Quality Control for the Muon Forward Tracker

Tomáš Herman

Czech Technical University in Prague
Faculty of Nuclear Sciences and Physical Engineering

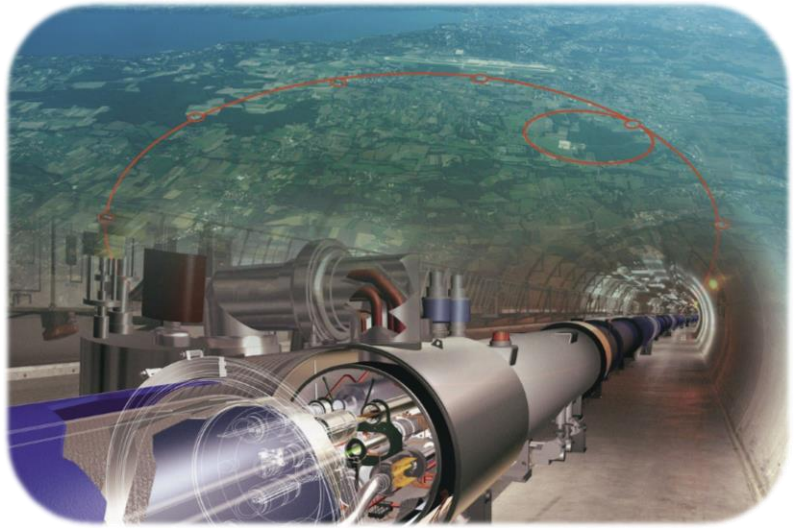
Třetí miniworkshop difrakce a ultraperiferálních srážek
15.09.2020
Děčín

Why we need QC

Why we need QC

- Price range of CERN is in the order of **10^{11} CZK**

  → Data we measure is very valuable



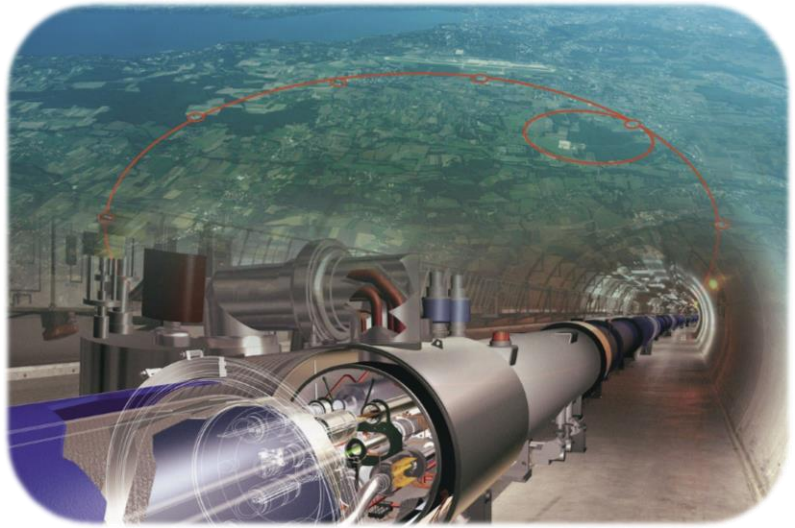
Why we need QC

- Price range of CERN is in the order of **10^{11} CZK**

  → Data we measure is very valuable

- We need to make sure our **detector is working properly**
  - Dead part of the detector
  - Fake signals
  - ….

- QC plays a key role in making sure our data is fine for physics analysis



What QC does

What QC does

- Types
  - **Fast** identification of problems

What QC does

- Types
  - **Fast** identification of problems

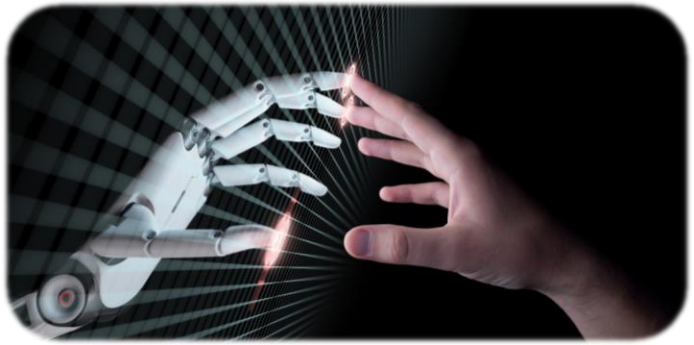
  - **Long-term** behaviour of the detector

