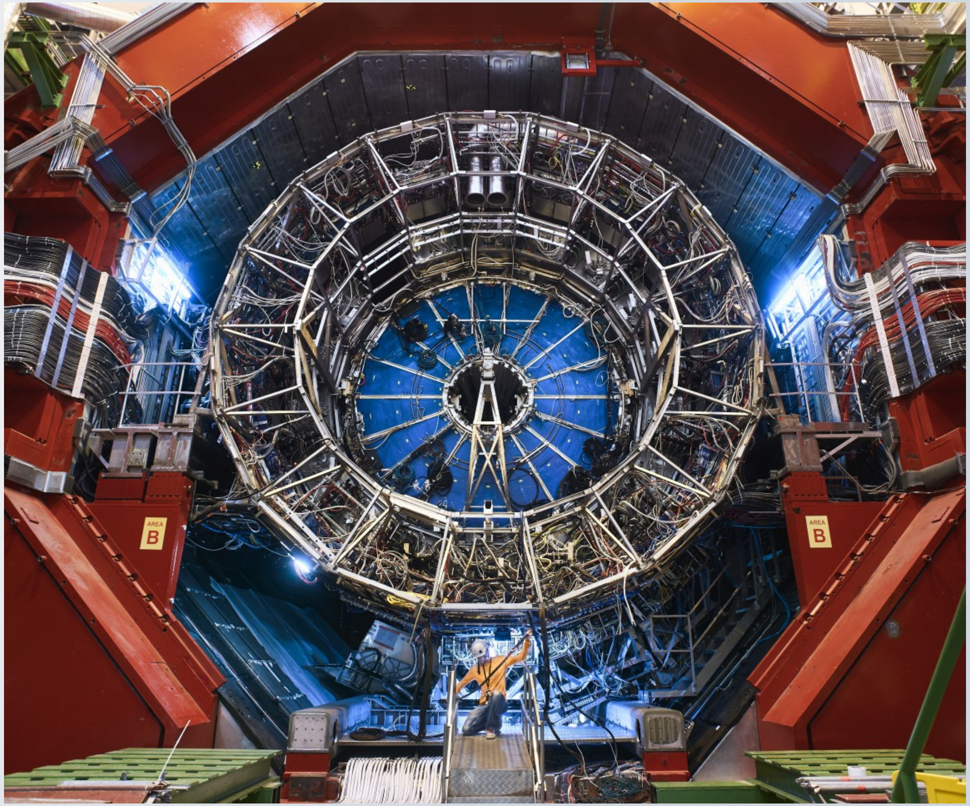
The ALICE data quality control system

04.11.2019

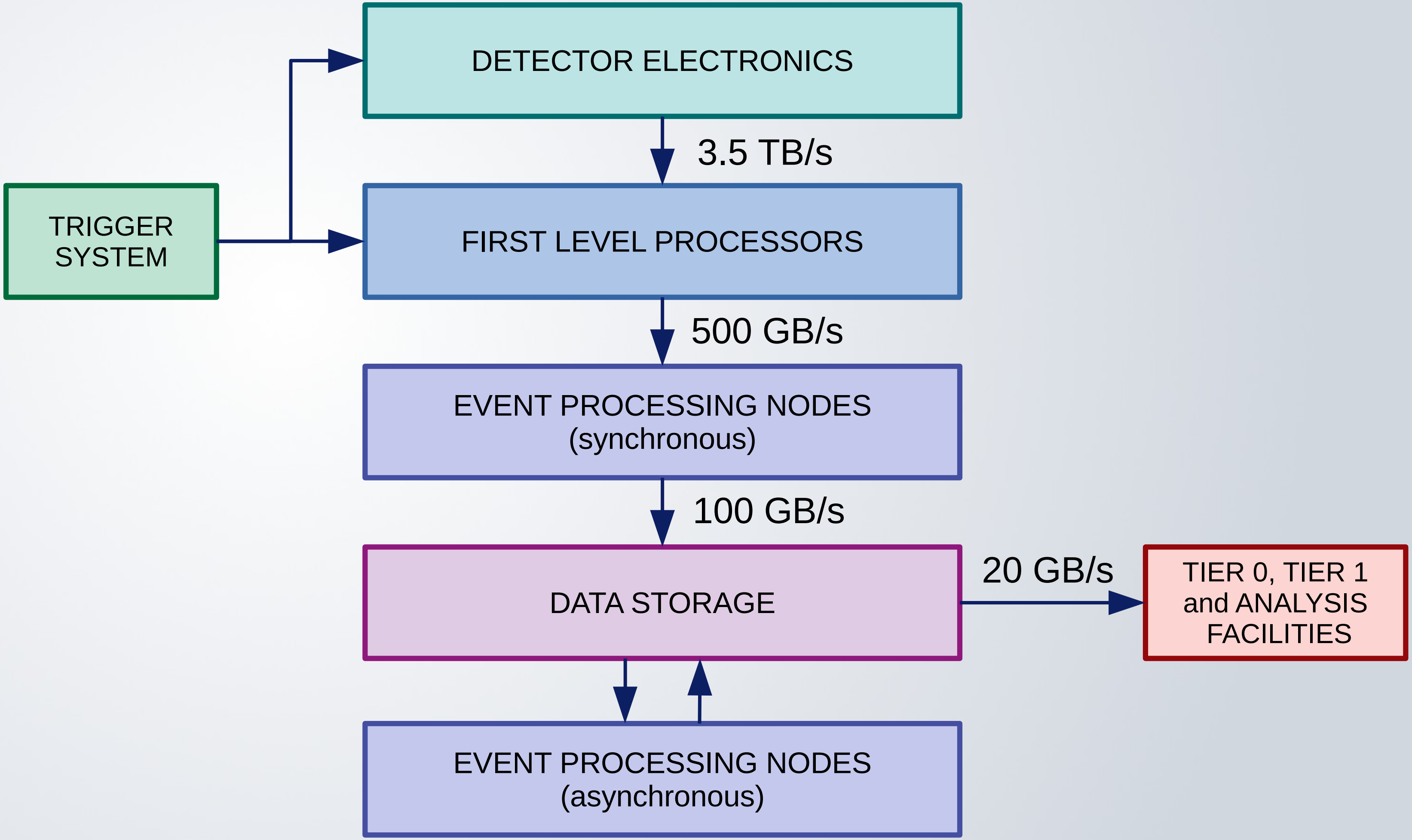
Piotr Konopka, CERN / AGH University of Science and Technology
Barthélémy von Haller, CERN
for the ALICE O^2/FLP project

ALICE 2.0

- New and upgraded detectors – higher tracking resolution, supporting higher collision rates
- Continuous readout
- Much more data:
  - Before: 40 GB/s input, 10 GB/s output
  - After: 3500 GB/s input, 100 GB/s output



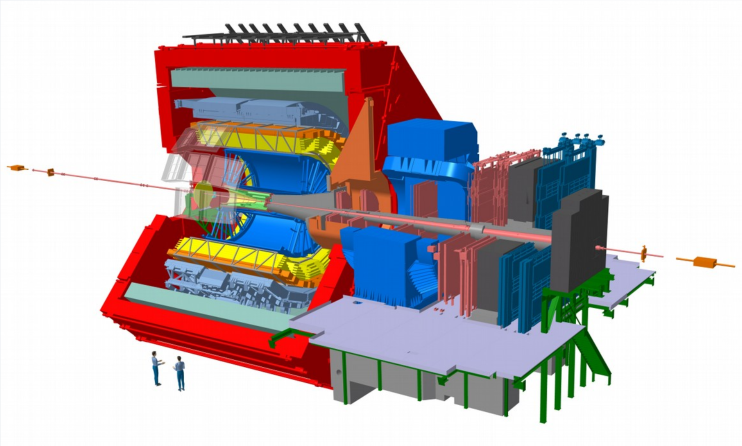
O^2 – the Online-Offline Computing System



O^2 - characteristics

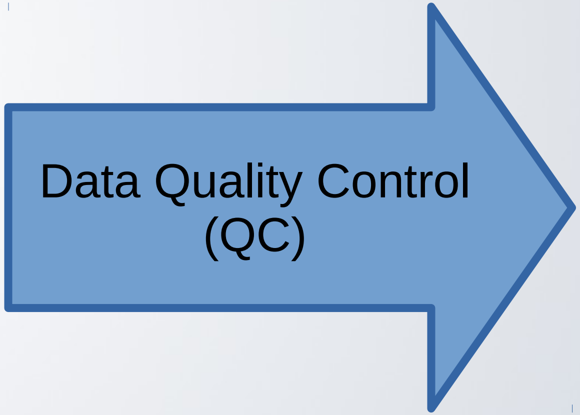
- Two major computing layers
- Highly heterogeneous
- Raw data replaced by processed data
- Framework – Data Processing Layer and FairMQ
  - Message–based system
  - Zero copy approach in the main processing flow

Data Quality Control in principle



\+

```
A few million lines
      of code
```




```
Quality::Good

Quality::Medium

Quality::Bad
```

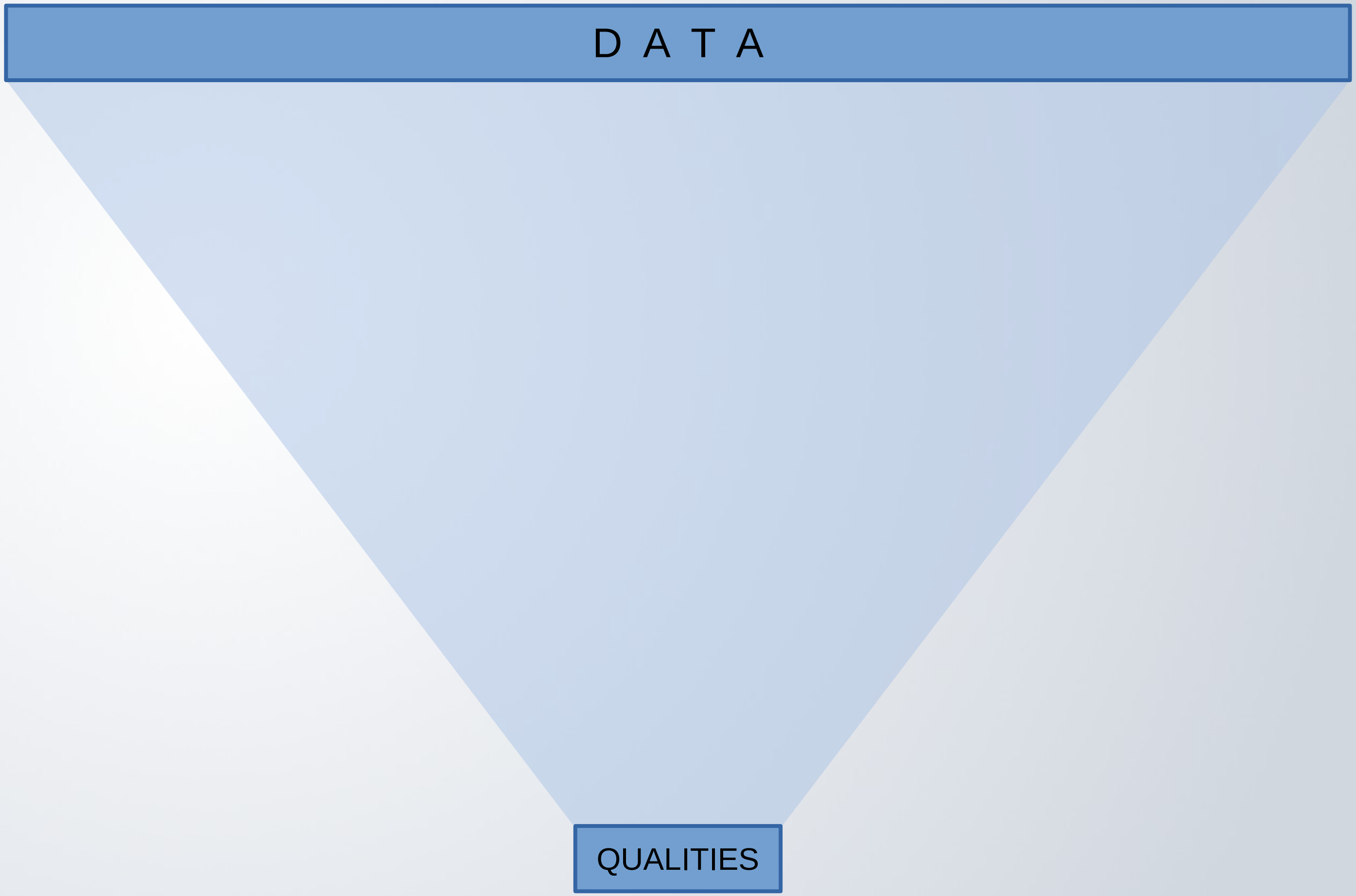
Lessons from the past

- Previously we had separate systems:
  - Data Quality Monitoring (DQM) for online processing
  - Quality Assurance (QA) for offline processing
- Different set of tools, but in the end both were used for online and offline data quality control
- Quality Control (QC) is both DQM and QA

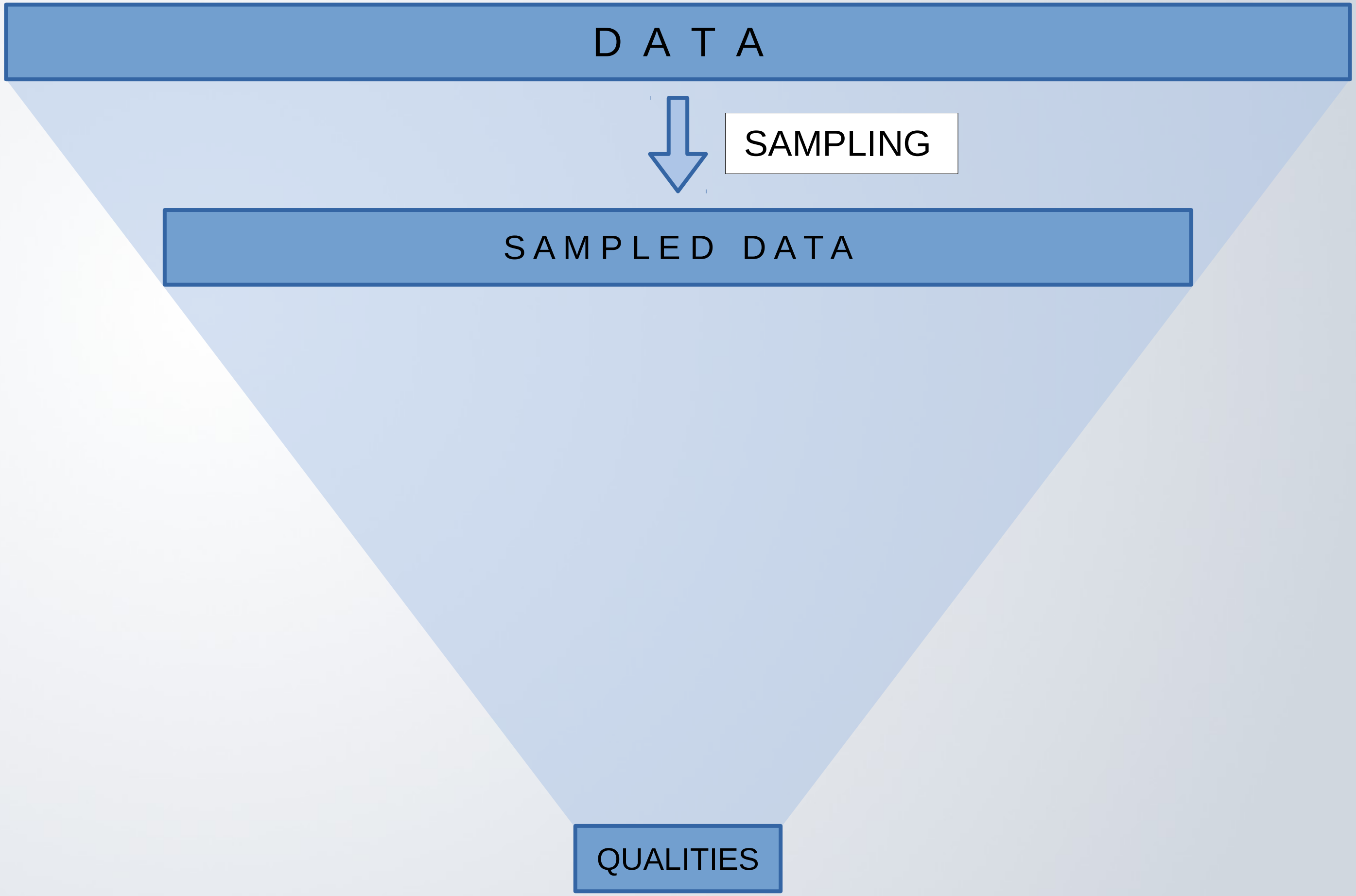
Challenges of the data Quality Control in ALICE

- Unification of the online and offline worlds
- Raw input data not preserved
  - Even higher importance of a reliable data quality control
  - Very large amount of data to look after (3.5 TB/s)
- ~15 detector teams with different use cases
- Around 100 Quality Control algorithms expected

