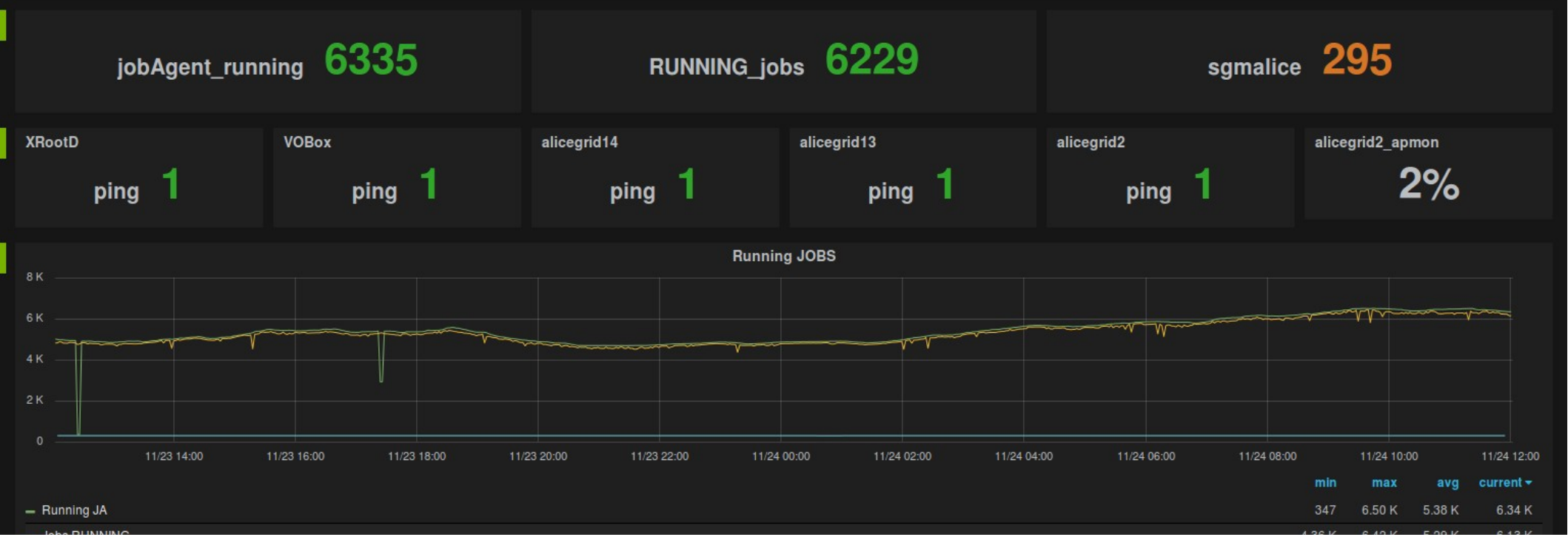


A Dashboard for the Bari INFN Datacenter

Gioacchino Vino
University and INFN Bari

ALICE Offline week - 24-27/11/2015

Outline

- Motivation
- Previous ALICE::Bari Monitoring
- The Dashboard
- Performances
- Outlook
- Conclusions

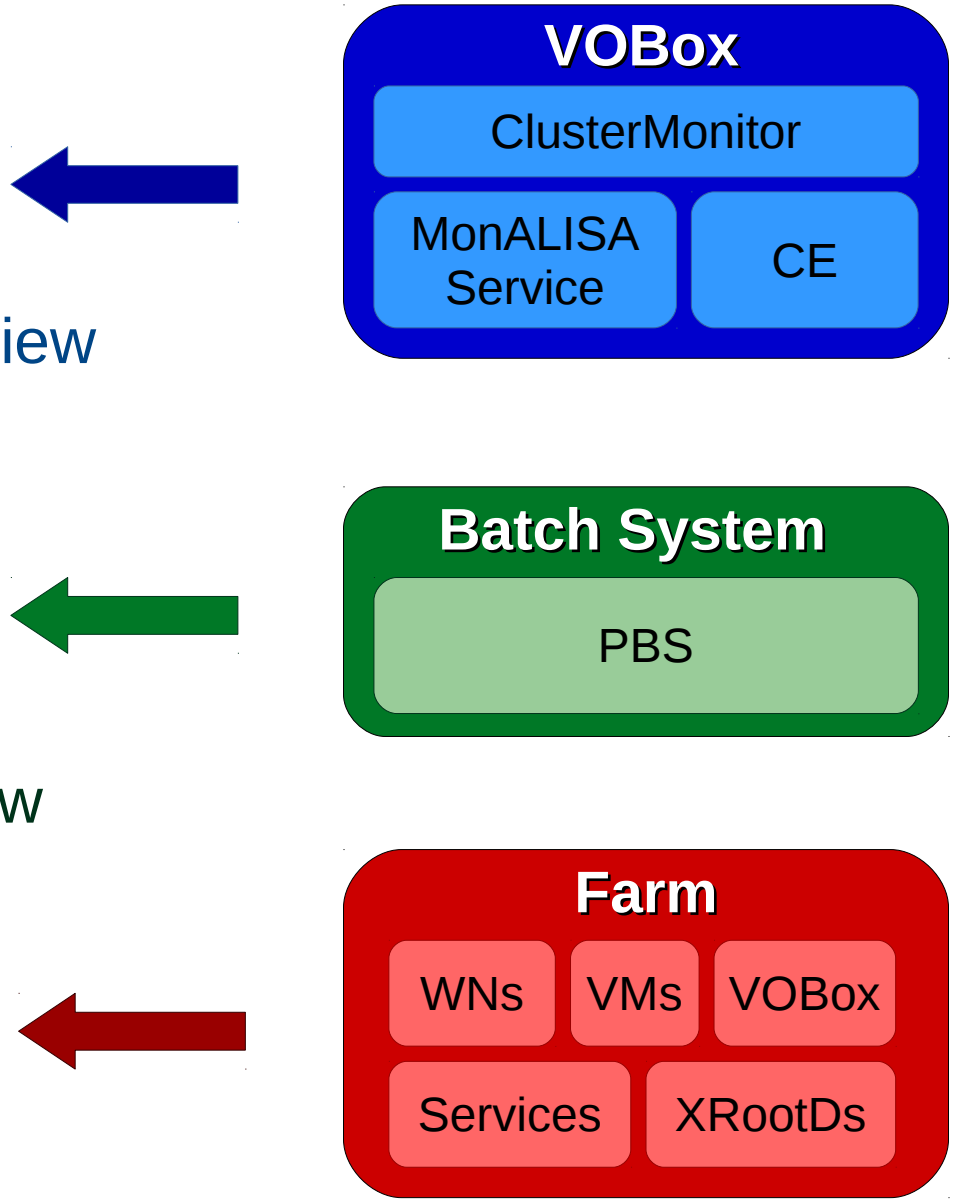
Motivation

- Concentrate in a **single** dashboard **all the information** concerning the ALICE activity in Bari Tier-2 datacenter
- Monitor **other ALICE activities** (e.g. VAF, local simulations, etc)
- Provide a **different** billing tool
- Allow for debug using real-time values coming from **multiple sources**
- Make statistical reports and time series
- Monitor activity in sites serving **different VOs**, experiments and projects