What QC does

- Types
  - **Fast** identification of problems

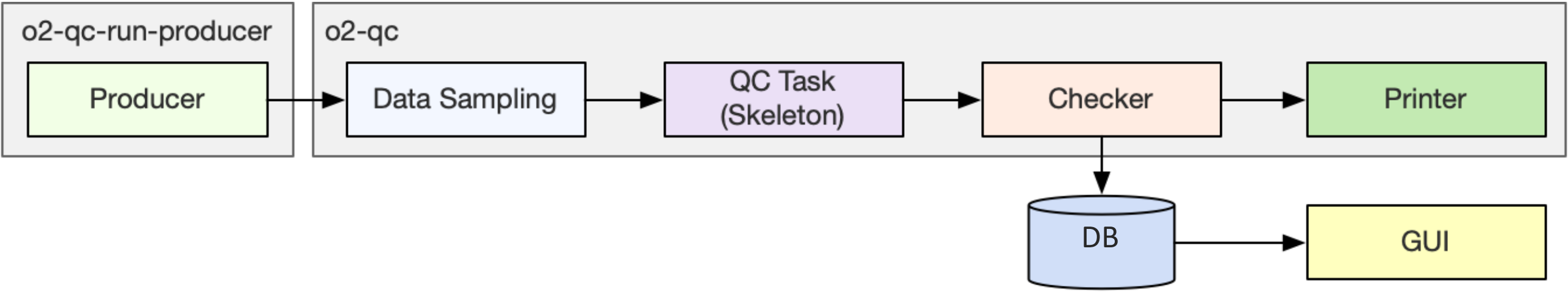
  - **Long-term** behaviour of the detector

- Monitoring
  - **Automatic checks**
    - Programmed and periodically applied on the measured data

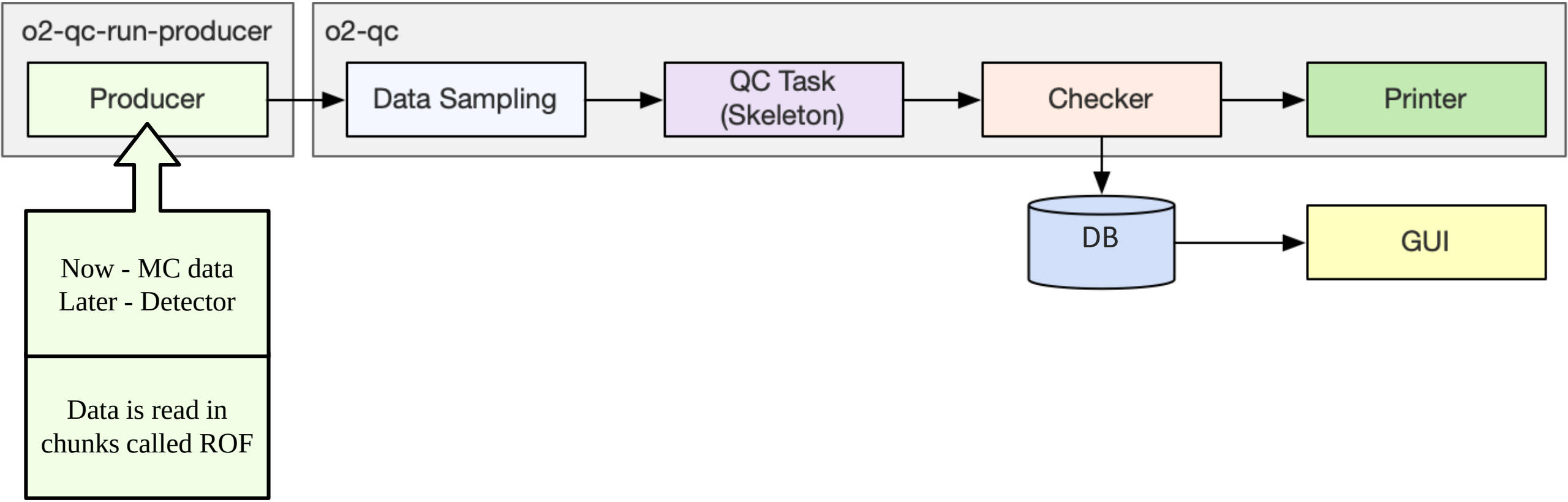


  - **Human checks**
    - Shifter in the control room looking at messages from the automatic checks
    - Identifying any other potential problems