Quality Control chain

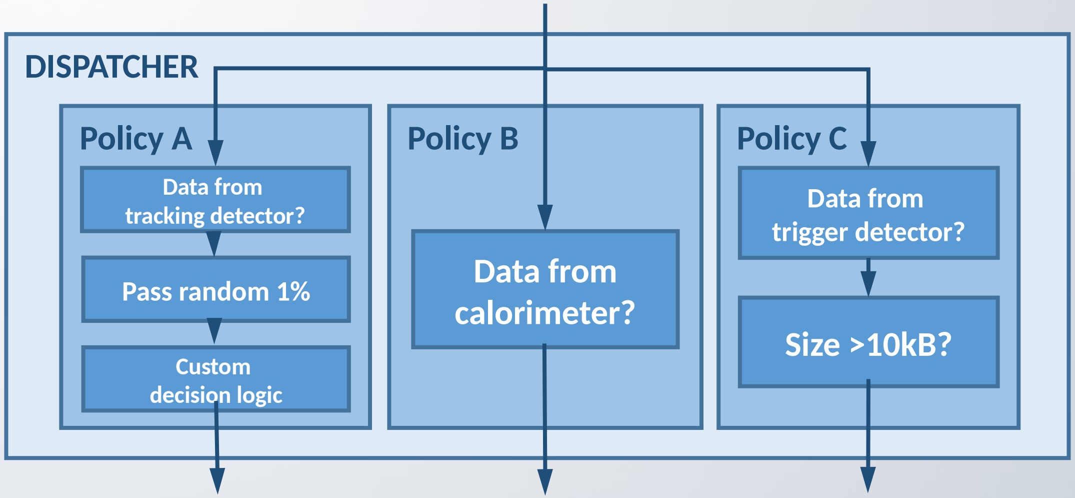


Quality Control chain

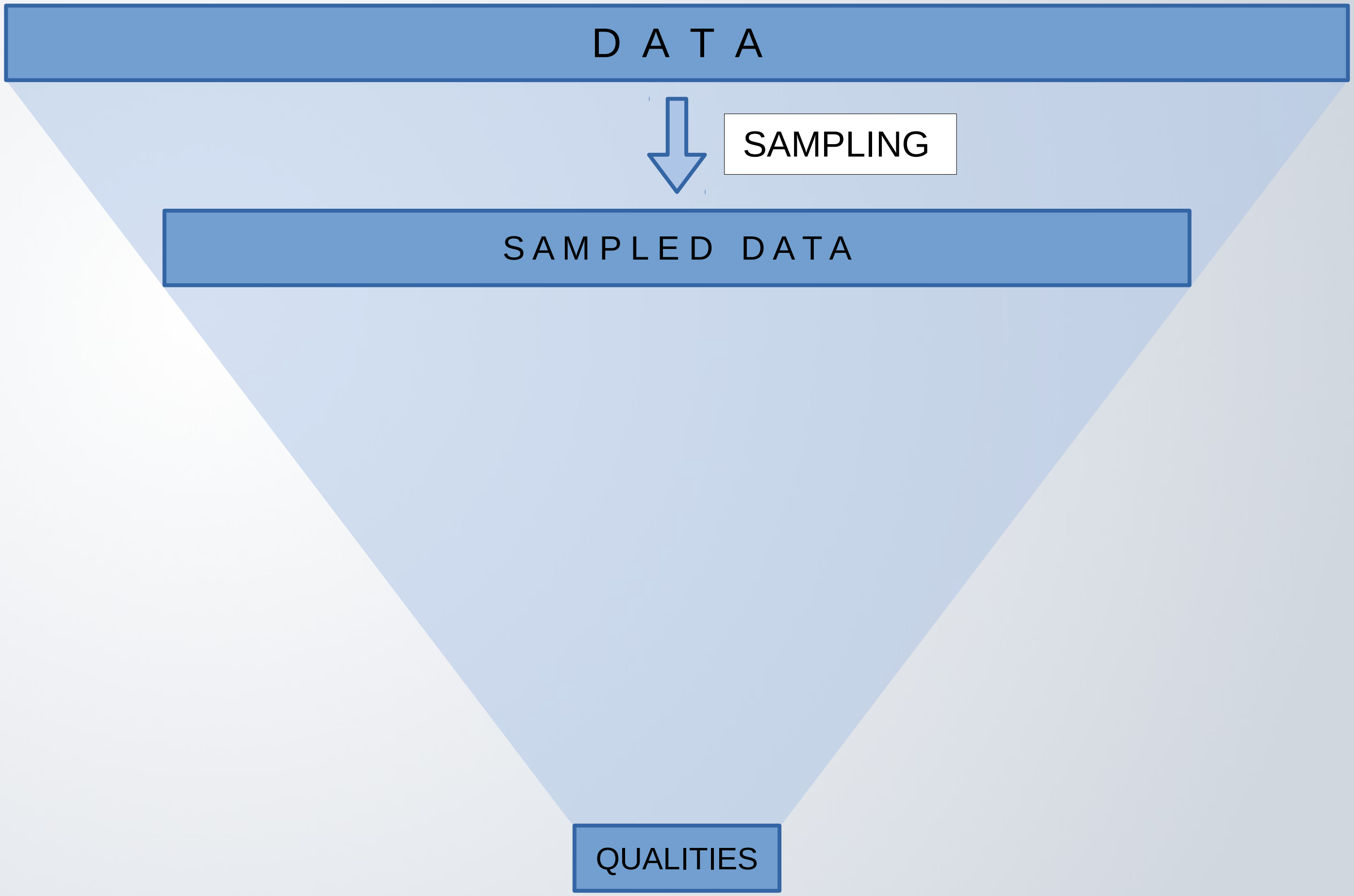


Data Sampling – what can it do?

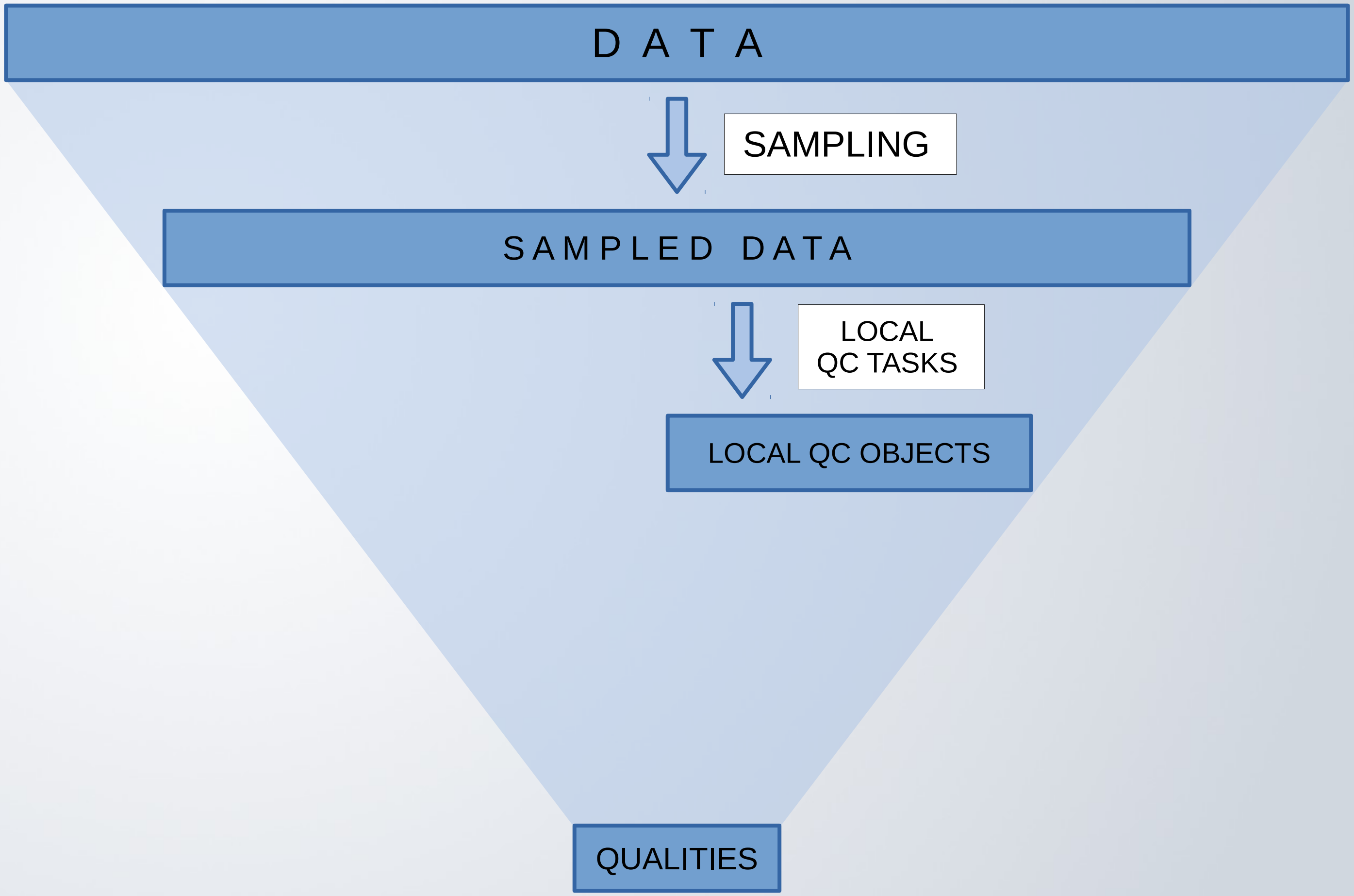
- Configurable sampling policies which can spy on any data
- Pseudo–random sampling of parallel data distributed among many machines
- Custom filtering as a plugin system
- Reconfiguration during the data taking



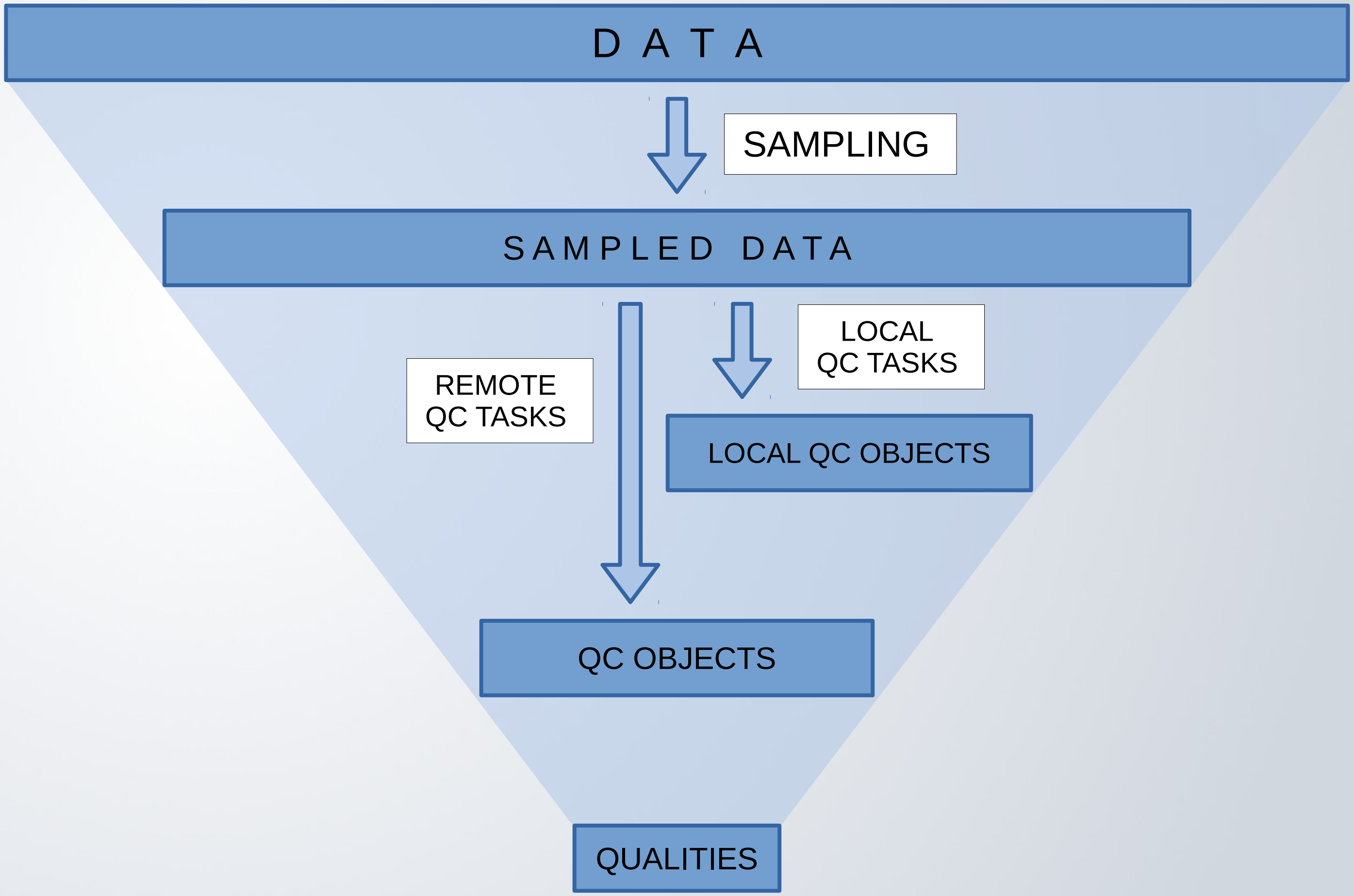
Quality Control chain



Quality Control chain

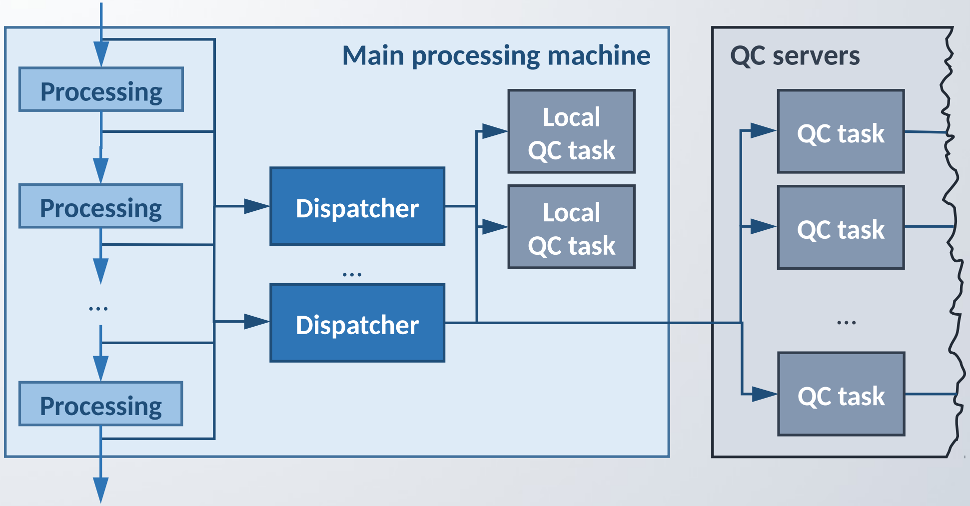


Quality Control chain

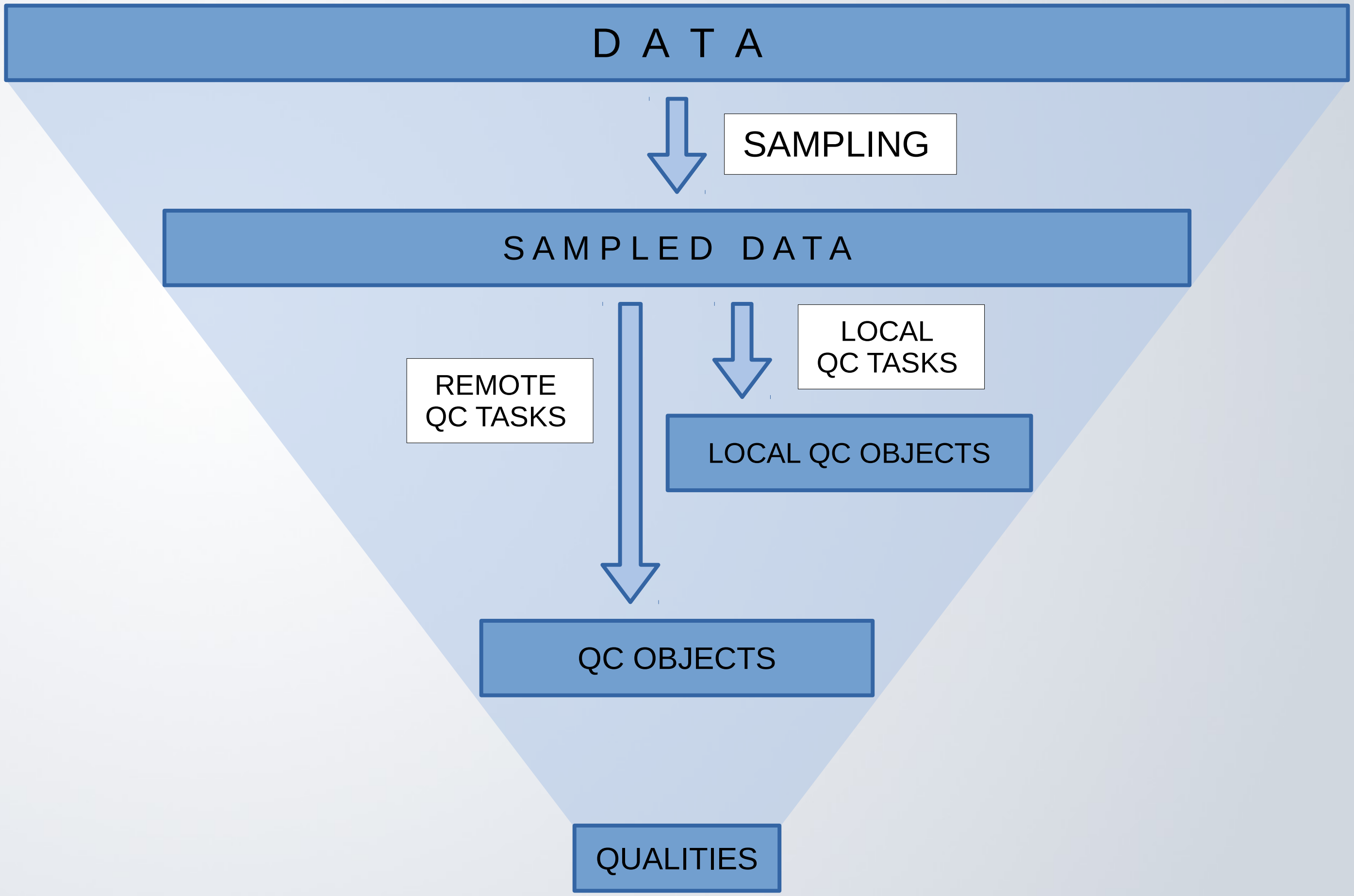


Quality Control Tasks

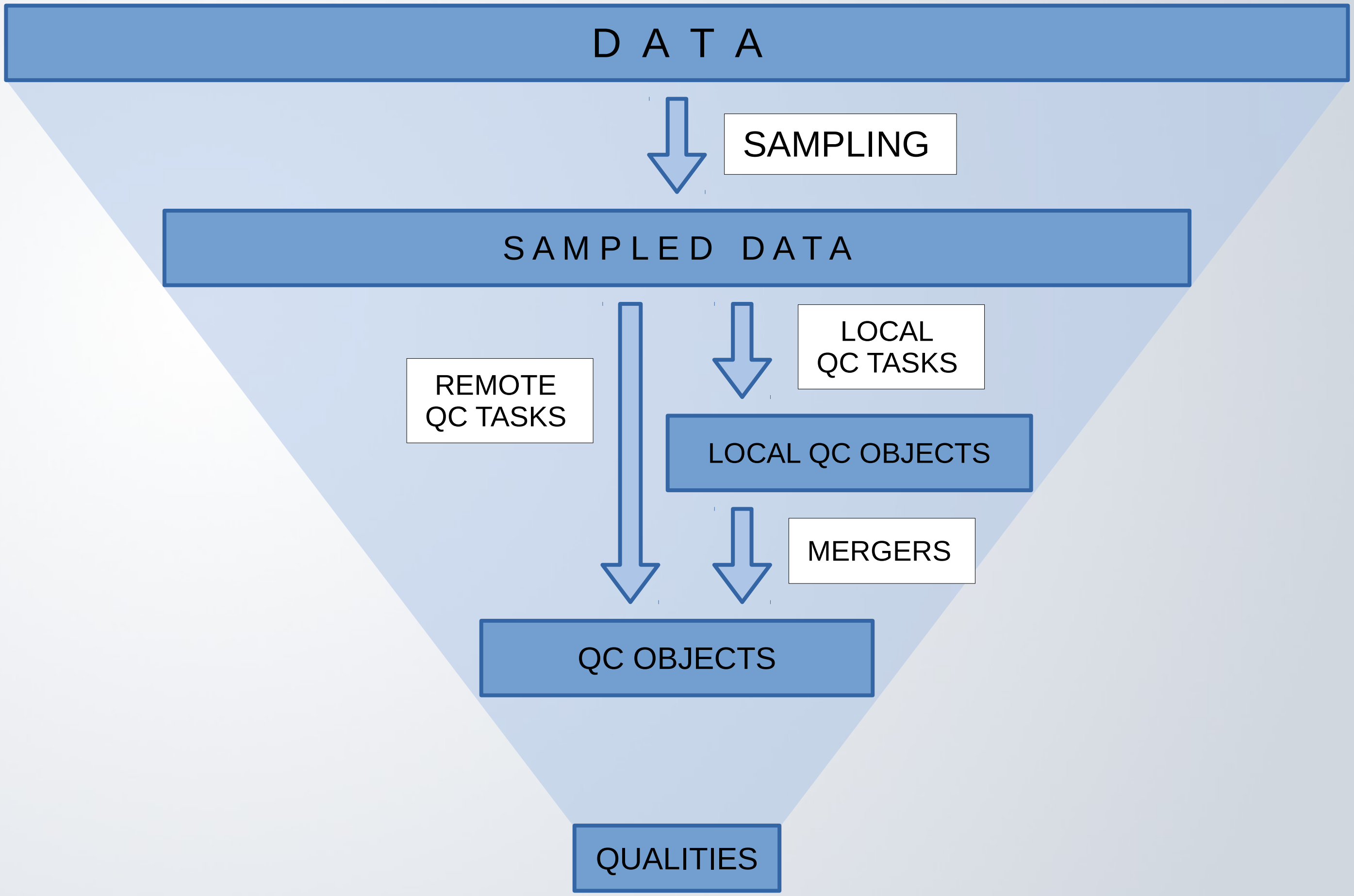
- User algorithms being executed under a common interface
- One or more QC objects generated, mostly ROOT objects
- Can run on the same as the main processing (locally) or on dedicated QC servers (remotely)