Previous ALICE::Bari Monitoring Systems

Monitoring on 3 levels:

- **MonALISA**:
  - Site facility
  - Network
  - Job state from ALICE point of view
- **TorqueMon**:
  - Queue status
  - Batch server status
  - Job state from PBS point of view
- **Zabbix**:
  - VMs state
  - Cloud infrastructure state



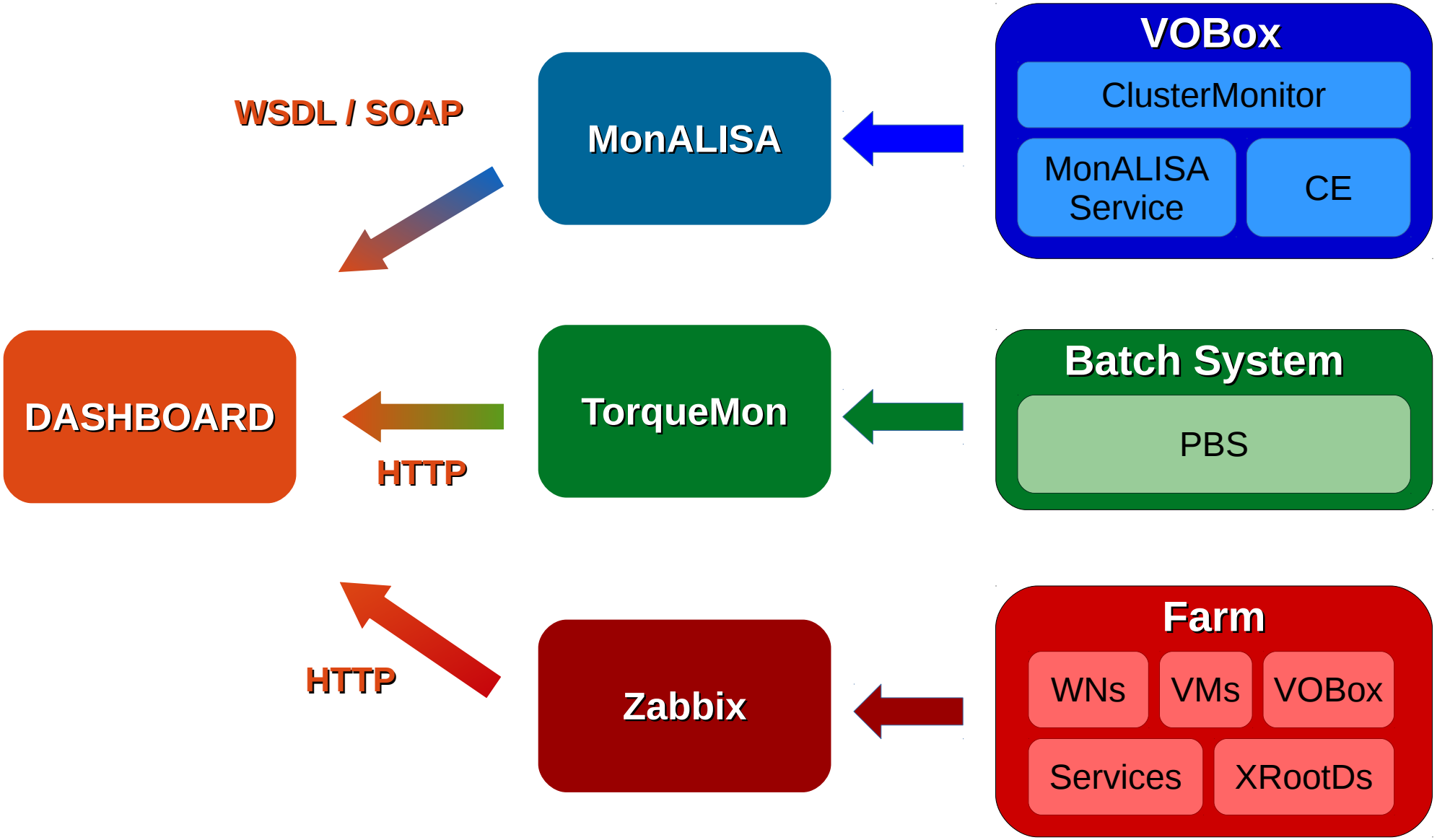
The Dashboard

Improve site monitoring wrt its **limitations**:

- Information **spread out** over 3 monitoring systems
- Existing GUIs **not** completely and easily **customizable** and **interactive**
- For Zabbix and TorqueMon **not** possible to **extract** monitoring info from the systems