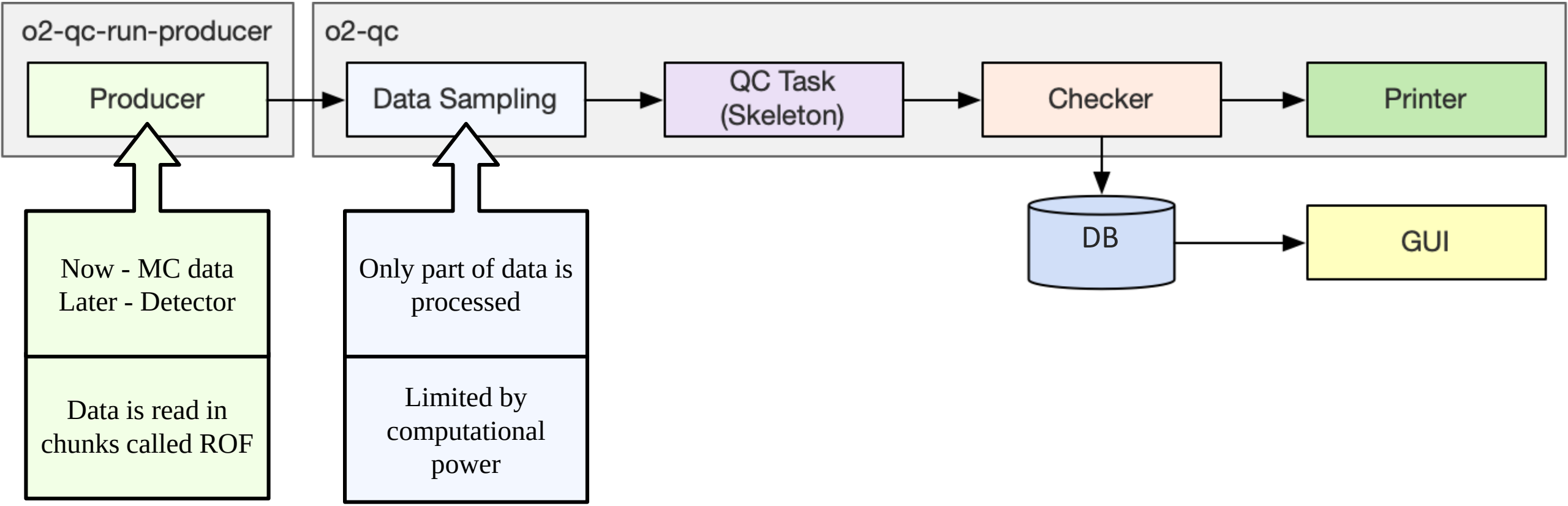
How does QC work



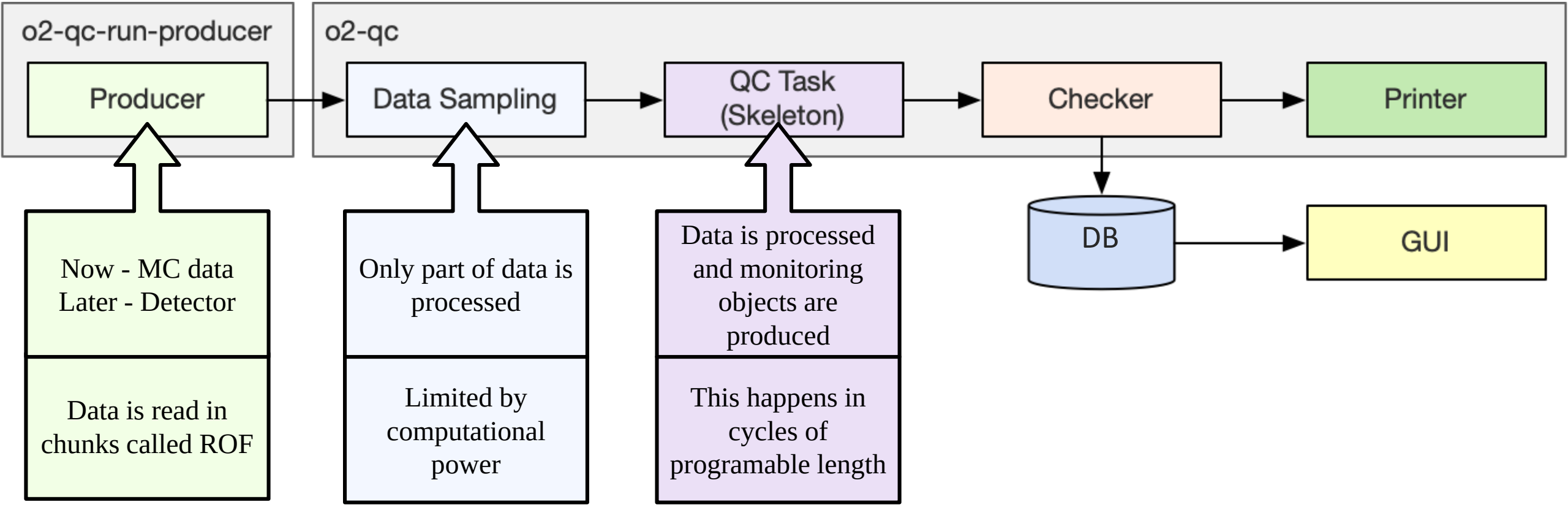
How does QC work



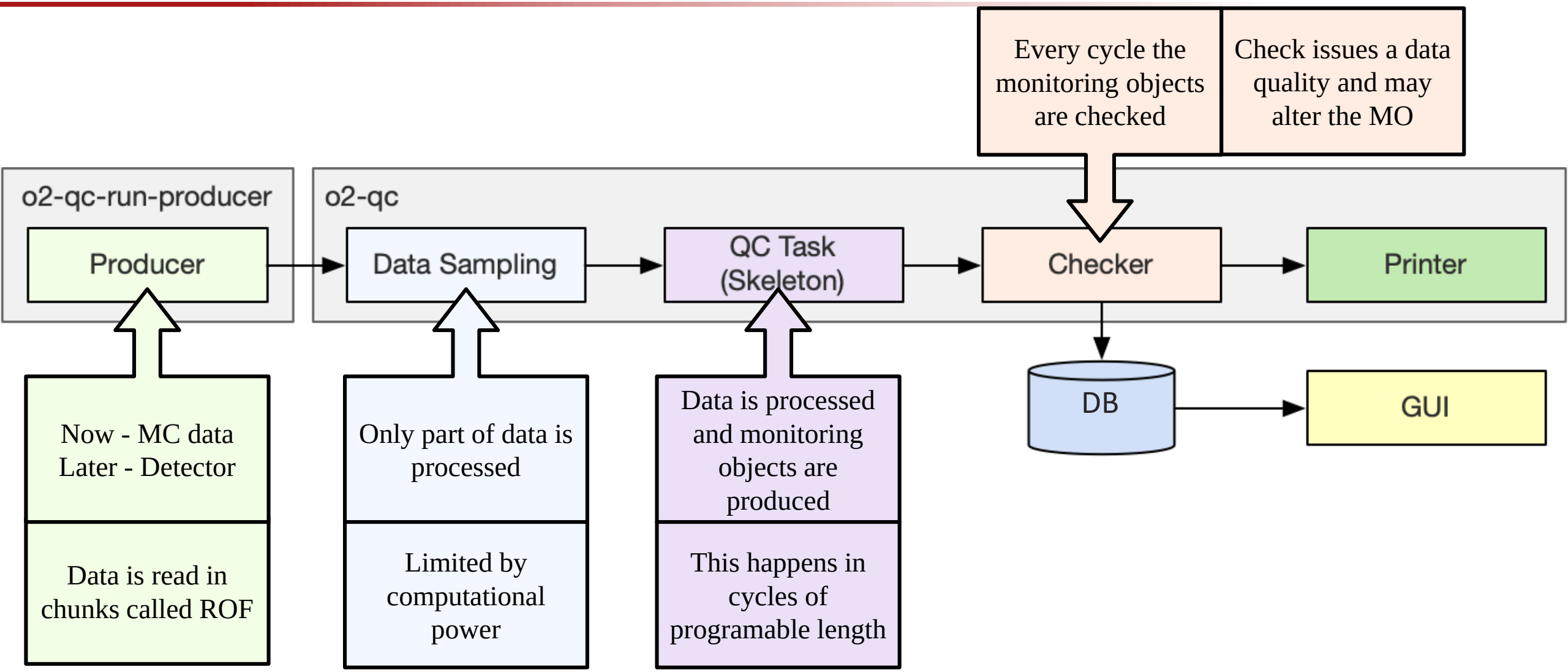
How does QC work



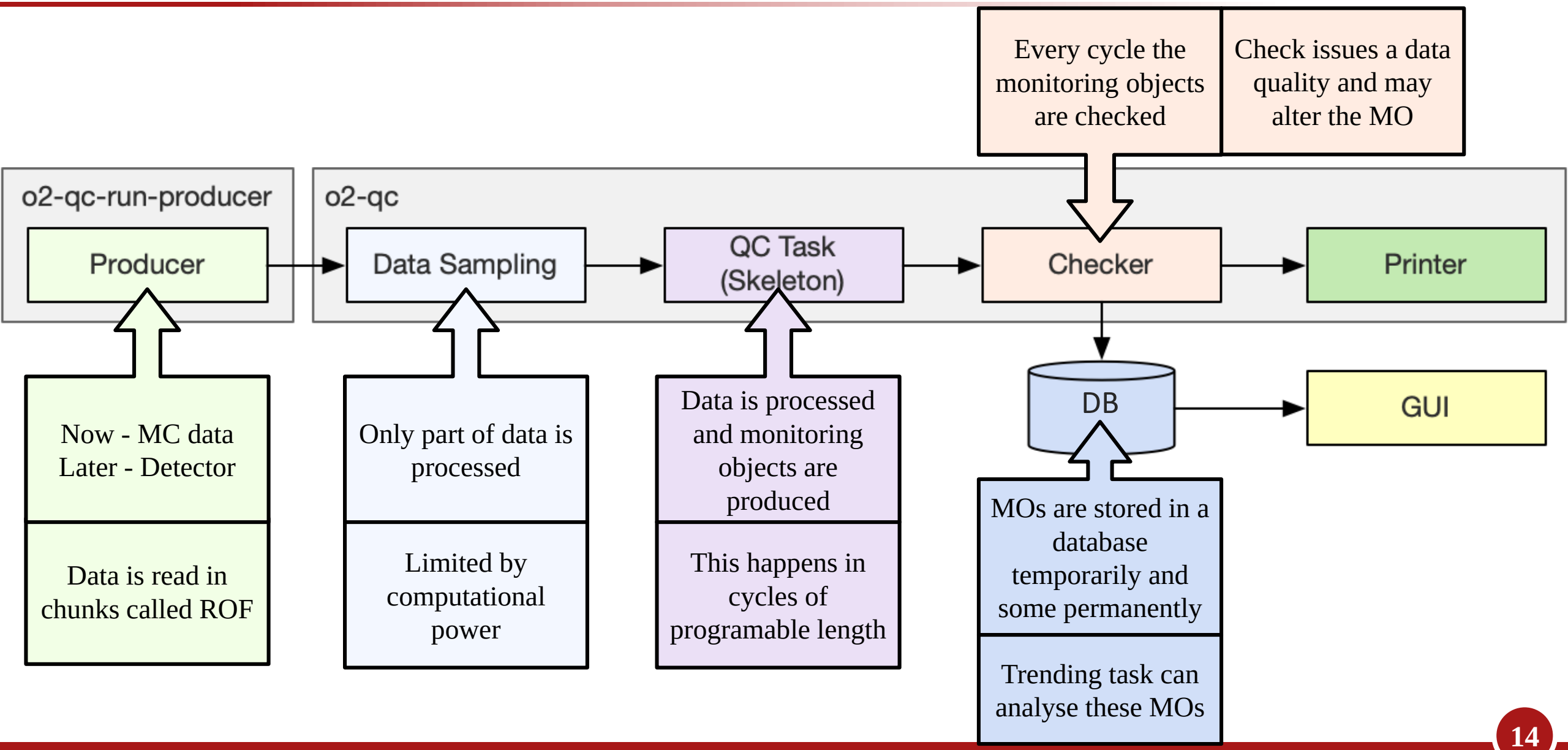
How does QC work



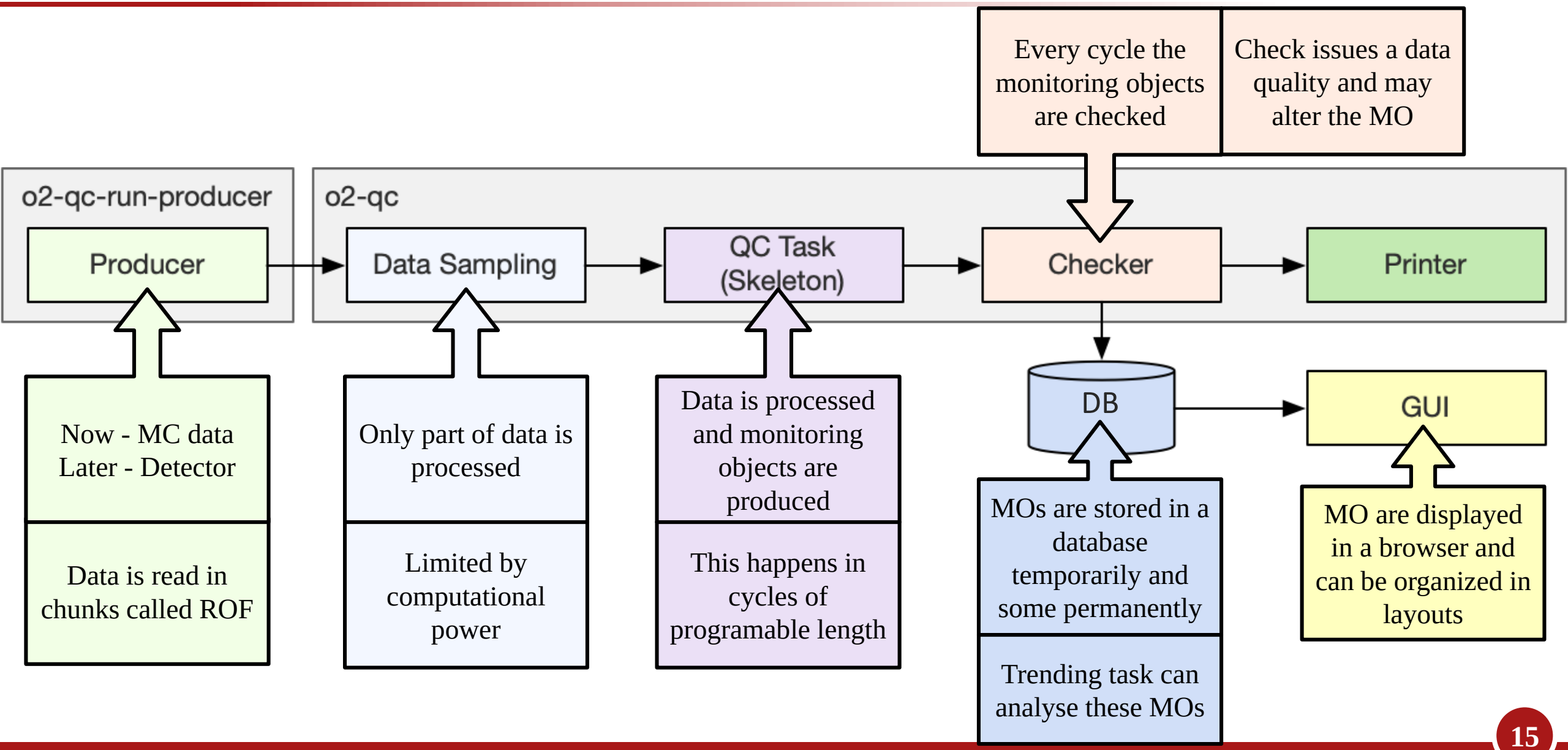
How does QC work



How does QC work



How does QC work



Time scales for the QC