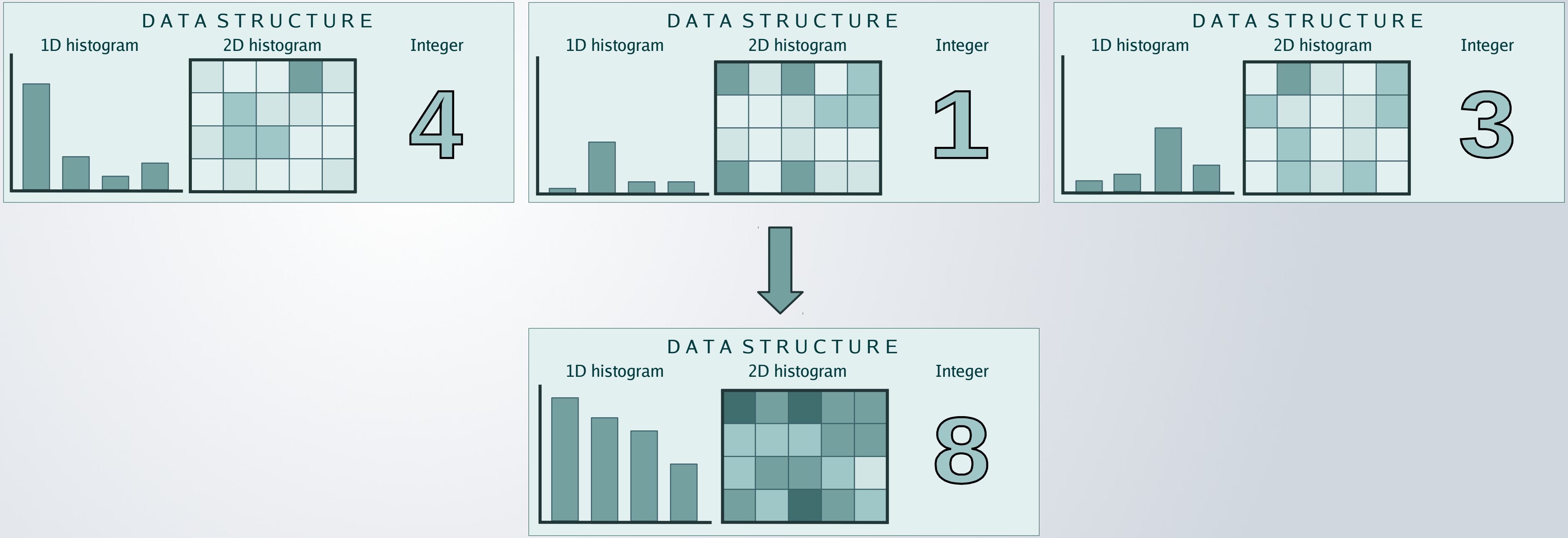
Quality Control chain



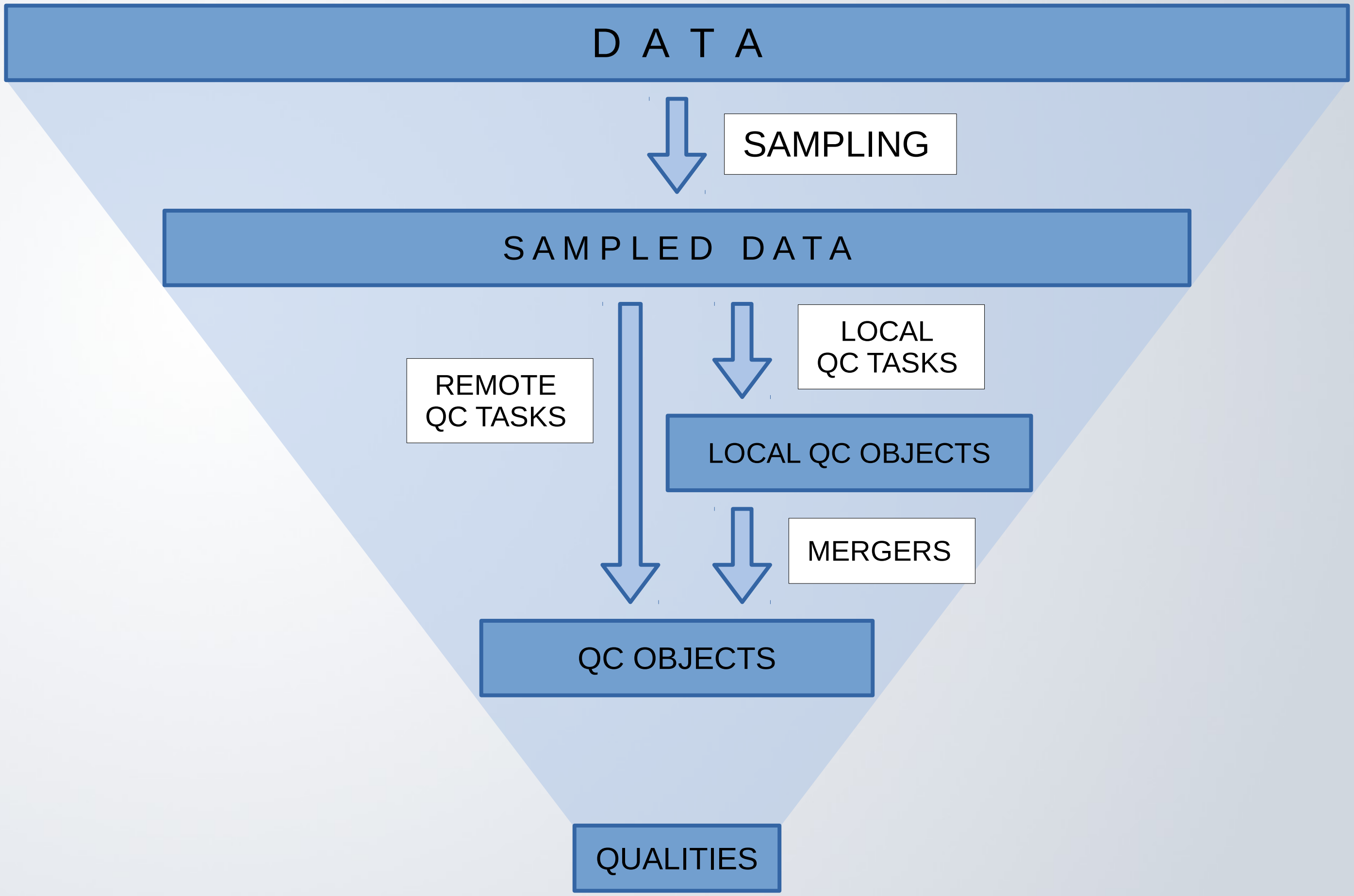
Quality Control chain



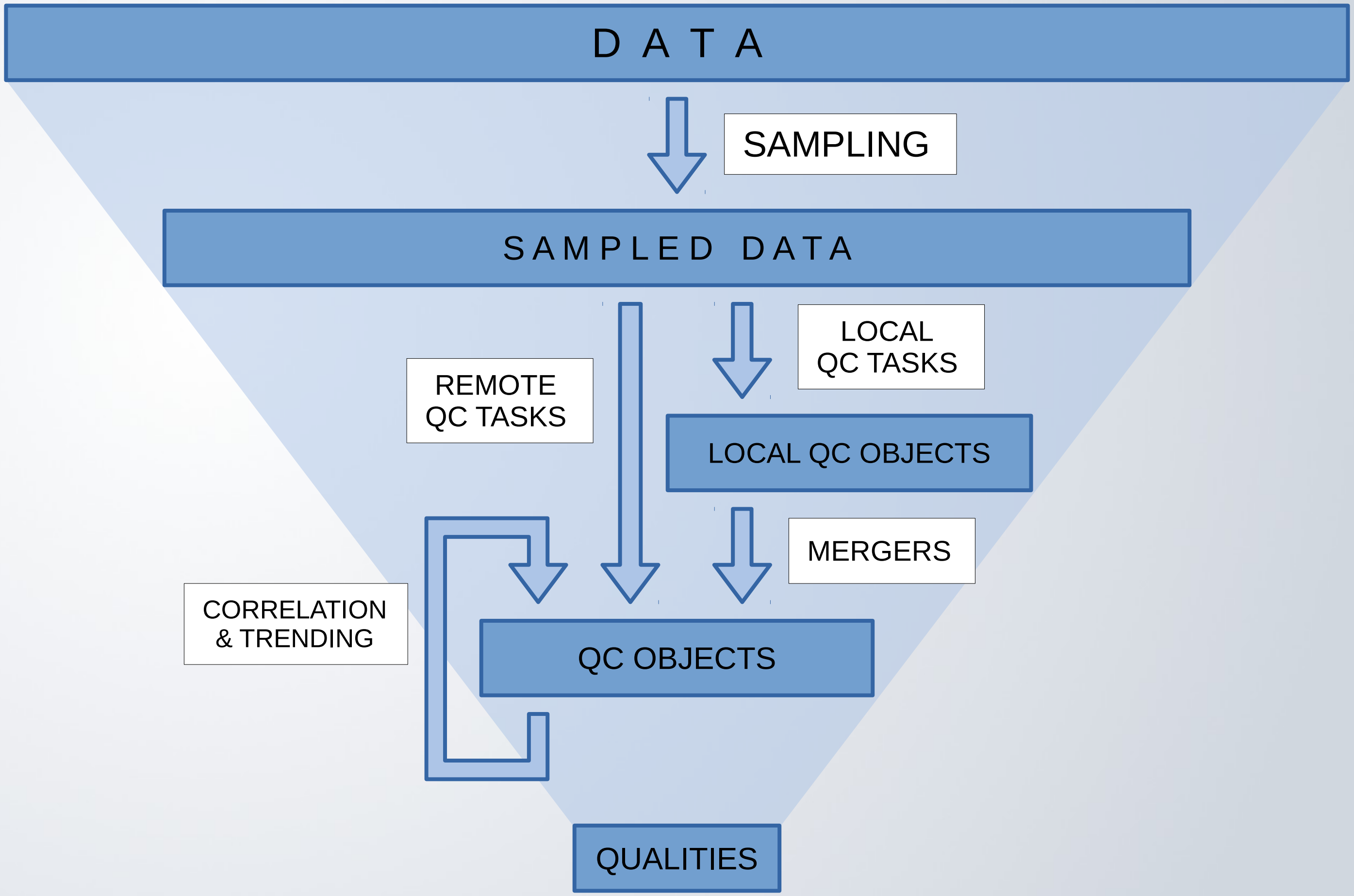
Mergers in principle



Quality Control chain



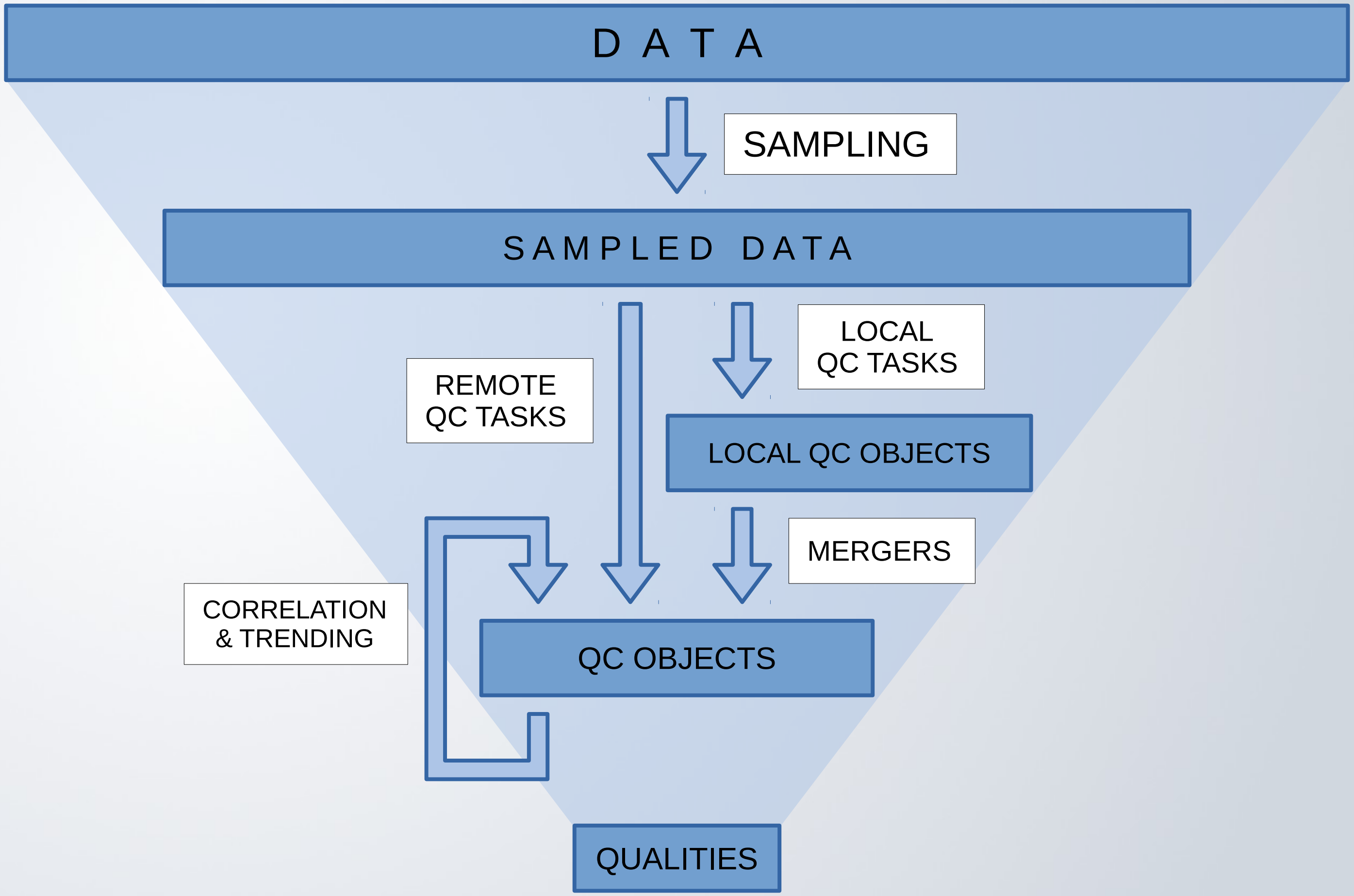
Quality Control chain



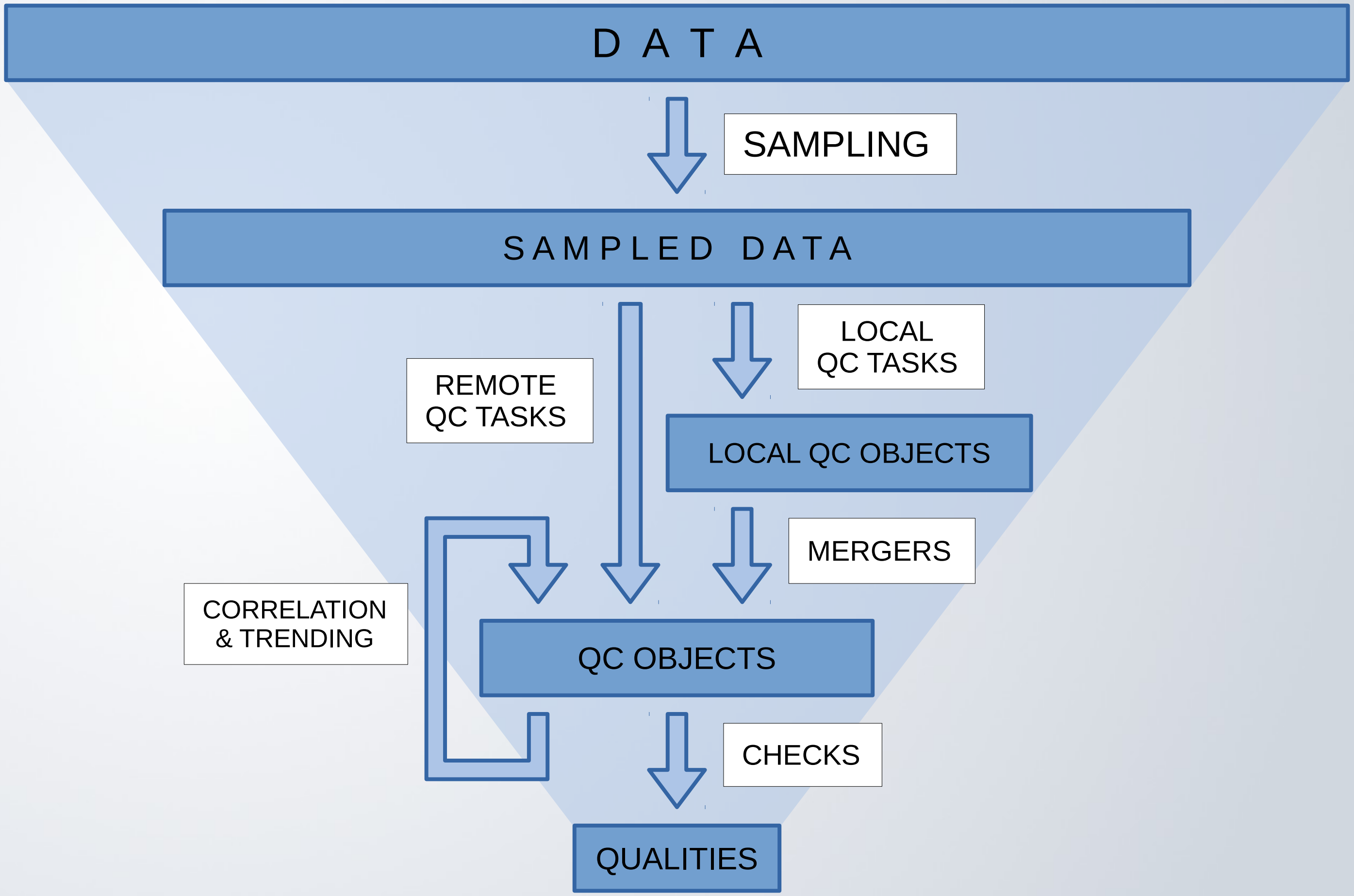
Postprocessing, Correlation and Trending

- Any task running asynchronously to the main data flow
- Input data are anything generated by QC
- Usually correlation or trending of specific values
- Triggered periodically, manually or on certain events (start of run, end of run, etc.)