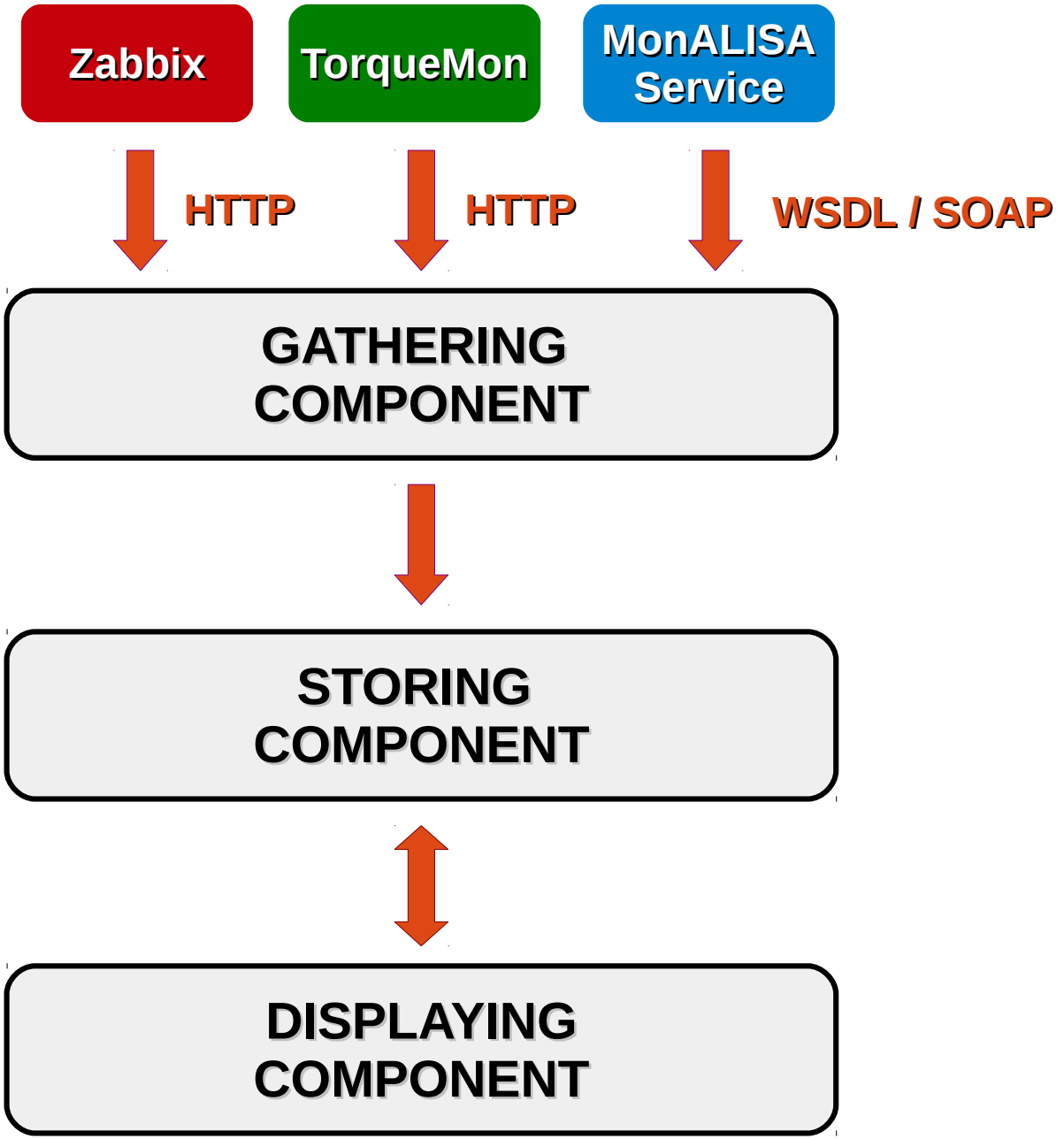
The Dashboard

Connection to data sources:



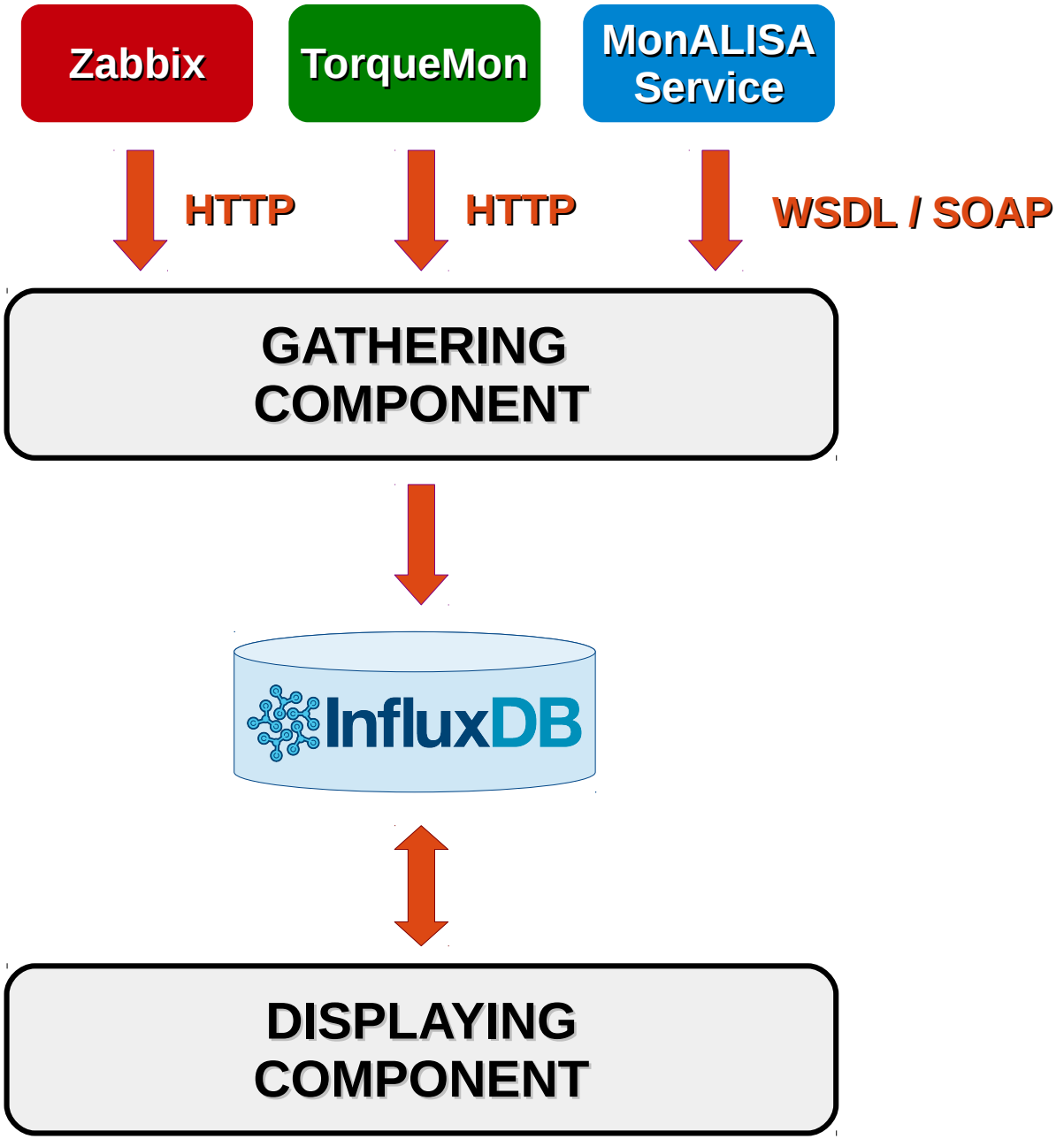
The Dashboard

Architecture:



The Dashboard

Architecture:



Storing Component

The Dashboard

Architecture:

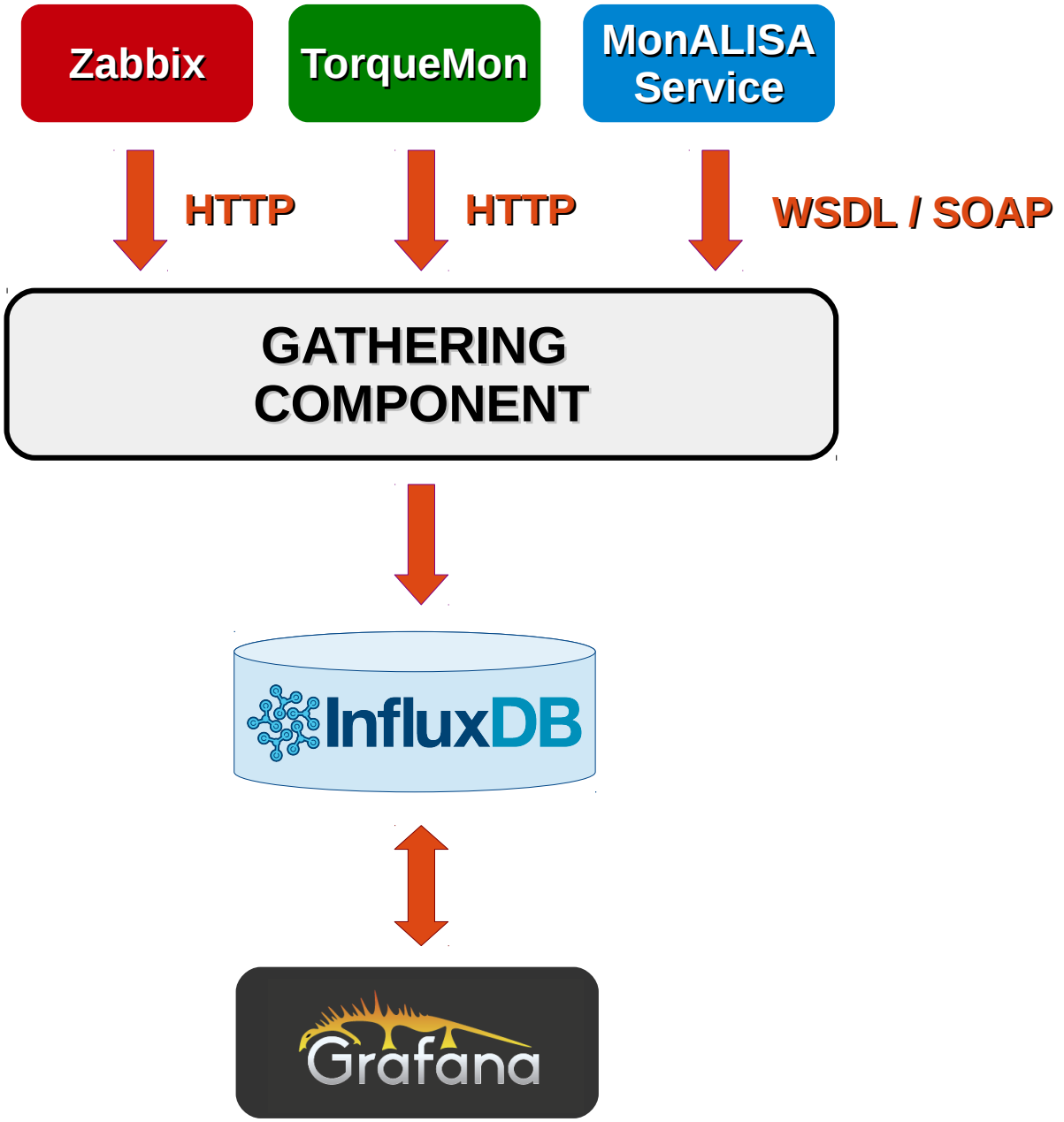
Zabbix | TorqueMon | MonALISA Service

- Open source **time series database** written in Go
- Based on LevelDB, a key-value database
- **HTTP / JSON interface** for managing, reading and writing data
- API for the main programming languages (JavaScript, Ruby, Python, Node.js, PHP, Java, Go, R and Perl)
- Provides a Web front-end for managing and debug
- **SQL like** query languange
- **On-the-fly aggregation** with different functions: mean, median, ...

```
SELECT MEAN(column_name) FROM series_name GROUP BY TIME(1d) ...
```

The Dashboard

Architecture:



Storing Component

Displaying Component

The Dashboard

Architecture:

Storin

Displa

- Front-end for Graphite, InfluxDB and OpenTSDB with powerful visualization features for time series data
- **No dependences**
- Written in Node.js
- It is a **client side application** that runs in a **browser**
- Graphical objects: graphs, singlestats, texts and images
- Graphs with metrics coming from different databases
- Plotted values can be **exported** in CSV format
- Custom data sources

The Dashboard

Architecture:

Zabbix

TorqueMon

MonALISA Service

Grafana

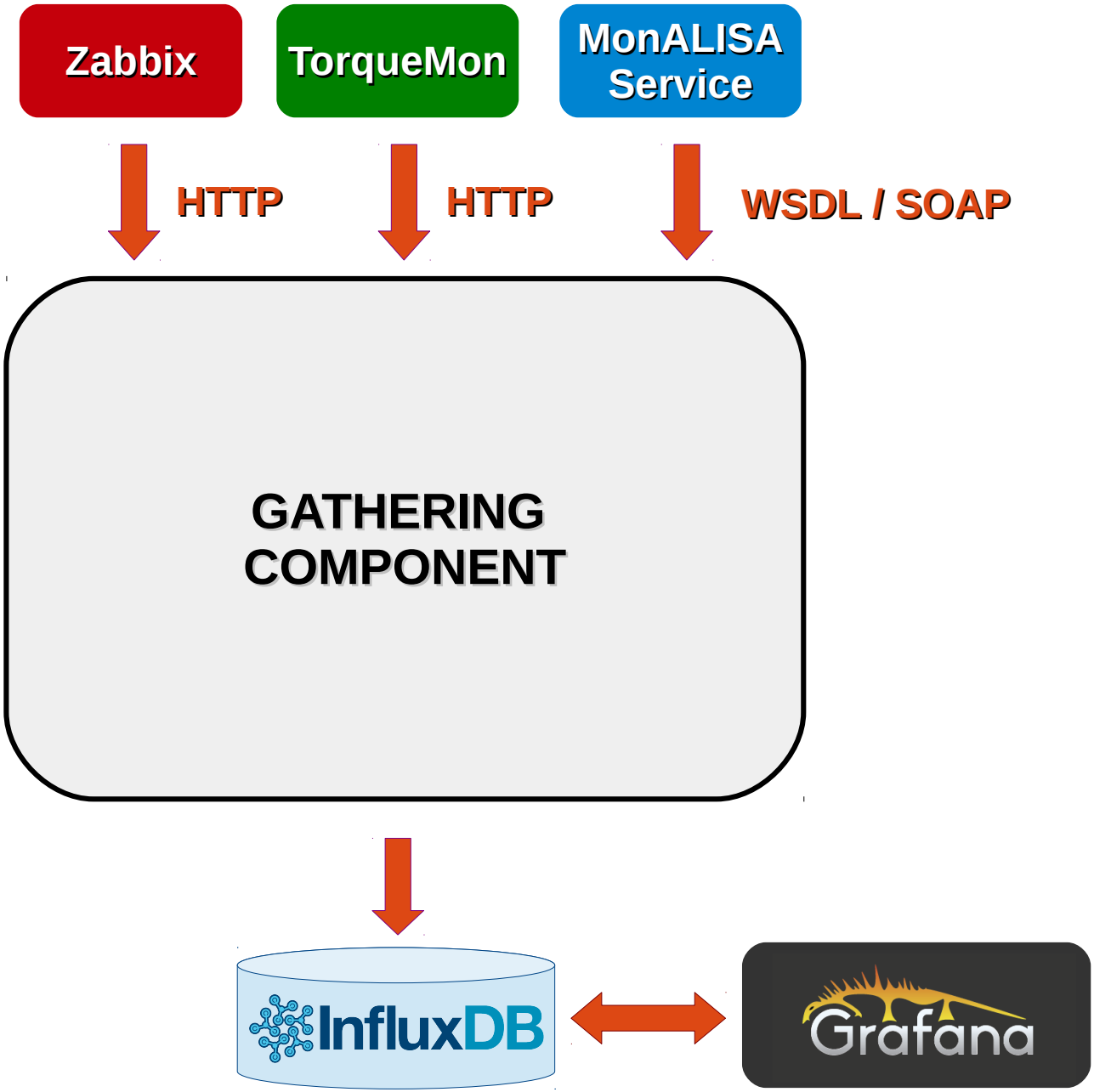
- Click and select region to zoom
- Multiple Y-axes
- Bars, Lines, Points
- Programmable refresh period

Storin

Displa

The Dashboard

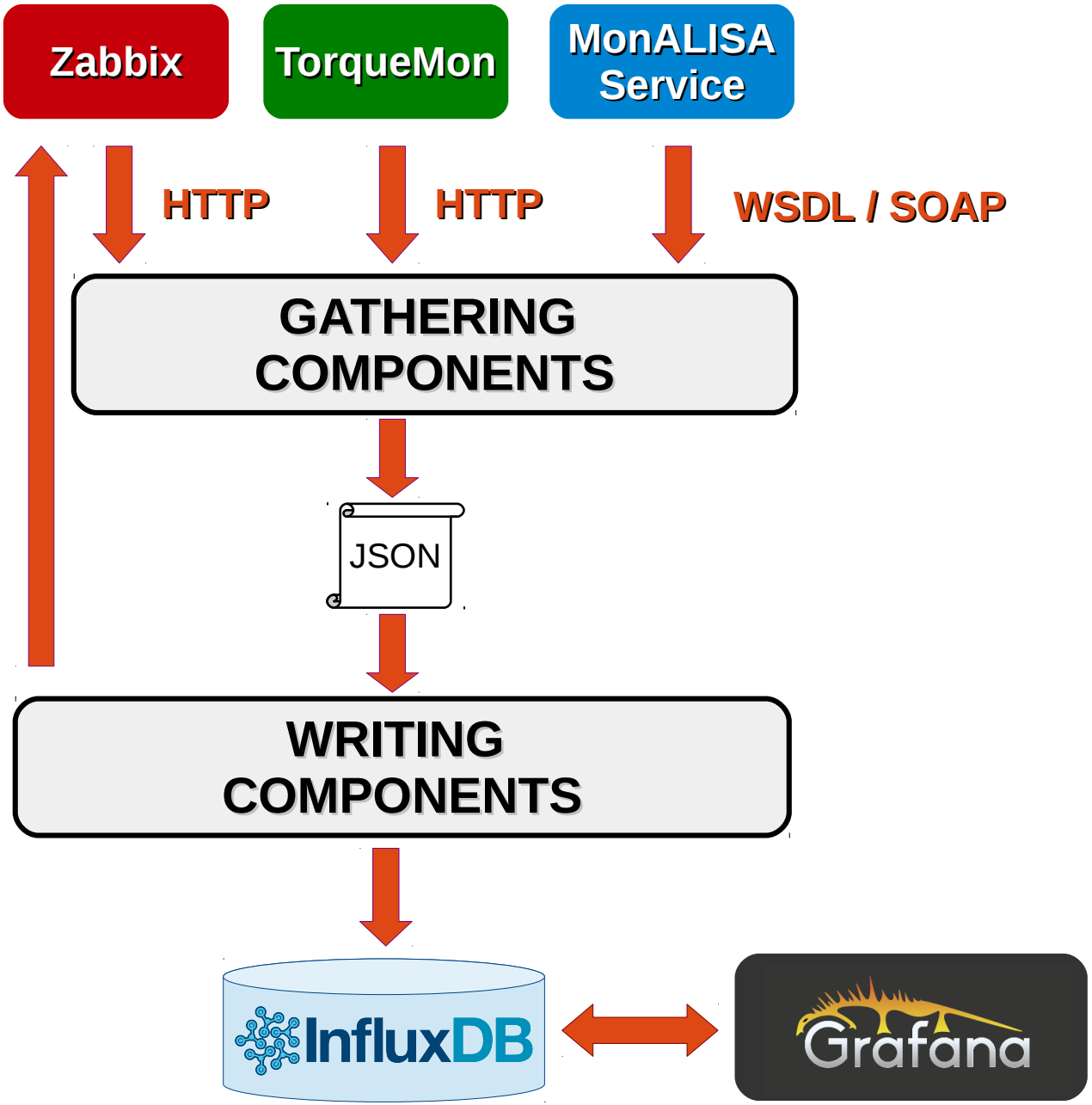
Architecture:



The Dashboard

Architecture:

- No software available for the selected data sources
- Code written in Python:
  - Object Oriented Programming Language
  - Thousands of extentions
  - InfluxDB Client
  - Zappix API
  - WSDL / SOAP module
- Modular architecture with a temporary JSON file containing data and metadata
- Gathered data can be forwarded toward others systems → Zabbix



The Dashboard

Example of a sensor code using the python Influx API:

- Needs an installation:

pip install influxdb

- Script:

```
#!/usr/bin/env python
from influxdb import InfluxDBClient
Host='my-hostname'
Port=8086
User='my-username'
Password='my-password'
DBname='my-dbname'
MetricName='temperature'
Timestamp=1445704538000
Value=26.0239284
Data=[ [ Timestamp, value ] ]
db=InfluxDBClient(Host,Port,User,Password,DBname)
db.write_time_value_points(MetricName,Data)
```

- InfluxDBClient dependences:
  - requests
  - json
  - socket (only for UDP)

The Dashboard

Example of a sensor code using Bash:

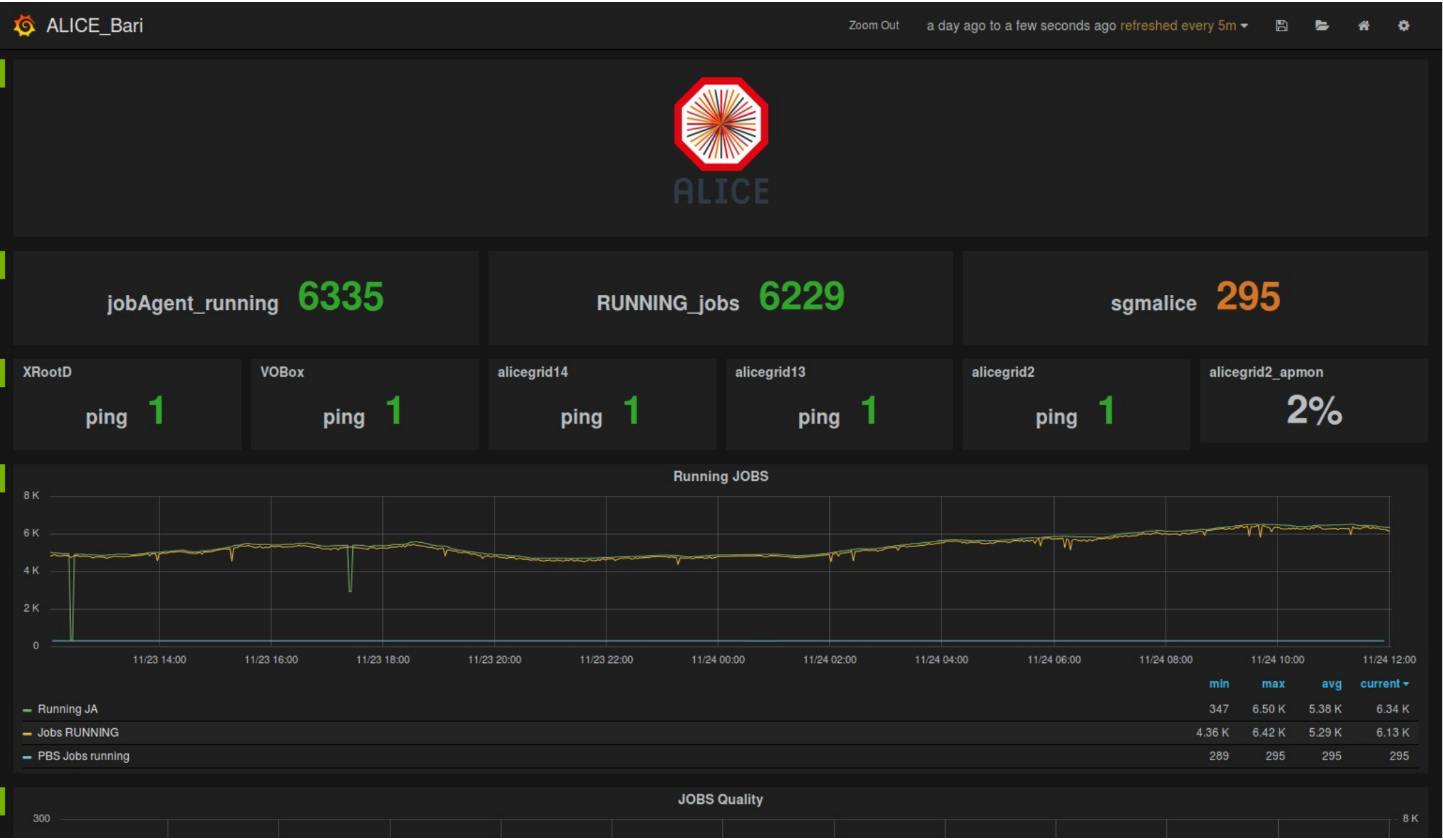
- No installation needed
- Script

```
#!/bin/bash
Host=”my-hostname”
Port=”8086”
User="my-username"
Password="my-password"
DBname="my-dbname"
MetricName=”temperature”
Timestamp=1445704538000
Value=26.0239284

data=$(echo [\{\"name\":\"${MetricName}\",\"columns\":[\"time\",\"value\"],\"points\":[[${Timestamp},${Value}]]\}])
curl -X POST 'http://${Host}:${Port}/db/${DBname}/series?u=${User}&p=${Password}'  --data ${data}
```