Collisions	≈500 kHz in pp and ≈50 kHz in PbPb
ROF/STF/CTF	≈4 µs/≈20 ms/≈20 ms
Sampling	Fraction of data going into QC - programmable for each task
Cycle	Checkign and publishing of MOs - programmable expectation 1 min
Permanent storage in DB	MOs go to the DB every cycle - later most of them are deleted Current expectation is to keep one each hour or at the end of each run
Runs	Several minutes to hours
Fills	Several runs
Periods	Several days/weeks

QC Tasks and Monitoring Objects

- Readout
  - To be defined

QC Tasks and Monitoring Objects

- Readout
  - To be defined

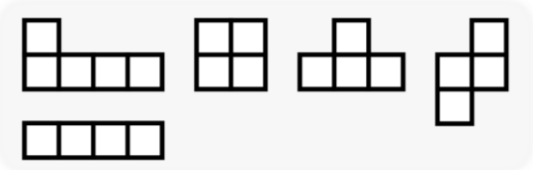
- Digits (individual pixel hits)
  - 2D hit maps for each chip
  - Detector layout with chip occupancy

QC Tasks and Monitoring Objects

- Readout
  - To be defined

- Digits (individual pixel hits)
  - 2D hit maps for each chip
  - Detector layout with chip occupancy

- Clusters
  - Only test histograms

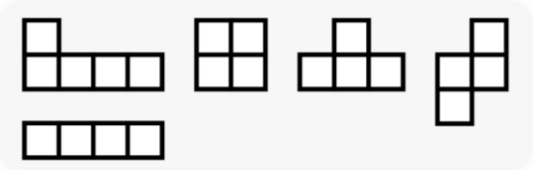


QC Tasks and Monitoring Objects

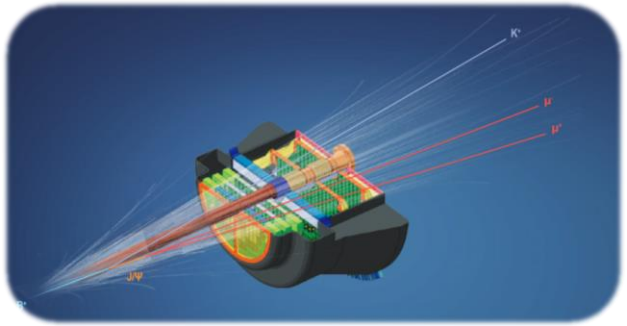
- Readout
  - To be defined

- Digits (individual pixel hits)
  - 2D hit maps for each chip
  - Detector layout with chip occupancy