Quality Control chain

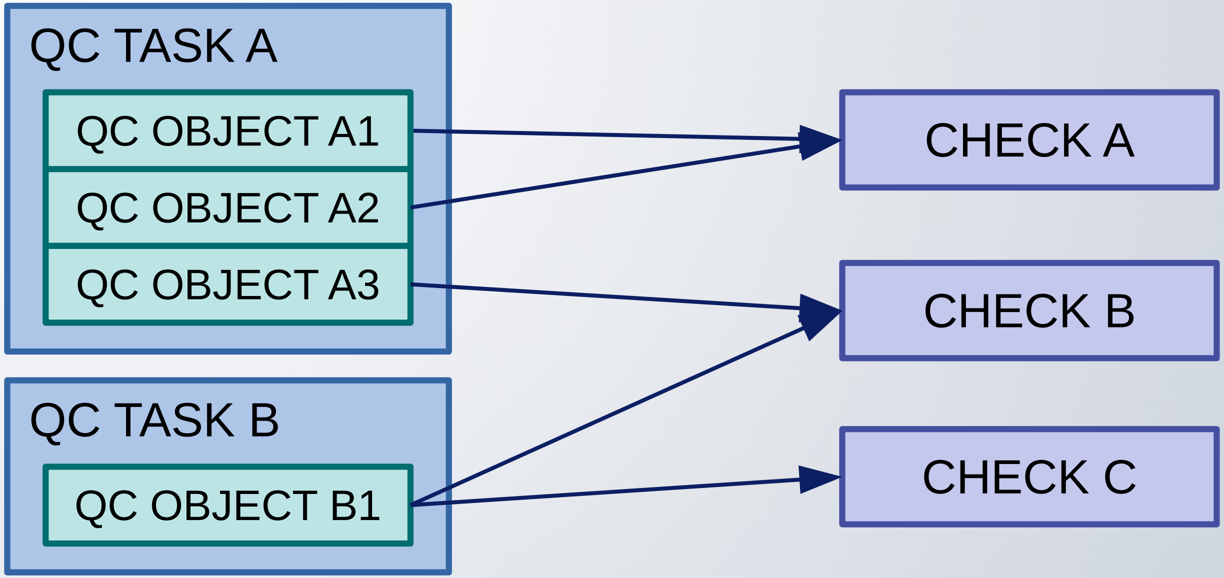


Quality Control chain

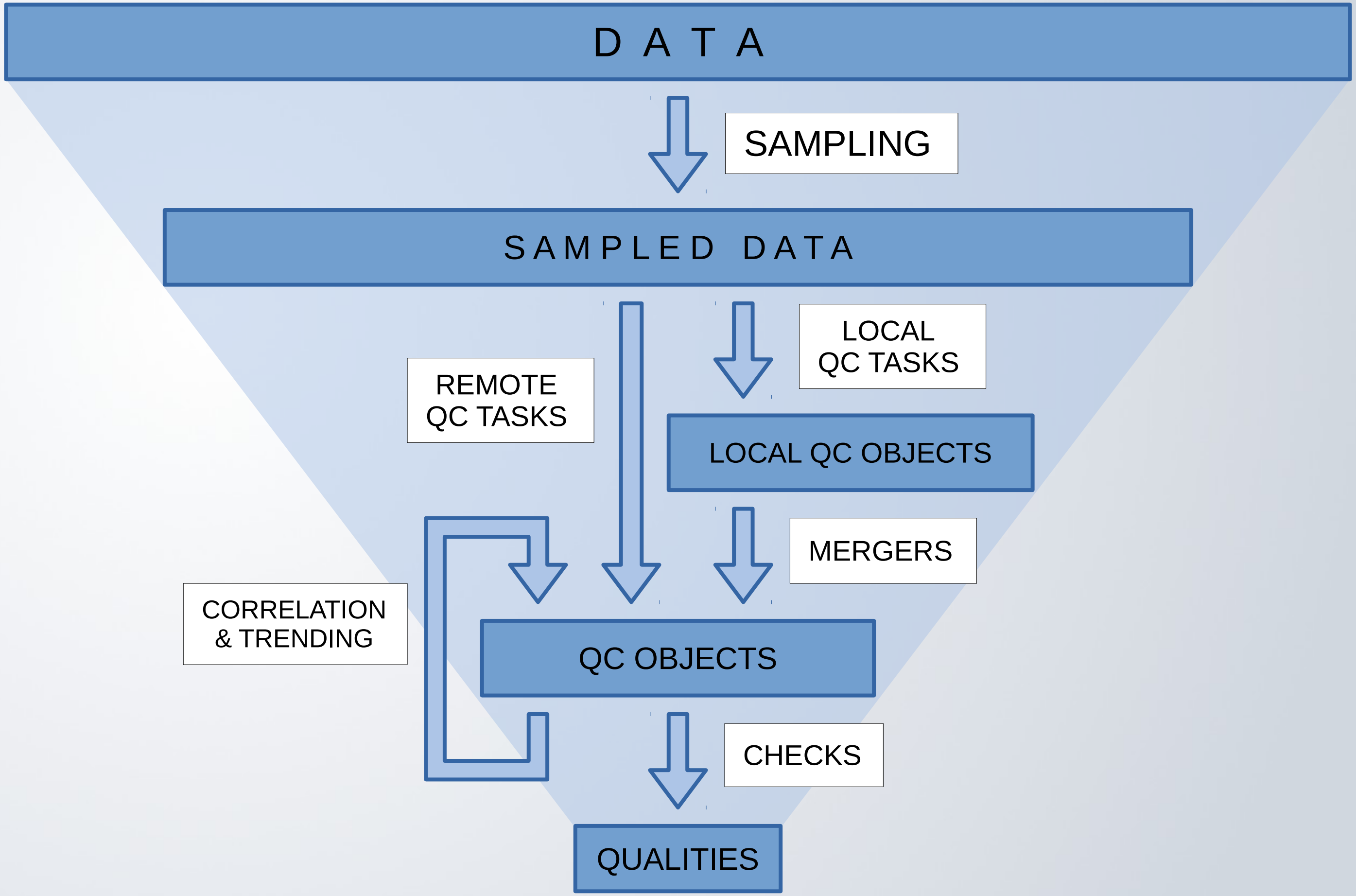


Checks

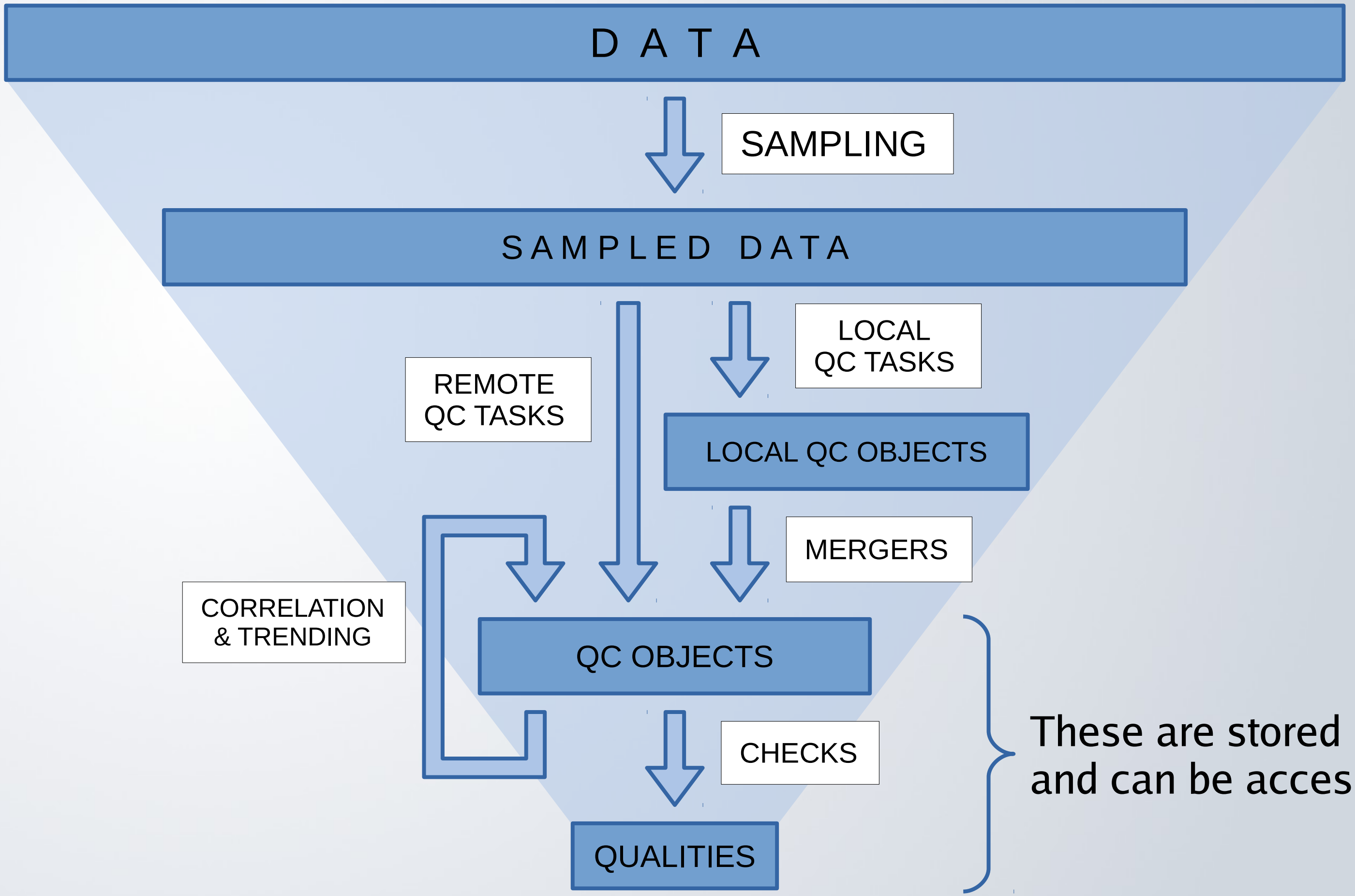
- User algorithms under a common interface
- Should return Qualities, optionally with a comment
- Usage of Machine Learning is currently investigated



Quality Control chain

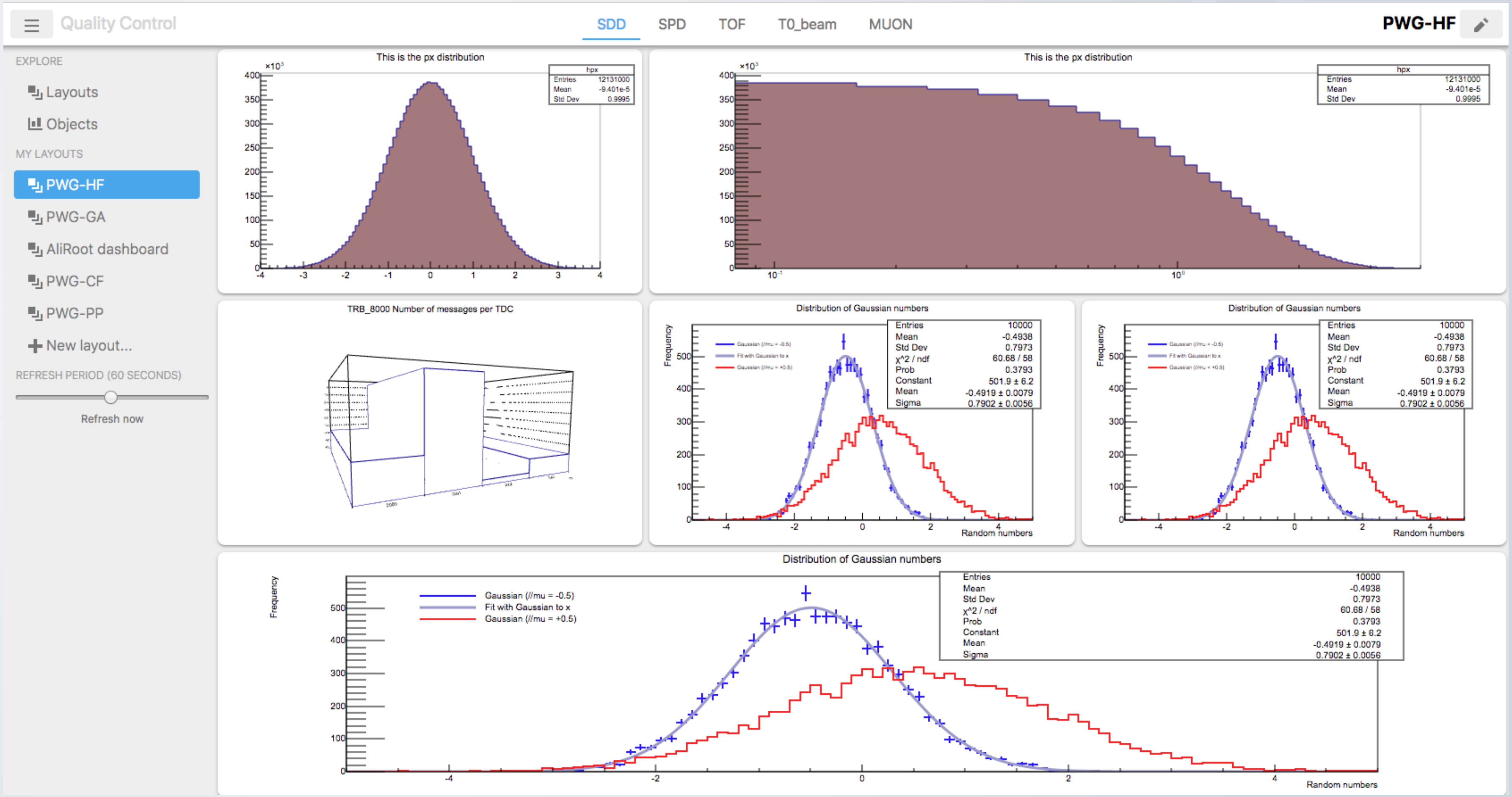


Quality Control chain



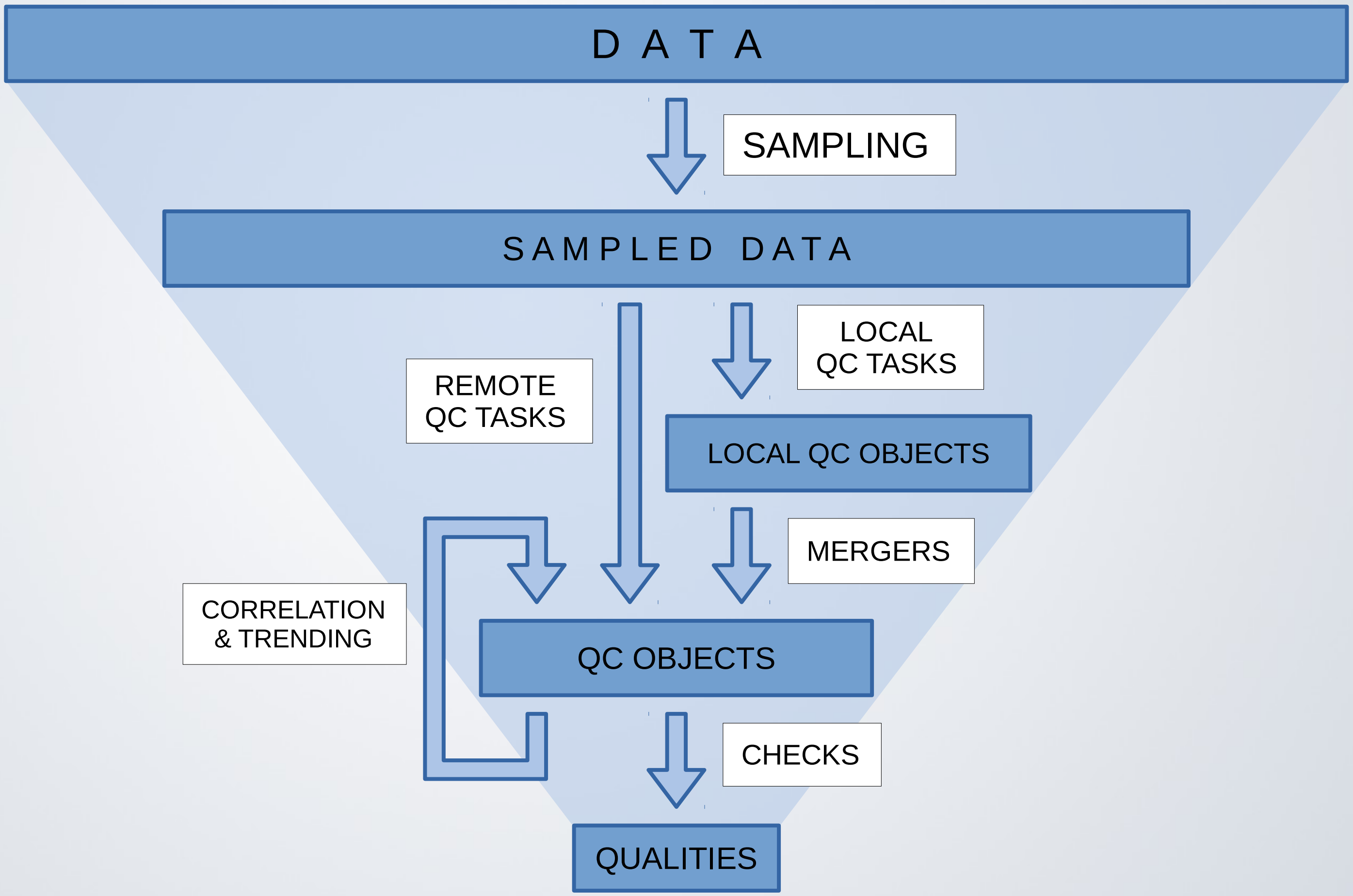
These are stored in a database and can be accessed in GUI

Quality Control GUI (QCG)