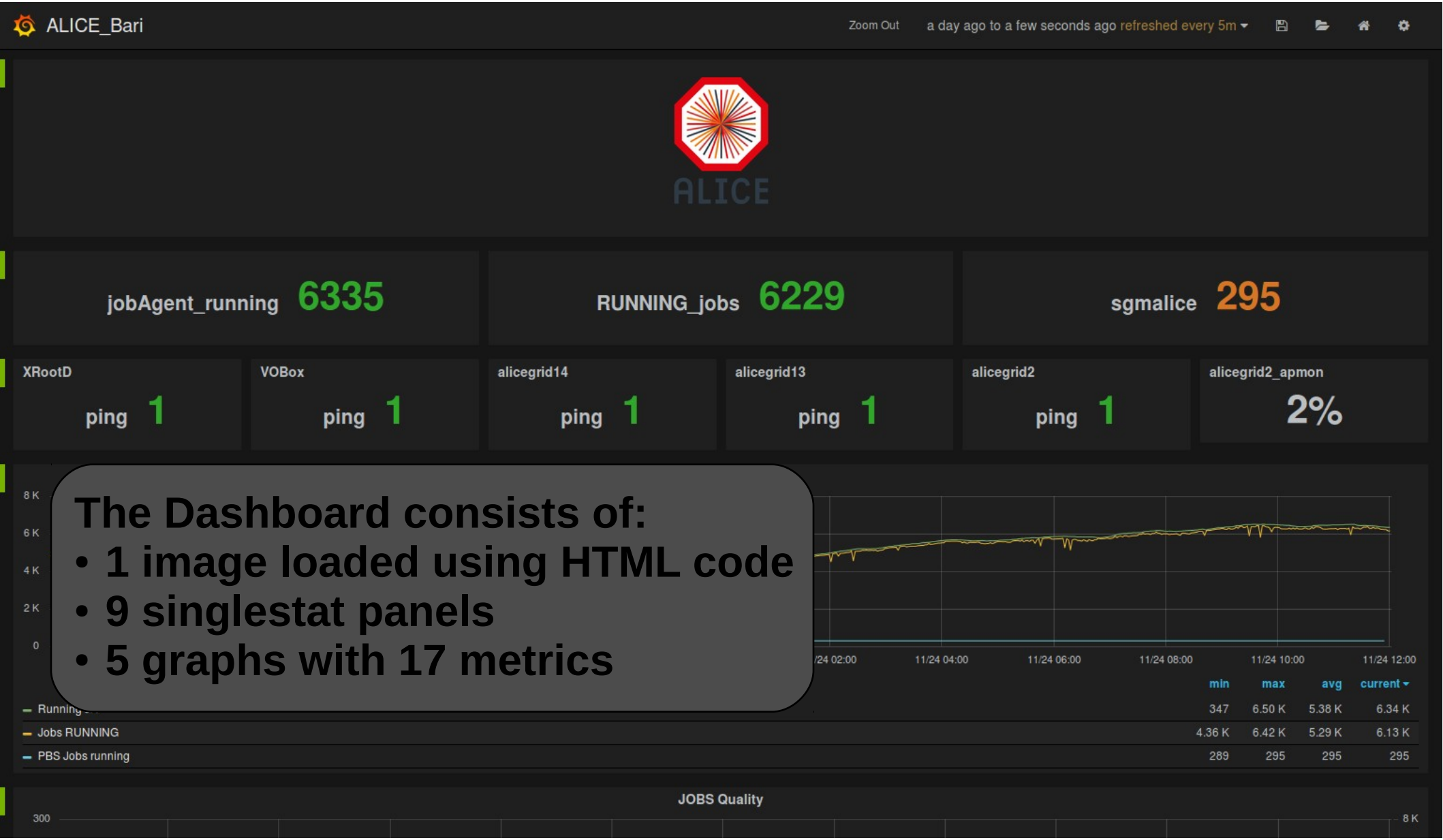
The Dashboard

ALICE::BARI Dashboard



The Dashboard

ALICE::BARI Dashboard



The Dashboard consists of:

- **1 image loaded using HTML code**
- **9 singlestat panels**
- **5 graphs with 17 metrics**

Performances

Extensively used along the last 14 months:

- Dashboard VM Characteristics:

VCPU	1
Memory	2 GB
Disk Space	22 GB
Operative System	Scientific Linux 6.5

- InfluxDB Performances:

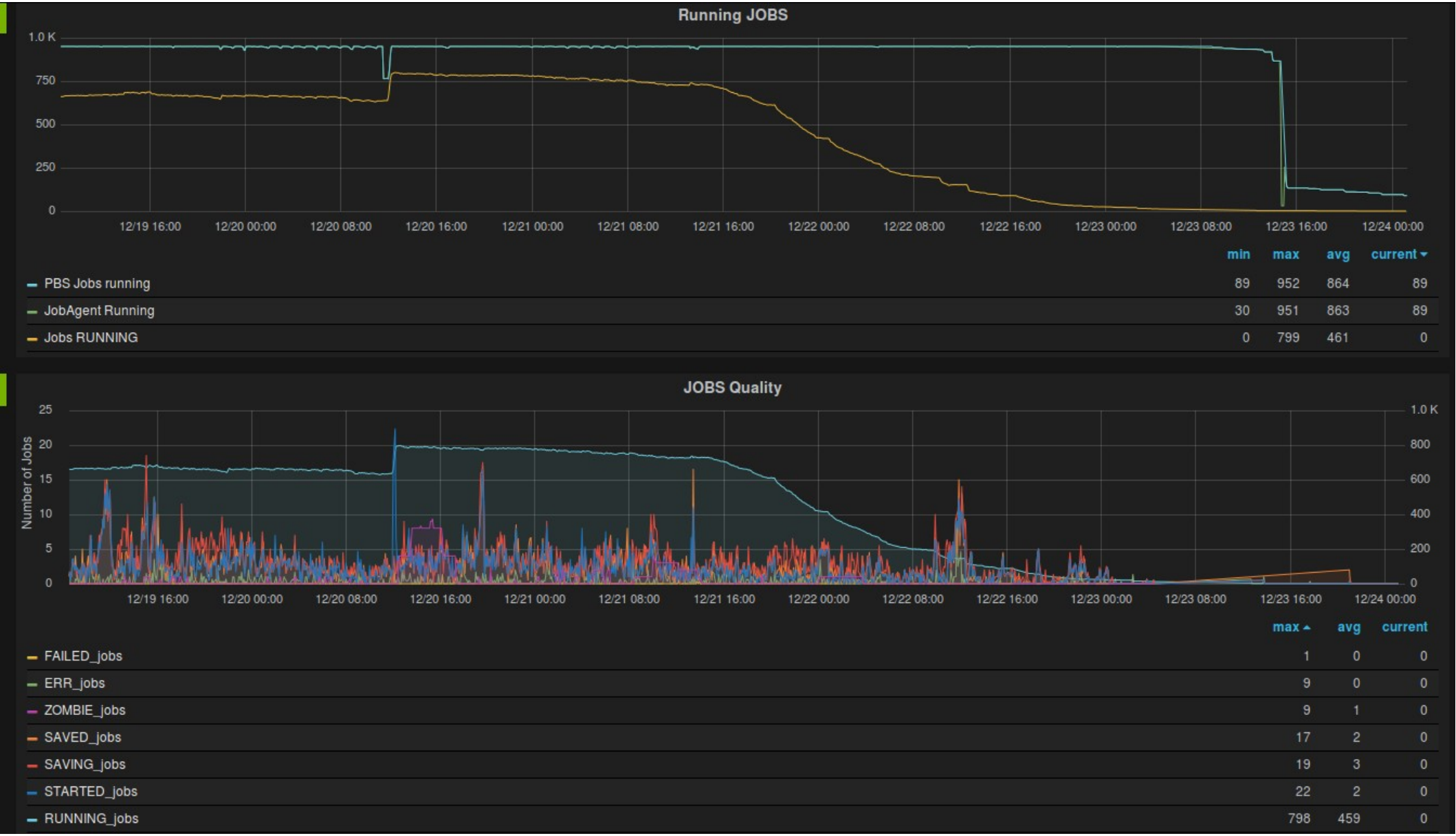
Parameter	Value
Metrics	203
Raw data information	20M values
Disk usage	400 MiB
Overall data retrieval time	~2000 sec
Daily aggregated data retrieval time	~200 sec for 50k values

- Overall Performances:

Time period	Aggregation step	Update time [sec]
1 day	1m	3
1 week	5m	6
1 month	30m	7
5 months	3h	24
1 year	3h	57

Performances

Diagnostic features



Performances

- A http server is been deployed to export the Dashboard
- The access to the Dashboard is enabled to everyone, from everywhere and with every “smart” device by using the url:

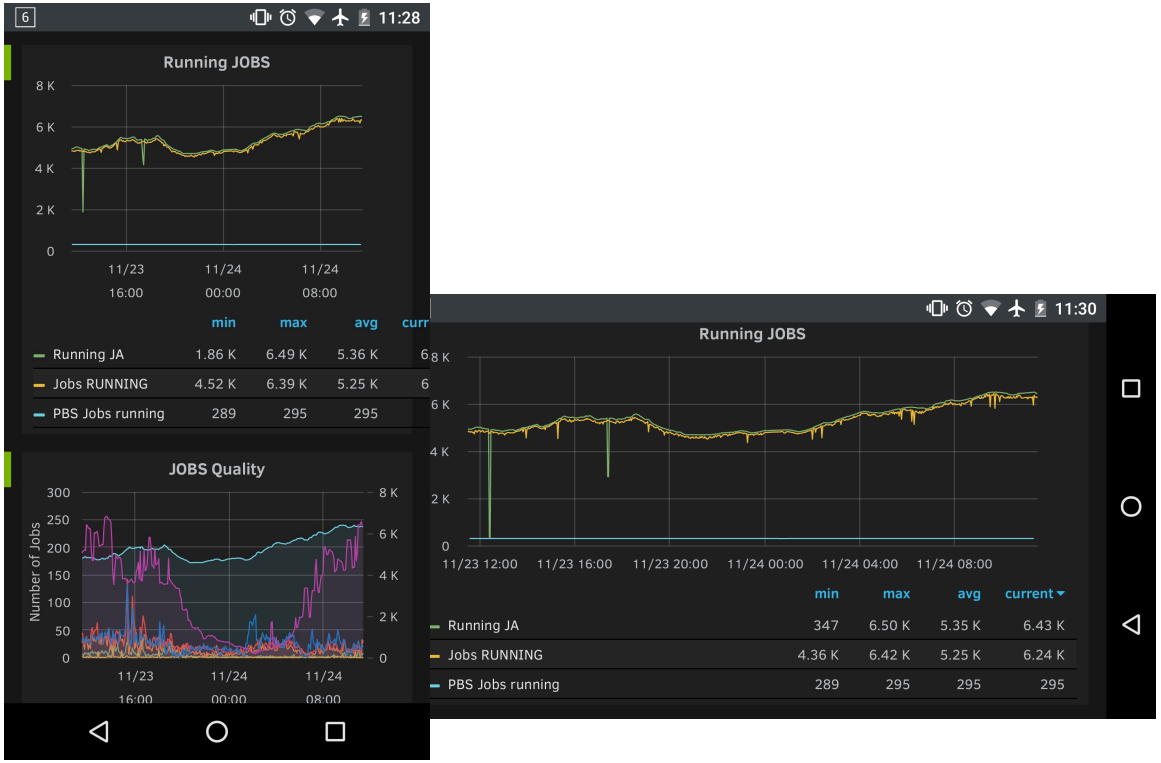
http://90.147.170.102/#/dashboard/db/alice_bari

Performances

- A http server is been deployed to export the Dashboard
- The access to the Dashboard is enabled to everyone, from everywhere and with every “smart” device by using the url:

  http://90.147.170.102/#/dashboard/db/alice_bari

- ...also using a smartphone:



Performances

Let's try!

Outlook

- Use **ApMon** to import data from MonALISA
- Port the Dashboard suite to **containers**
- Allow authentication via **AAI – Apache**
- Implement further sensors to import data from the most popular **data sources**
- Extend the facility to the other **Italian sites**
- Create a dashboard for the **Italian ALICE computing**

Conclusions

- The developed dashboard is a powerful tool for statistical and debugging purposes
- Currently used since more than one year to monitor the ALICE activity in the Bari Tier2
- It allows debugging activities by **cross-checking** info coming from different data sources
- It **adds** local site information to MonALISA
- Can be used to monitor **other experiments or projects**.

Thank you

for your attention!