- Clusters
  - Only test histograms



- MFT Tracks
  - ϕ angle distribution for +/- tracks

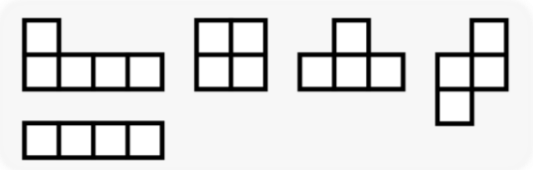


QC Tasks and Monitoring Objects

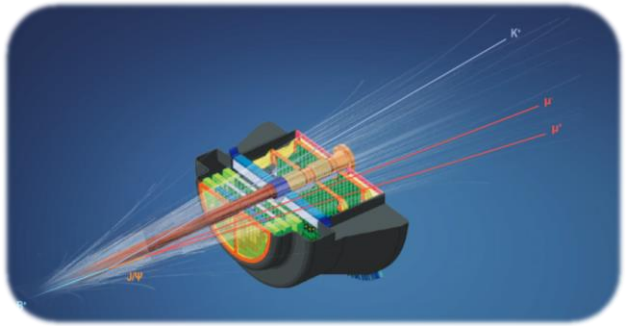
- Readout
  - To be defined

- Digits (individual pixel hits)
  - 2D hit maps for each chip
  - Detector layout with chip occupancy

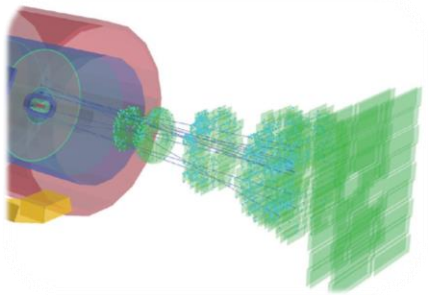
- Clusters
  - Only test histograms



- MFT Tracks
  - ϕ angle distribution for +/- tracks



- MFT+MUON Tracks
  - To be discussed



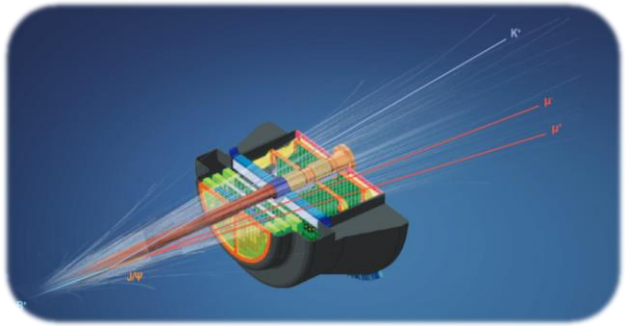
QC Tasks and Monitoring Objects

- Readout
  - To be defined

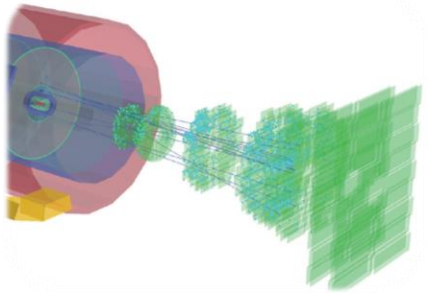
- Digits (individual pixel hits)
  - 2D hit maps for each chip
  - Detector layout with chip occupancy

- Clusters
  - Only test histograms

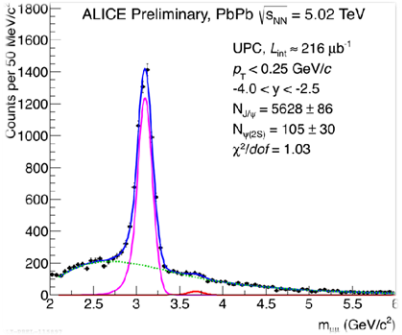
- MFT Tracks
  - ϕ angle distribution for +/- tracks