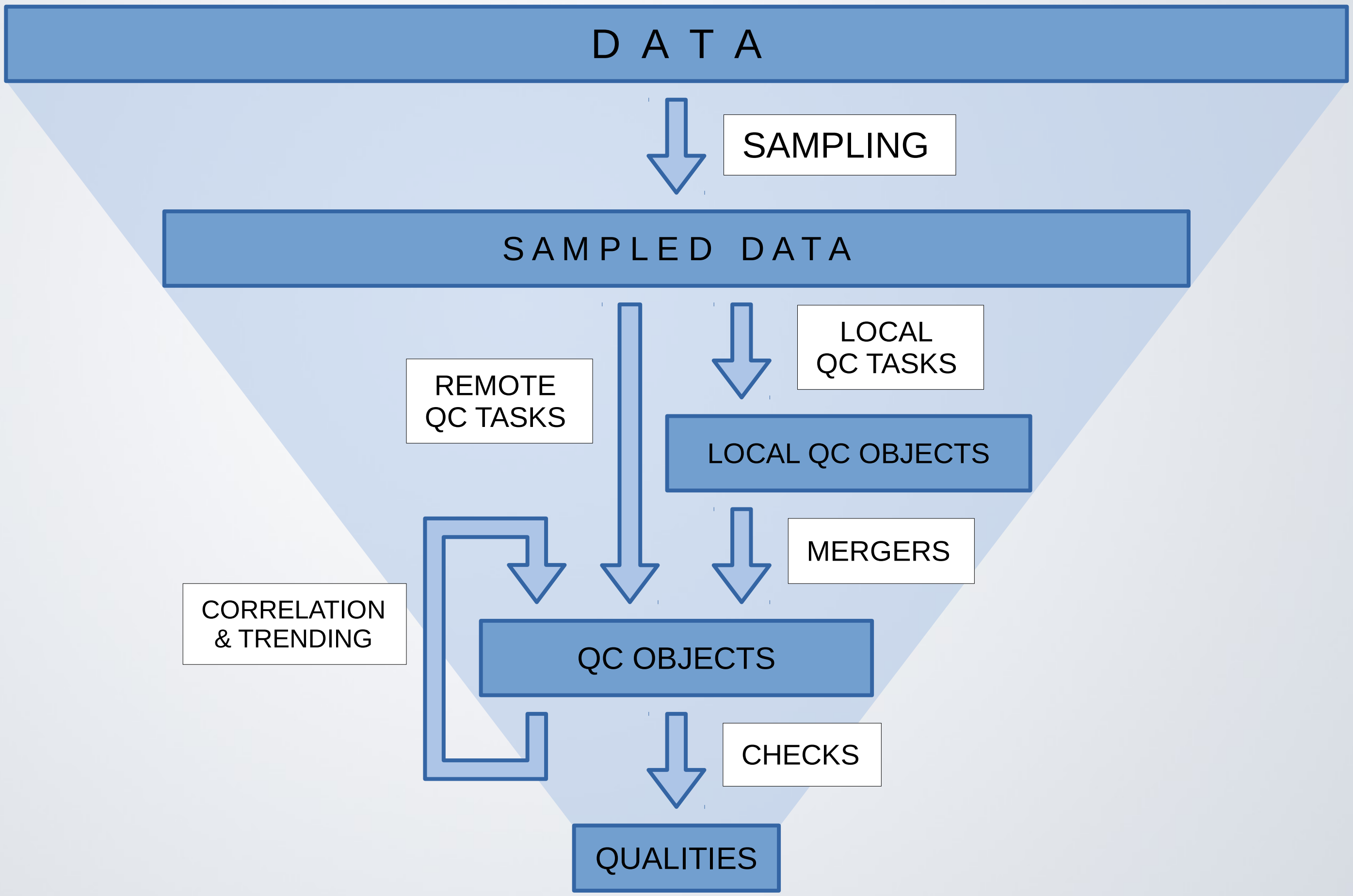


Adam Węgrzynek, George Raduta, Vladimir Kosmala

Quality Control chain – data rates

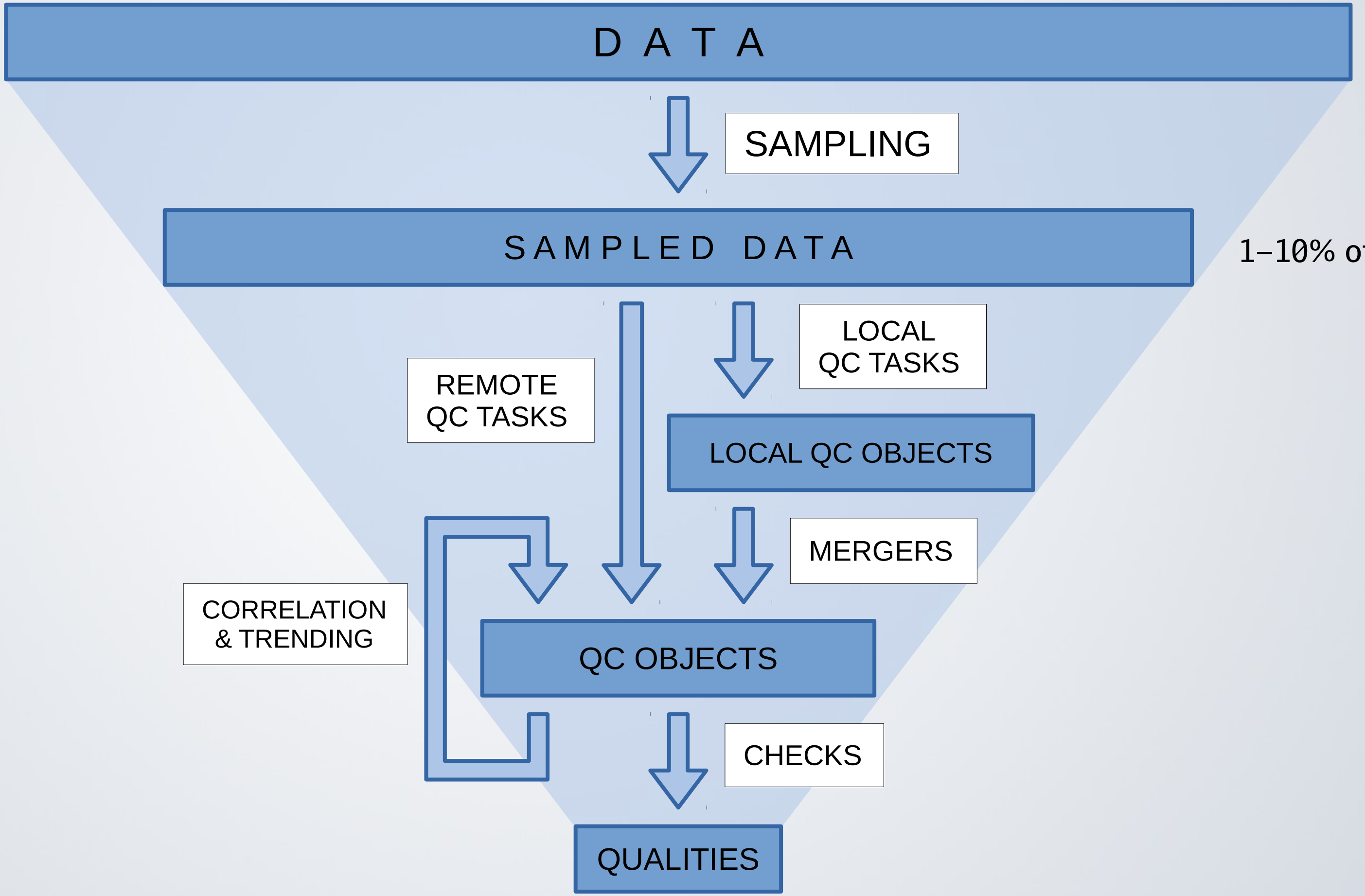


Quality Control chain – data rates



3.5 TB/s of raw data +
temporary and permanent derivative data

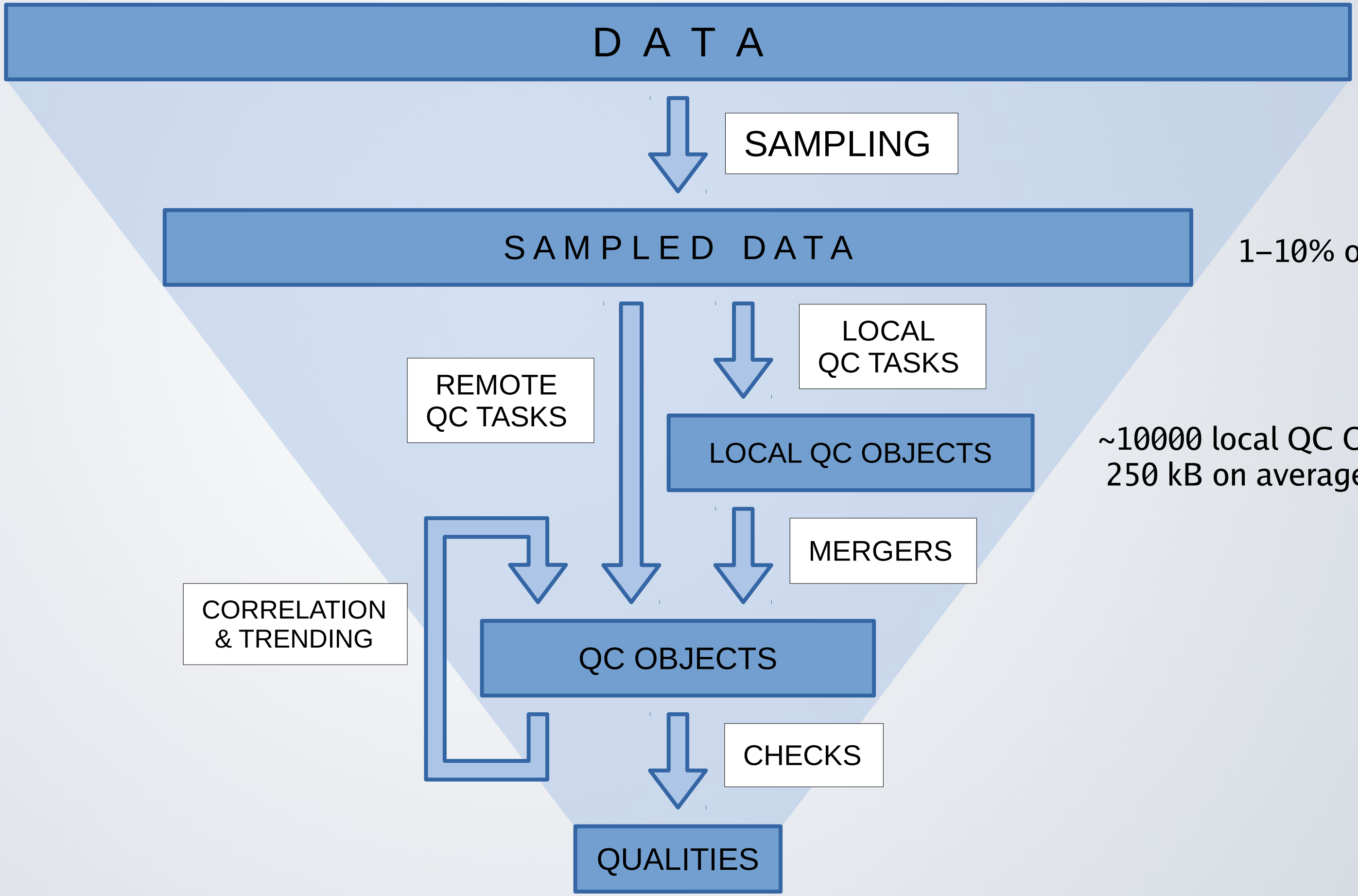
Quality Control chain – data rates



3.5 TB/s of raw data + temporary and permanent derivative data

1–10% of data, 100% when needed

Quality Control chain – data rates

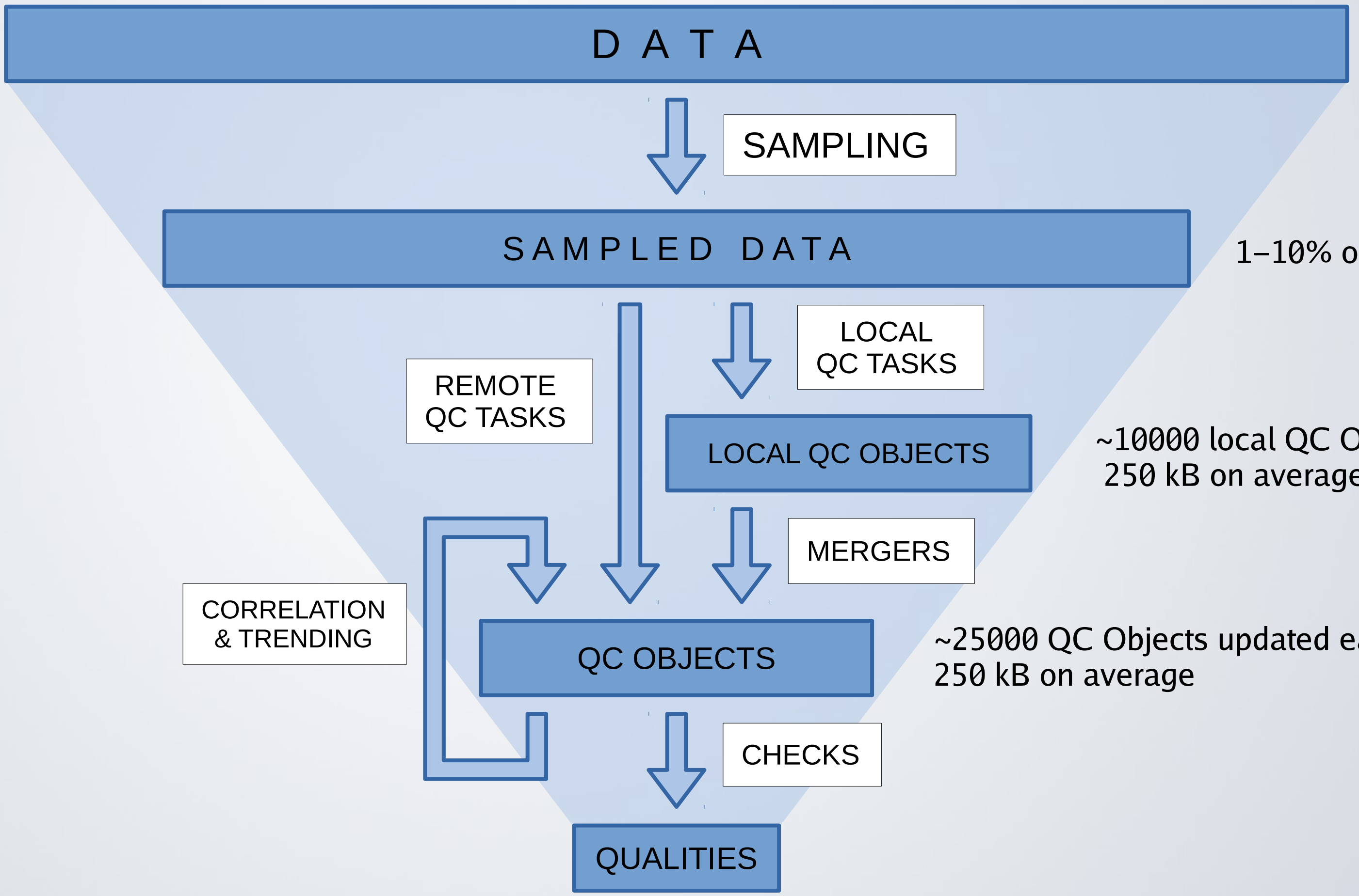


3.5 TB/s of raw data +
temporary and permanent derivative data

1–10% of data, 100% when needed

~10000 local QC Objects updated each minute,
250 kB on average

Quality Control chain – data rates



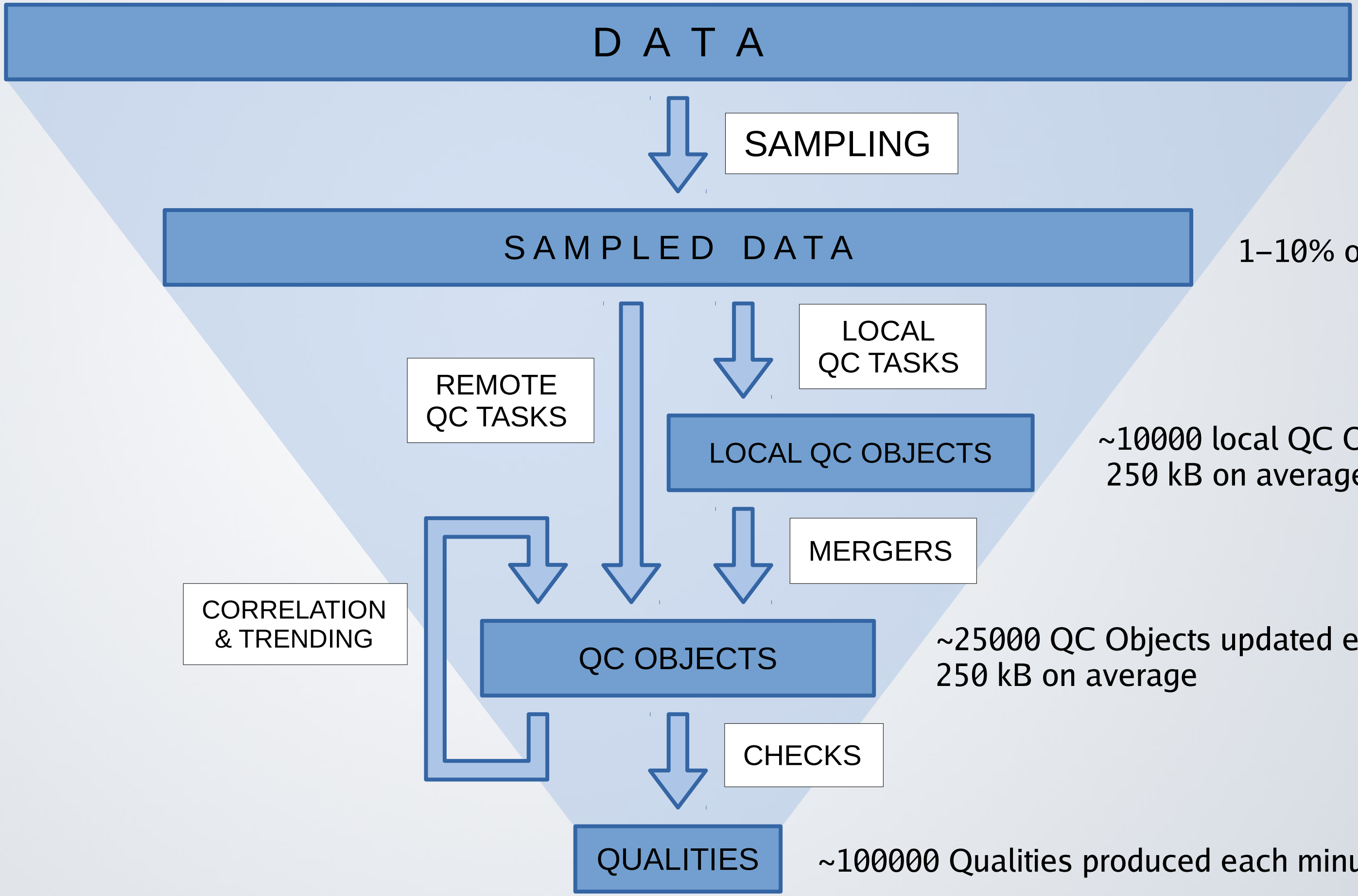
3.5 TB/s of raw data + temporary and permanent derivative data

1–10% of data, 100% when needed

~10000 local QC Objects updated each minute, 250 kB on average

~25000 QC Objects updated each minute, 250 kB on average

Quality Control chain – data rates



3.5 TB/s of raw data + temporary and permanent derivative data

1–10% of data, 100% when needed

~10000 local QC Objects updated each minute, 250 kB on average

~25000 QC Objects updated each minute, 250 kB on average

~100000 Qualities produced each minute

Performance – possible points of saturation

- Amount of data and messages being evaluated by Dispatcher
- Amount of data and messages being copied by Dispatcher
- Amount of data and messages being monitored by QC tasks
- Amount, size and kind of QC objects being merged
- Amount, size and kind of QC objects being checked
- Amount, size and kind of QC objects stored in the repository

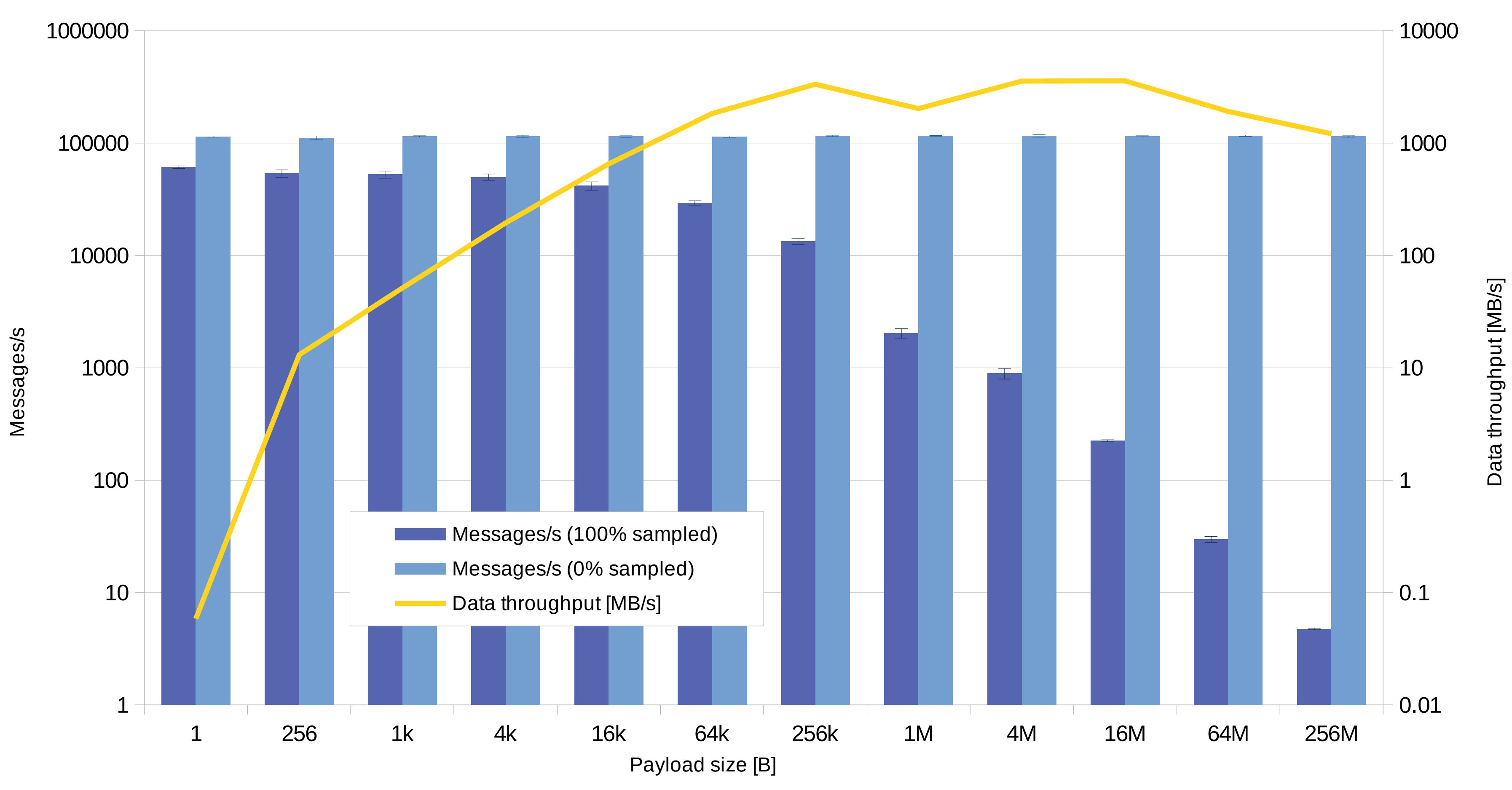
Performance – possible points of saturation

- Amount of data and messages being evaluated by Dispatcher
- Amount of data and messages being copied by Dispatcher
- Amount of data and messages being monitored by QC tasks
- Amount, size and kind of QC objects being merged
- Amount, size and kind of QC objects being checked
- Amount, size and kind of QC objects stored in the repository
- ...of course, one element can inhibit others when run on the same machine

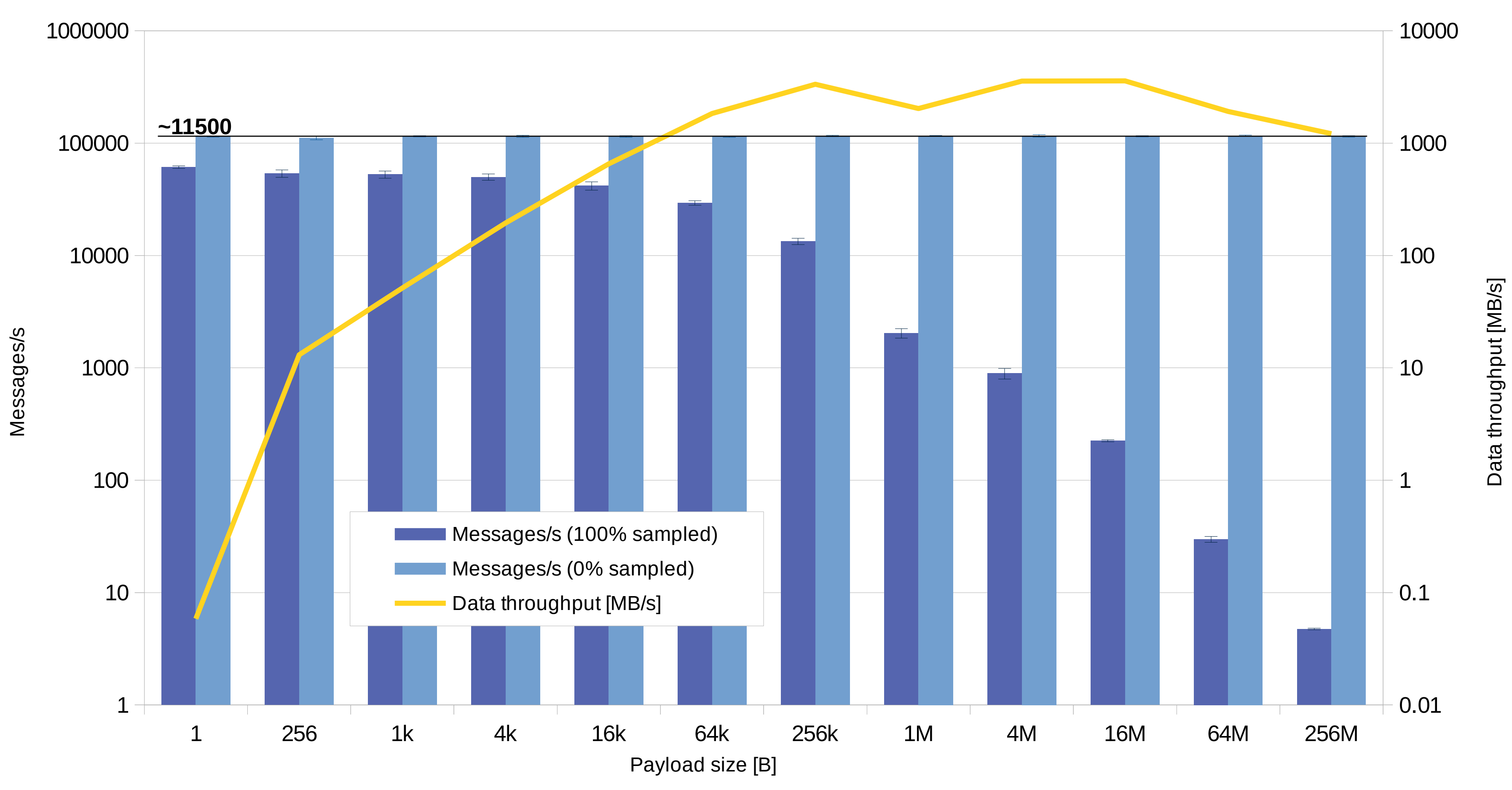
Performance – possible points of saturation

- Amount of data and messages being evaluated by Dispatcher
- Amount of data and messages being copied by Dispatcher
- Amount of data and messages being monitored by QC tasks
- Amount, size and kind of QC objects being merged
- Amount, size and kind of QC objects being checked
- Amount, size and kind of QC objects stored in the repository
- ...of course, one element can inhibit others when run on the same machine
- …we won't have time to cover all of them in this presentation

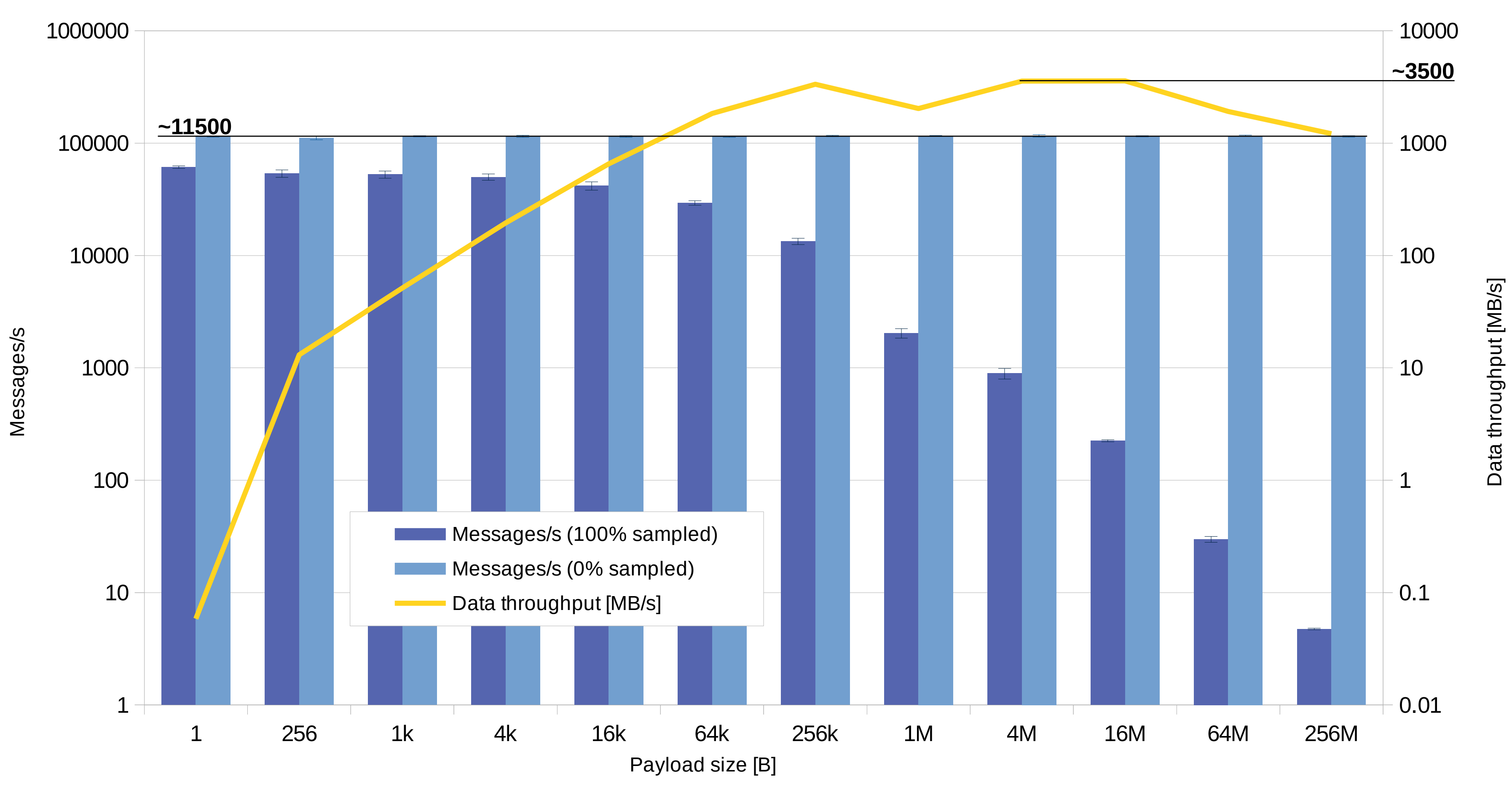
Dispatcher performance



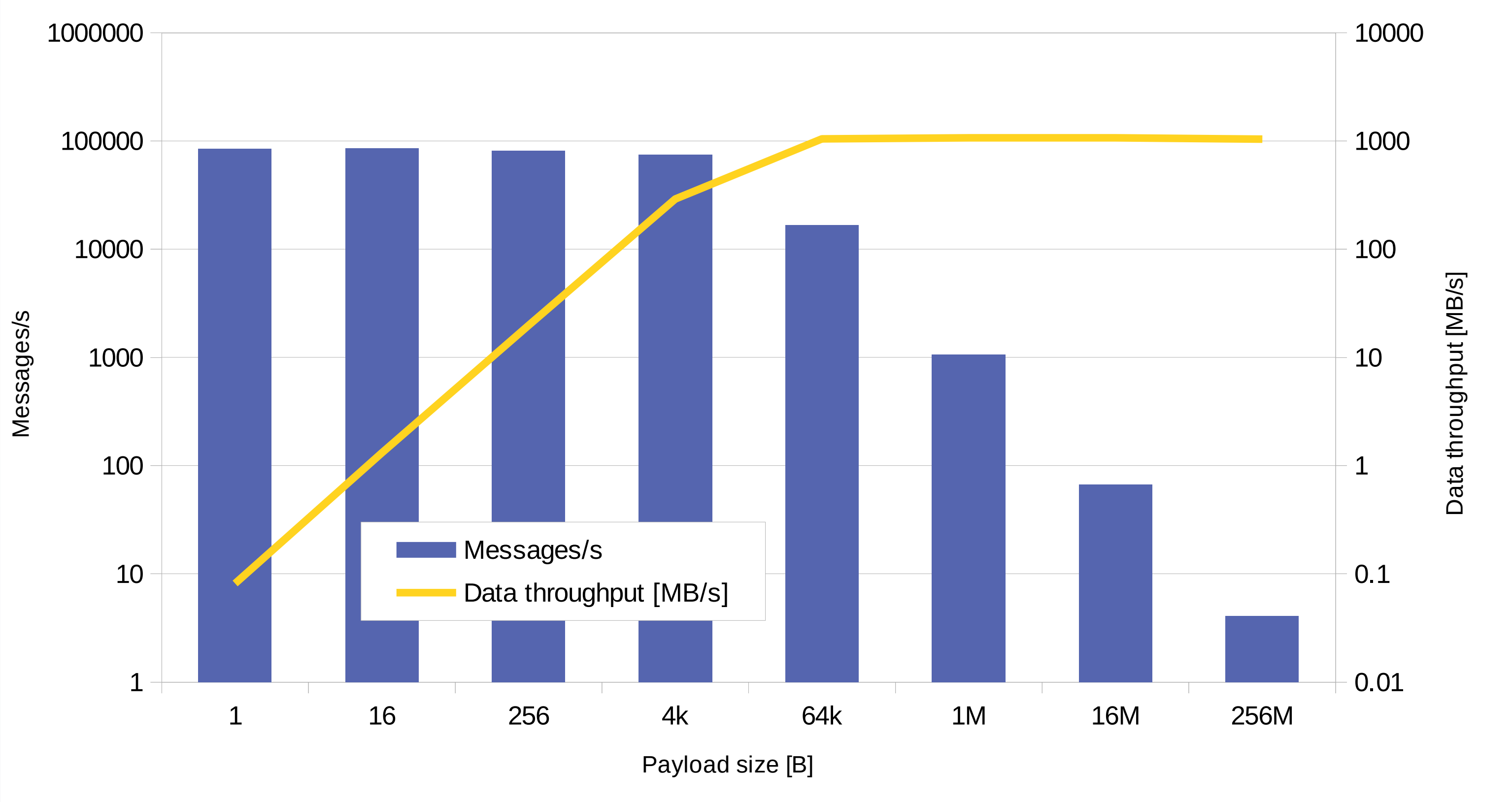
Dispatcher performance



Dispatcher performance



QC task performance – data intake (8 producers)



Other performance results

- Assuming QC Object size of 1 MB:

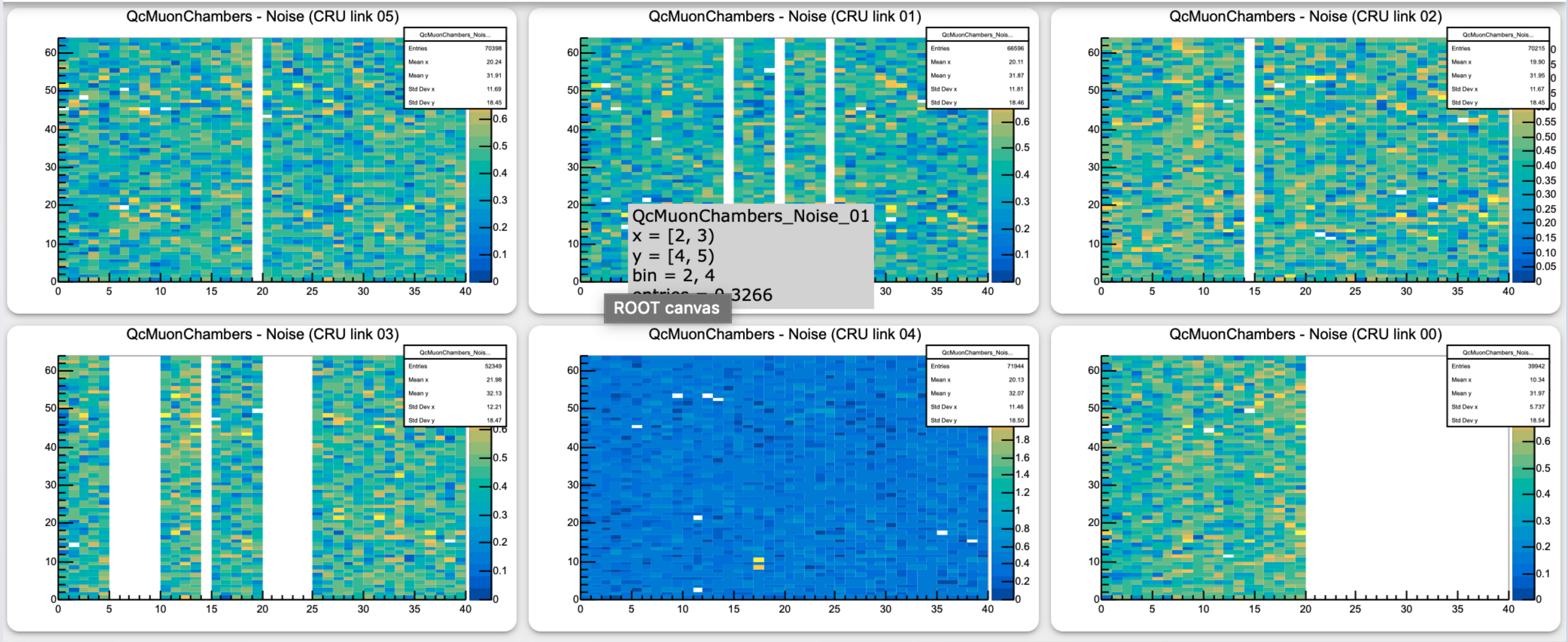
Component	Performance	Comment
QC Task	1000 objs/s published	
Merger	400 objs/s merged	1D histograms (TH1F)
Checker	2000 checks/s ran	4 checks per object
Database	400 objs/s stored	20 tasks x 20 objs/s

- The results are good and meet our needs

Status of the framework

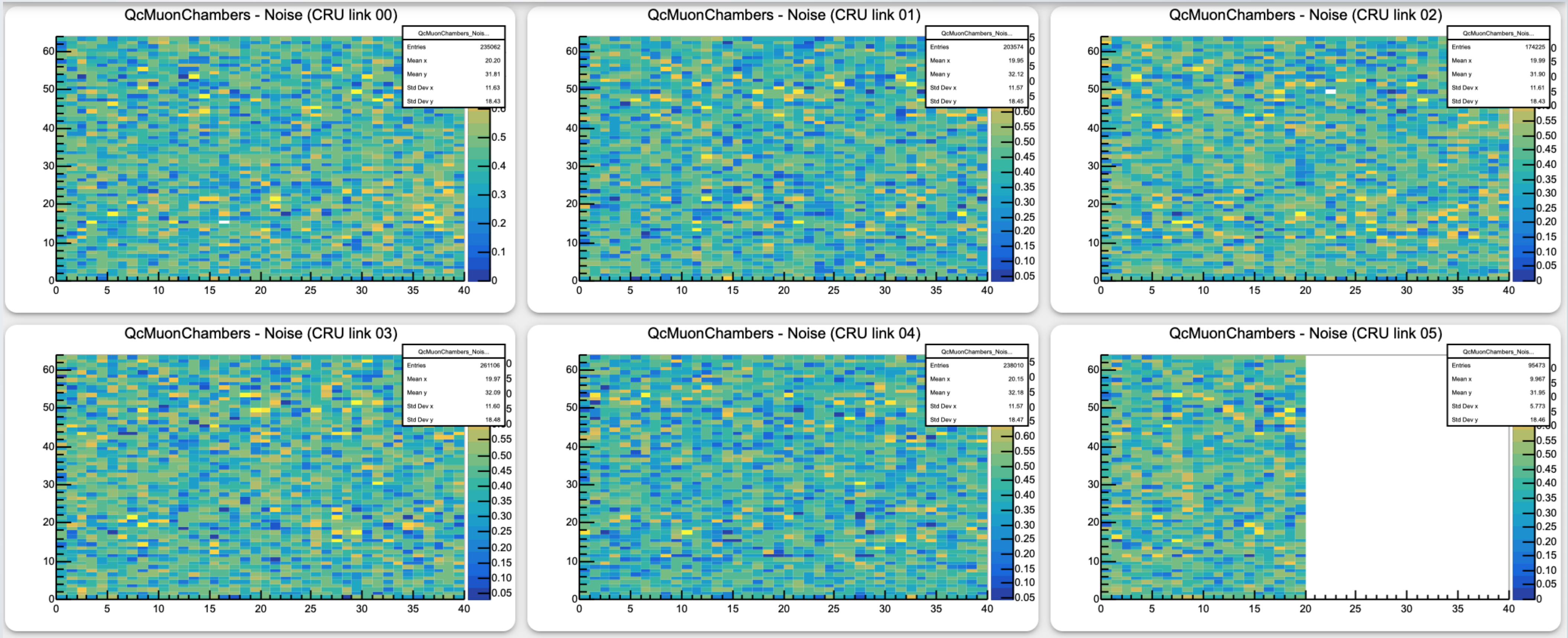
- Standalone machine setups already working
- Mergers to be benchmarked on a large scale, we will use the results to find the weak spots and choose the best merging strategies
- Correlation and Trending convenience classes currently under development
- QC detector teams have started development of their libraries, they are already used for commissioning

Quality Control of the Muon Chambers



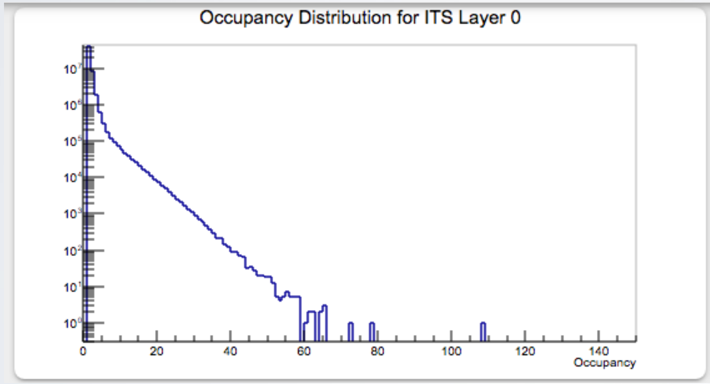
Andrea Ferrero

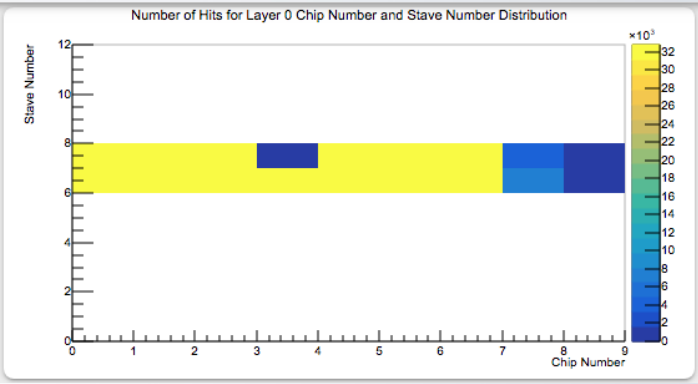
Quality Control of the Muon Chambers

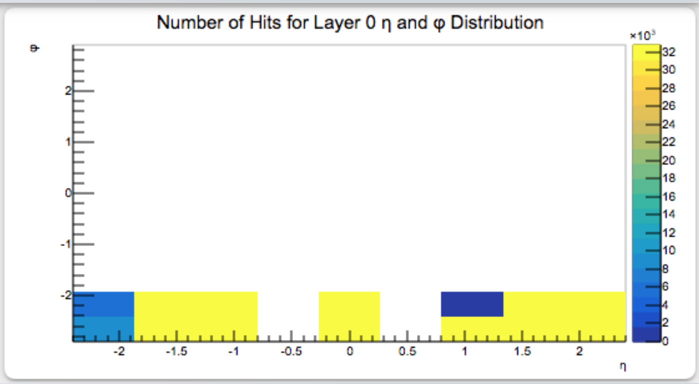


Andrea Ferrero

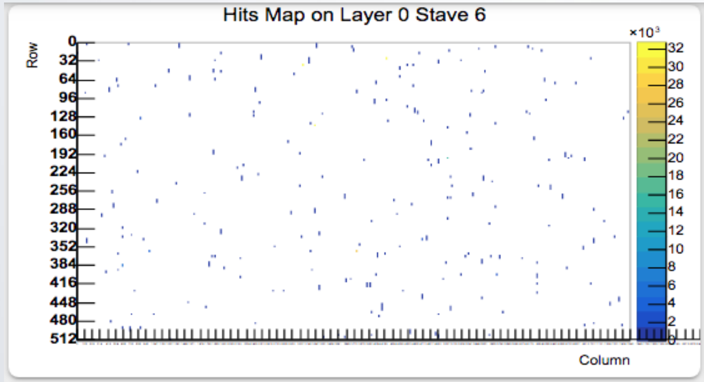
Quality Control of the Inner Tracking System

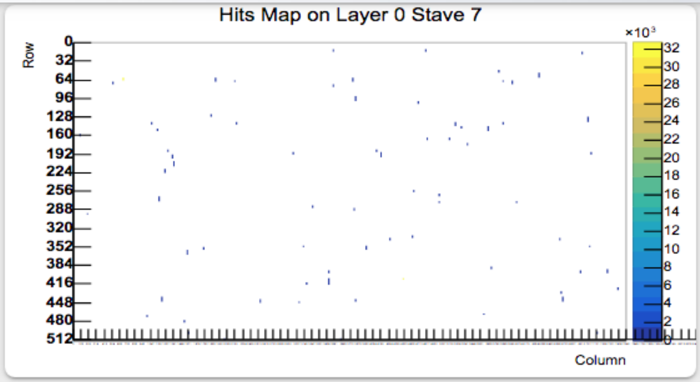






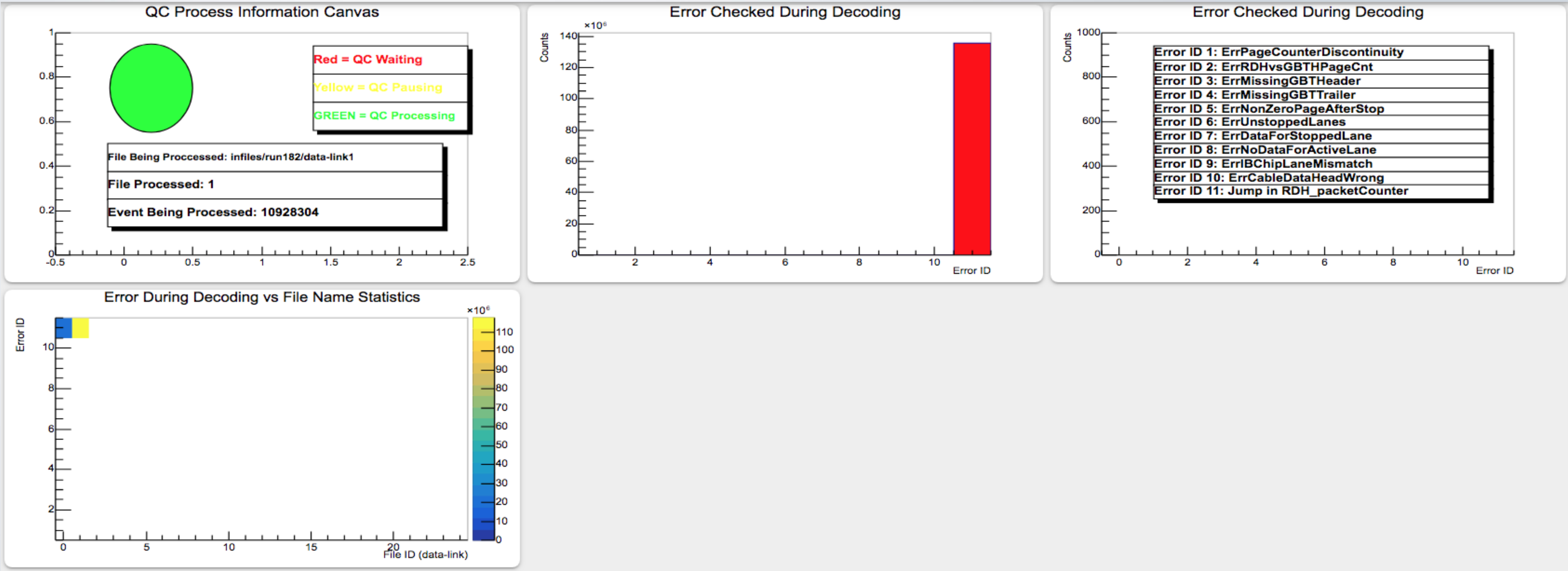
Zhaozhong Shi, Markus Keil





Zhaozhong Shi, Markus Keil

Quality Control of the Inner Tracking System



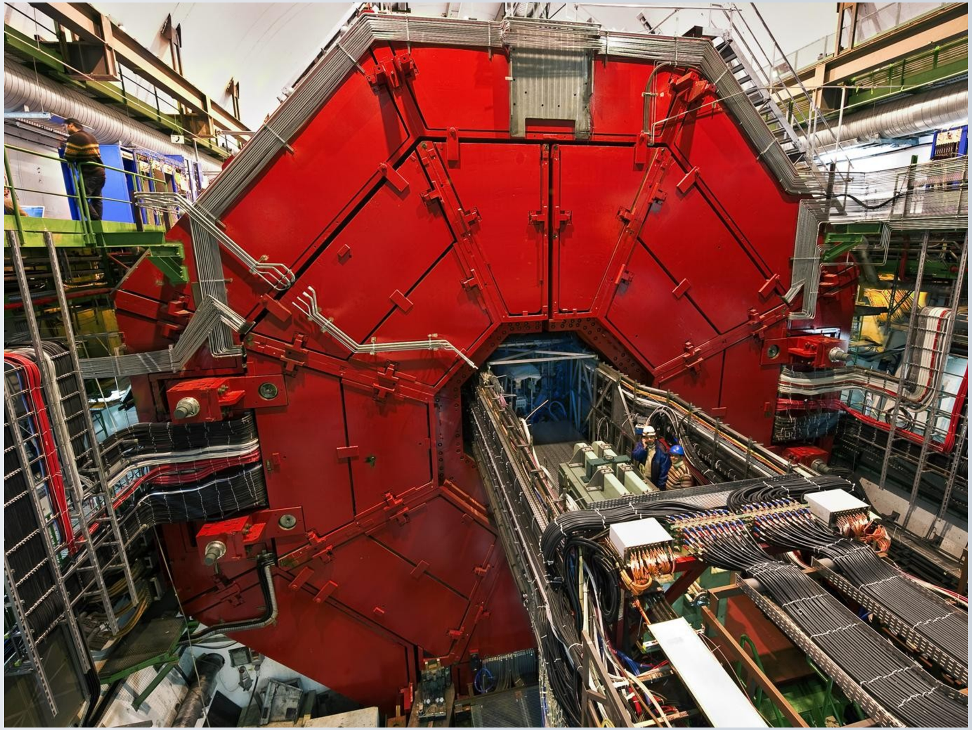
Zhaozhong Shi, Markus Keil

Future plans

- A lot of feedback expected during the detector commissioning phase
- Investigating different merging strategies
- Correlation and Trending convenience classes
- Large scale benchmarks
- Fine–tuning the complete setup

Summary

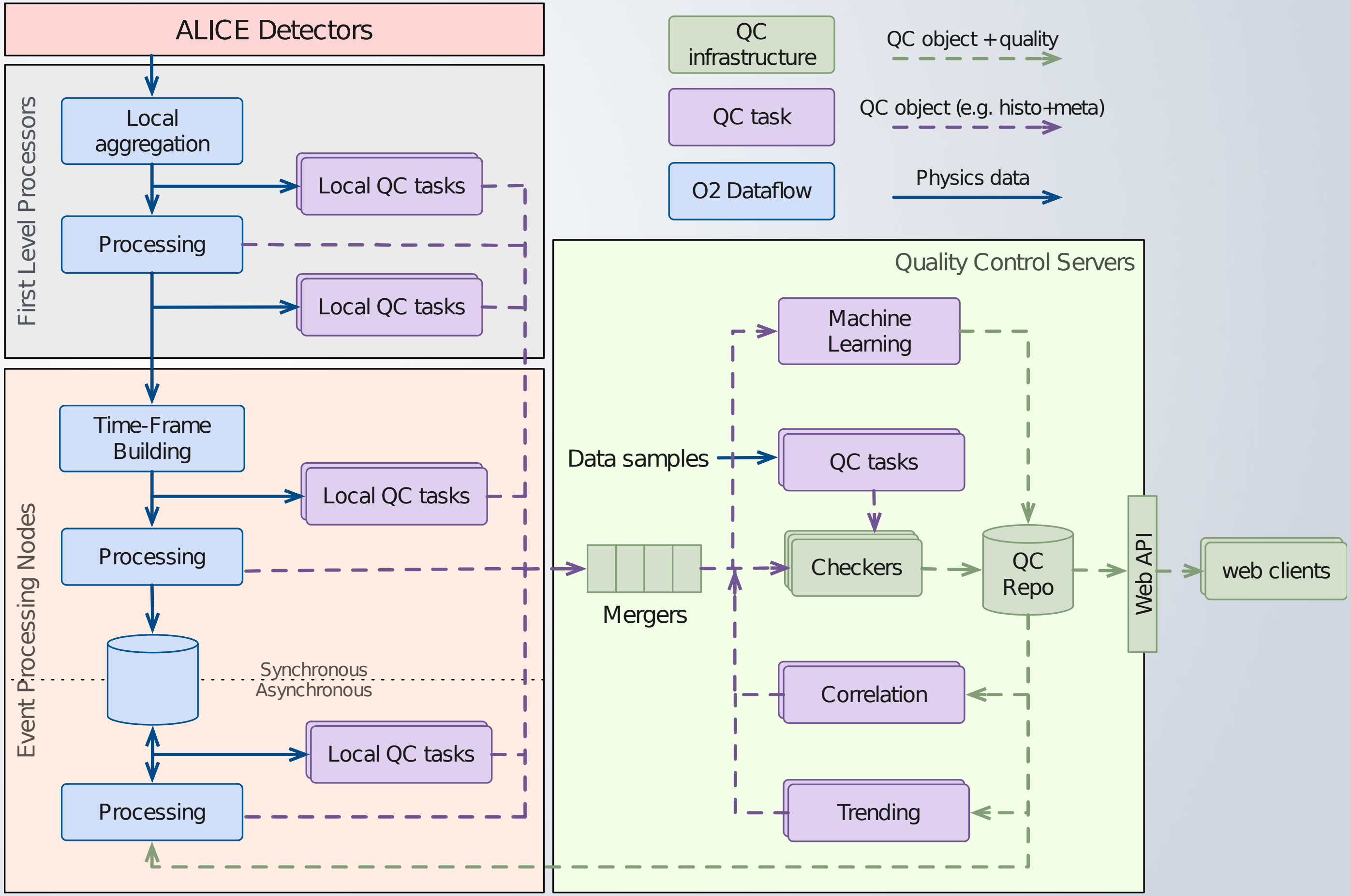
- A new Quality Control framework for the ALICE experiment
- It combines the former online Data Quality Monitoring and offline Quality Assurance
- The overhead of QC framework is small – the biggest factor will be user algorithms and data transport
- Currently used by the detector teams



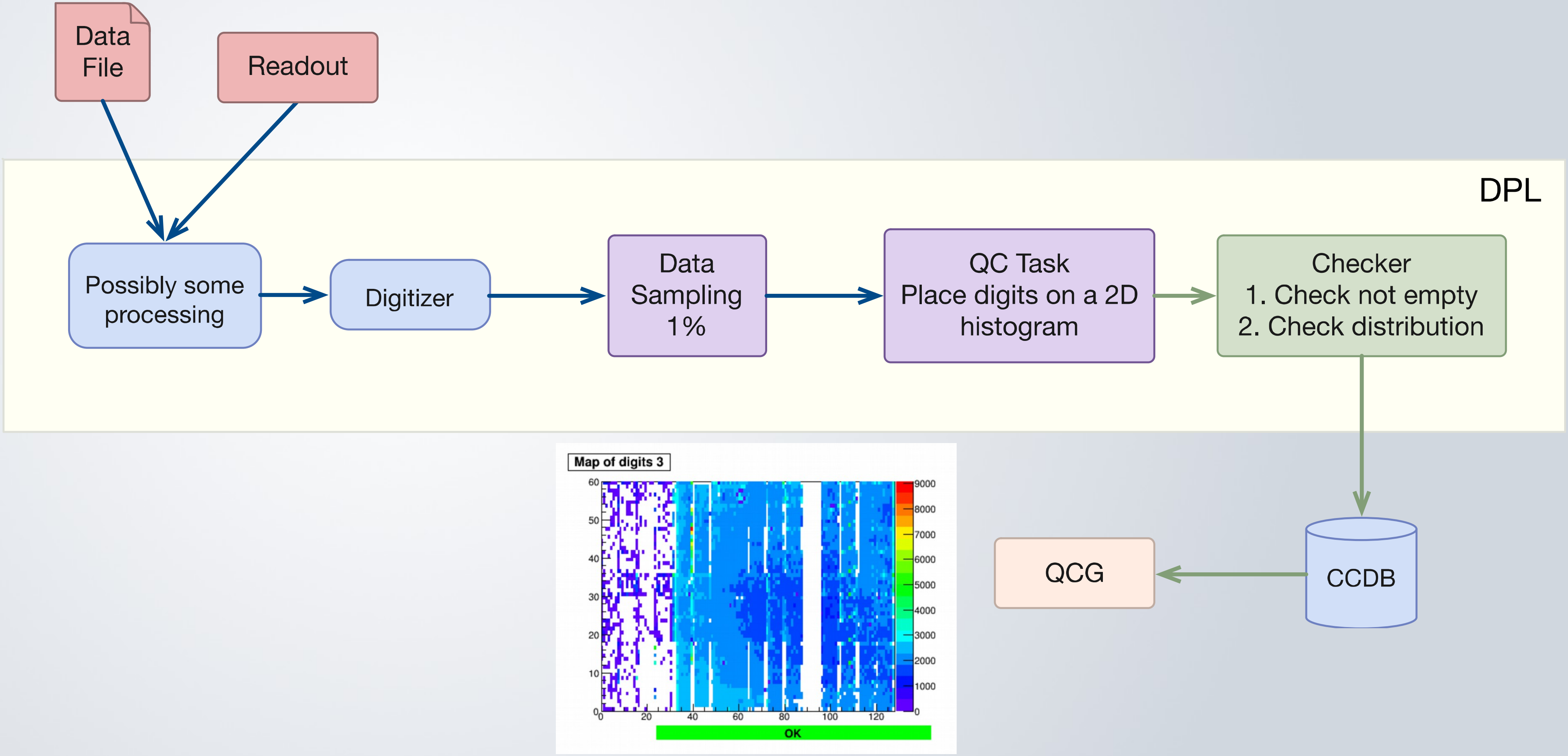
Thank you

Backup slides

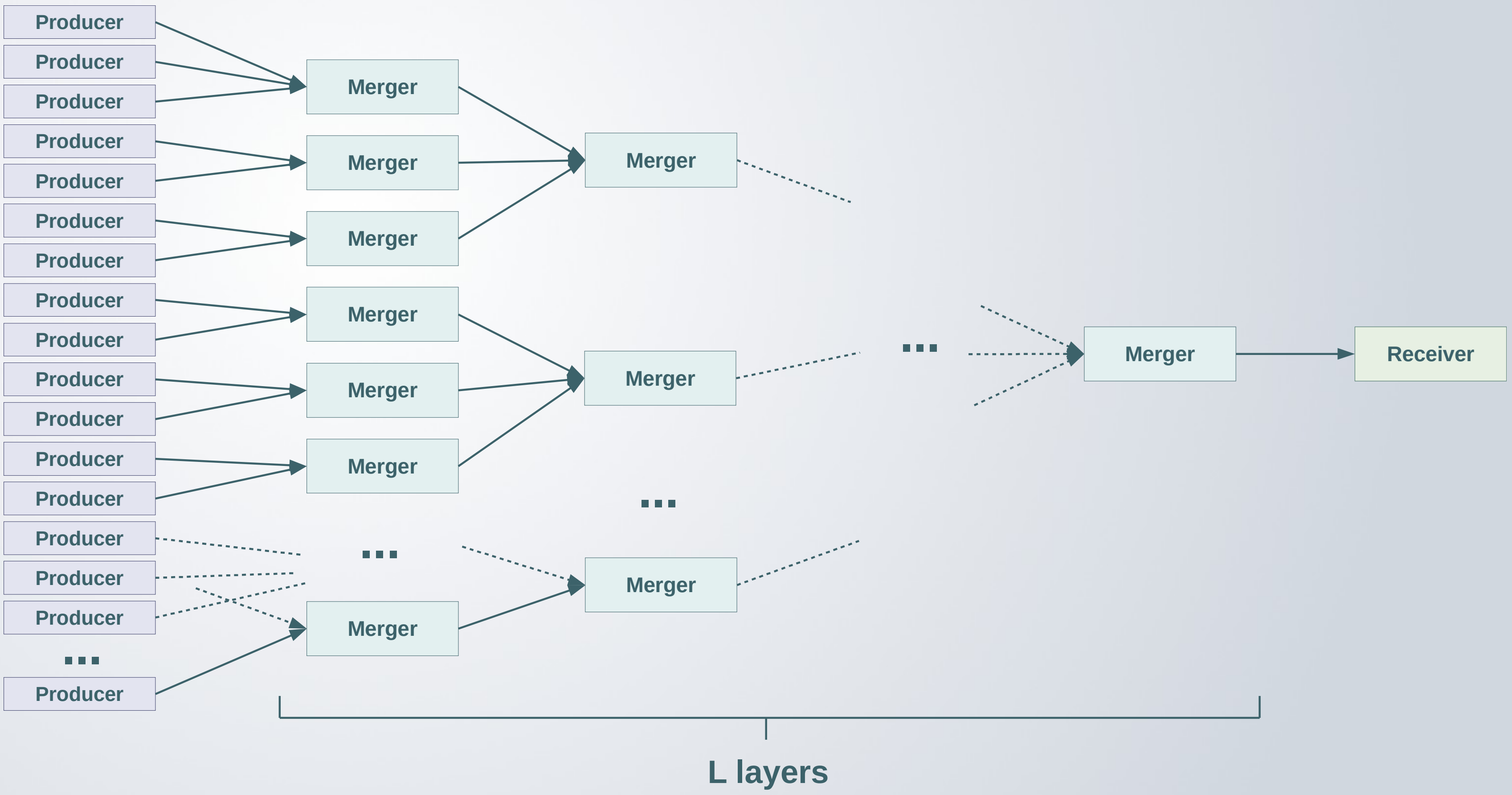
Quality Control – global view



Example of a QC chain



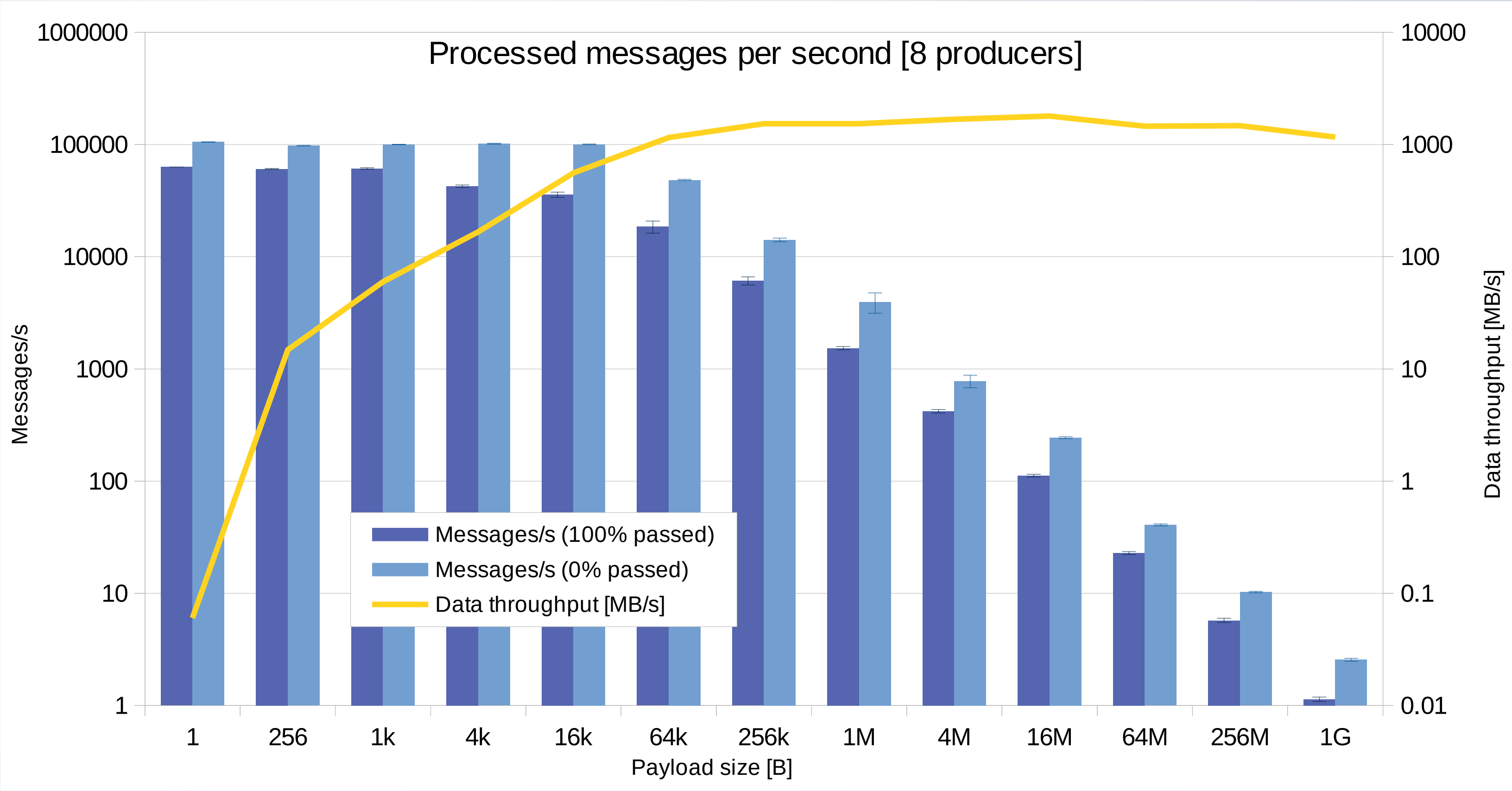
Mergers - topologies



QC objects repository

- Based on ALICE's Conditions and Calibration DataBase (CCDB)
- Can be installed locally
- A shared online instance for development

Dispatcher performance without shared memory

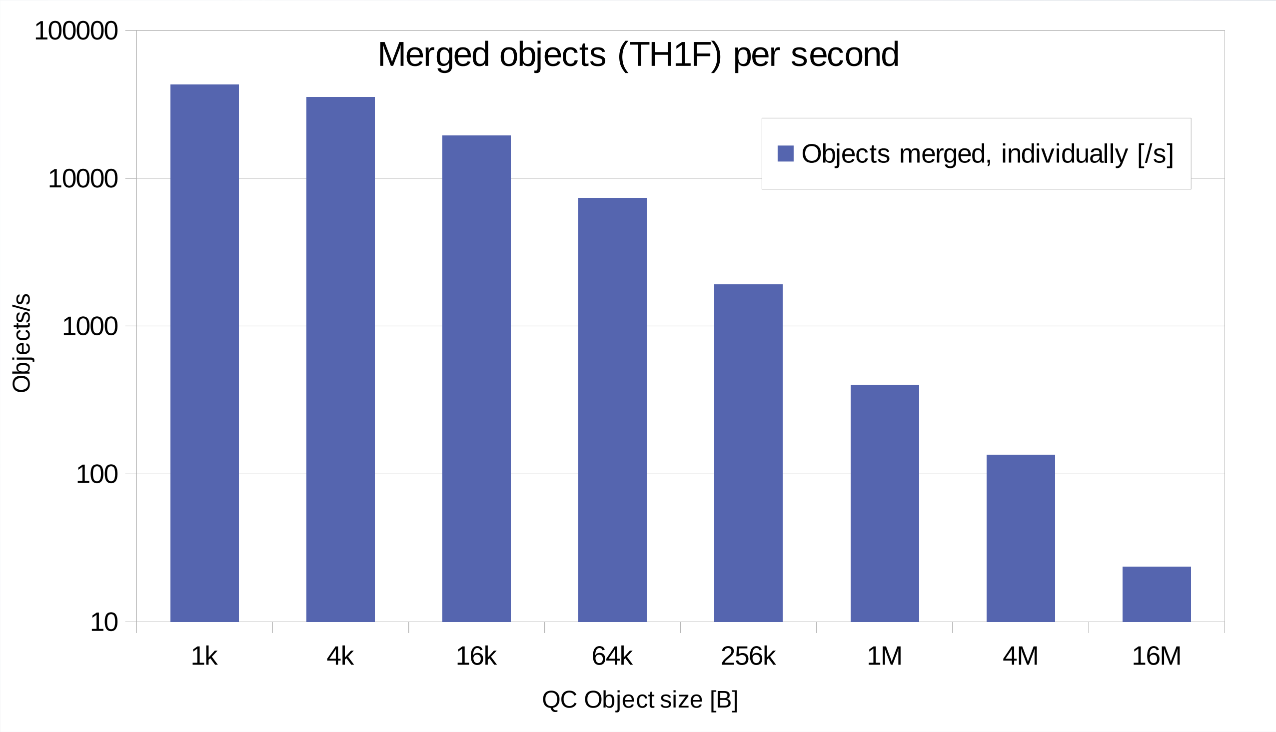


QC Task performance – data publication

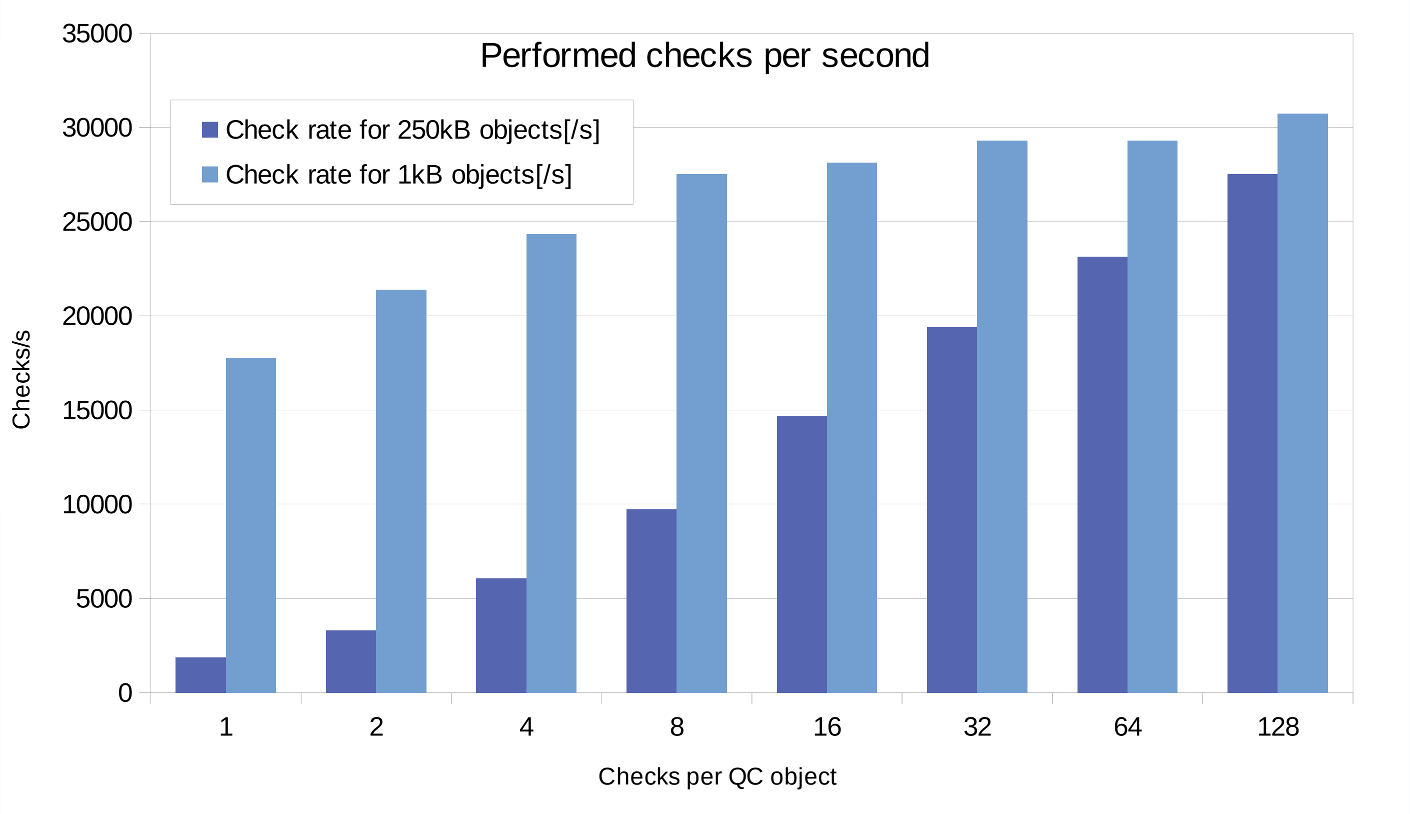
- We plan to publish an updated QC object once per minute
- For the test, we do it once per second
- Achieved rates for 2D histograms:

Object size (in RAM)	1kB	10kB	100kB	1MB	10MB
Number of objects [/s]	>100000	>30000	>8100	>1000	>70

Mergers performance



Checkers performance



Database performance

- QC target rate : 25000 obj/m –> 400 obj/s
  - 10 kB, 100 kB, 500 kB, 1 MB, 2.5 MB, 5 MB objects
  - 20 tasks
  - 20 objects published per seconds per task
- Results
  - 400 obj/s achieved up to 1 MB per object
  - 100 obj/s for 2.5 MB
- The objects are planned to be much smaller than 1 MB in most cases

Barthélémy von Haller, https://indico.cern.ch/event/686151/contributions/3117562/attachments/1705355/2747629/wp7-ccdb-O2TB.pdf