- MFT+MUON Tracks
  - To be discussed



- Physics
  - Under discussion for UPC



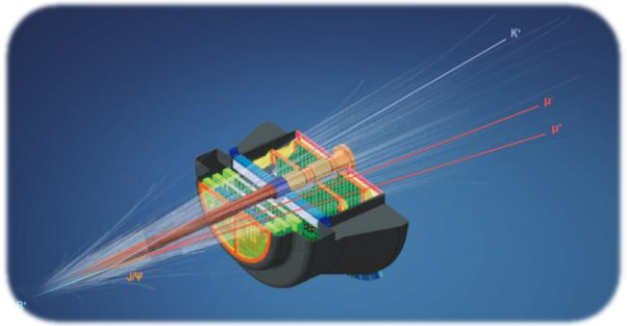
QC Tasks and Monitoring Objects

- Readout
  - To be defined

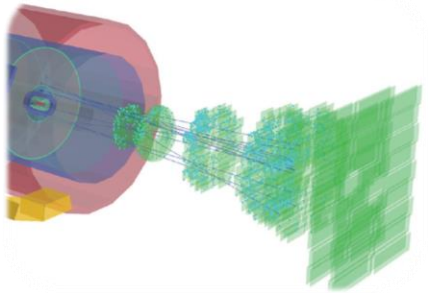
- Digits (individual pixel hits)
  - 2D hit maps for each chip
  - Detector layout with chip occupancy

- Clusters
  - Only test histograms

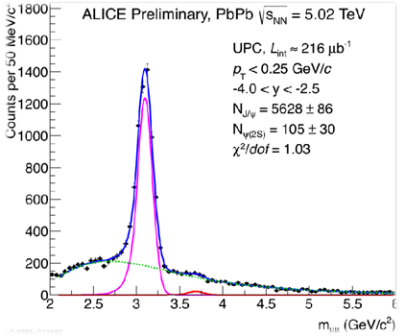
- MFT Tracks
  - ϕ angle distribution for +/- tracks



- MFT+MUON Tracks
  - To be discussed



- Physics
  - Under discussion for UPC



- Calibration
  - Due to low amount of data QC is probably not the optimal place for it

Objects to be monitored in the Digit QC Task

Currently the basic QC object

2D hit maps 512 x 1024

Layout with all disk half faces with chips as individual bins

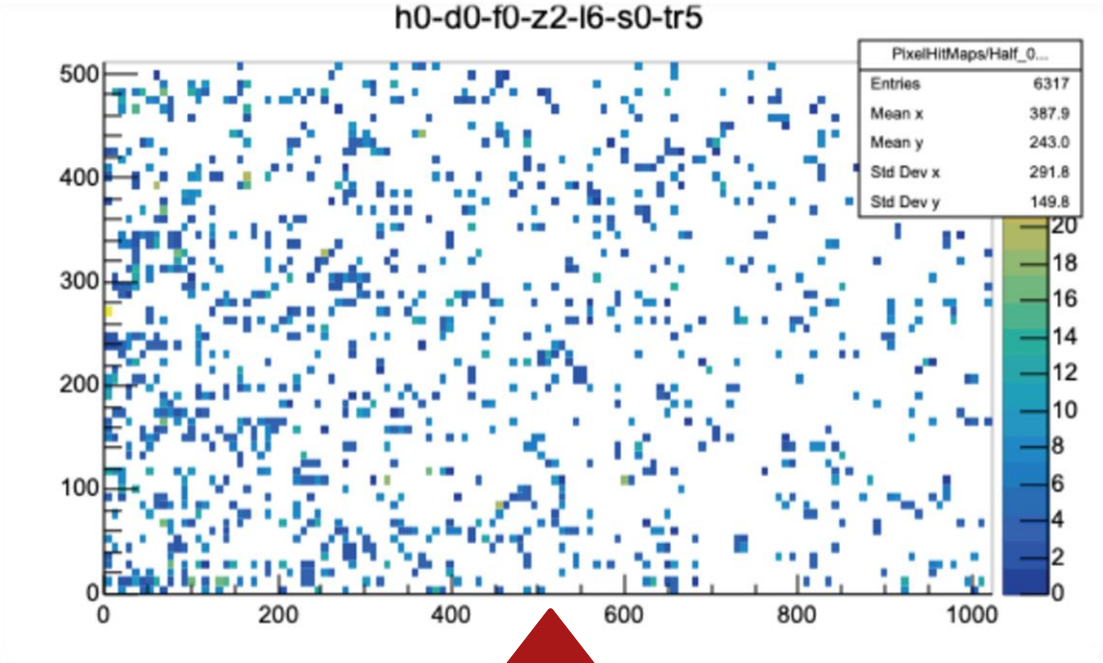
Pixels	512 x 1 024 x 936 = 490 733 568
Double columns	512 x 936 = 479 233
Regions	32 x 936 = 29 952
Chips	936
Ladders	280
Zones	80
Half-disk faces	20
Disks	5

- Lot of objects to monitor
- Lot of them overlap

Need for efficient data handling without duplicate information

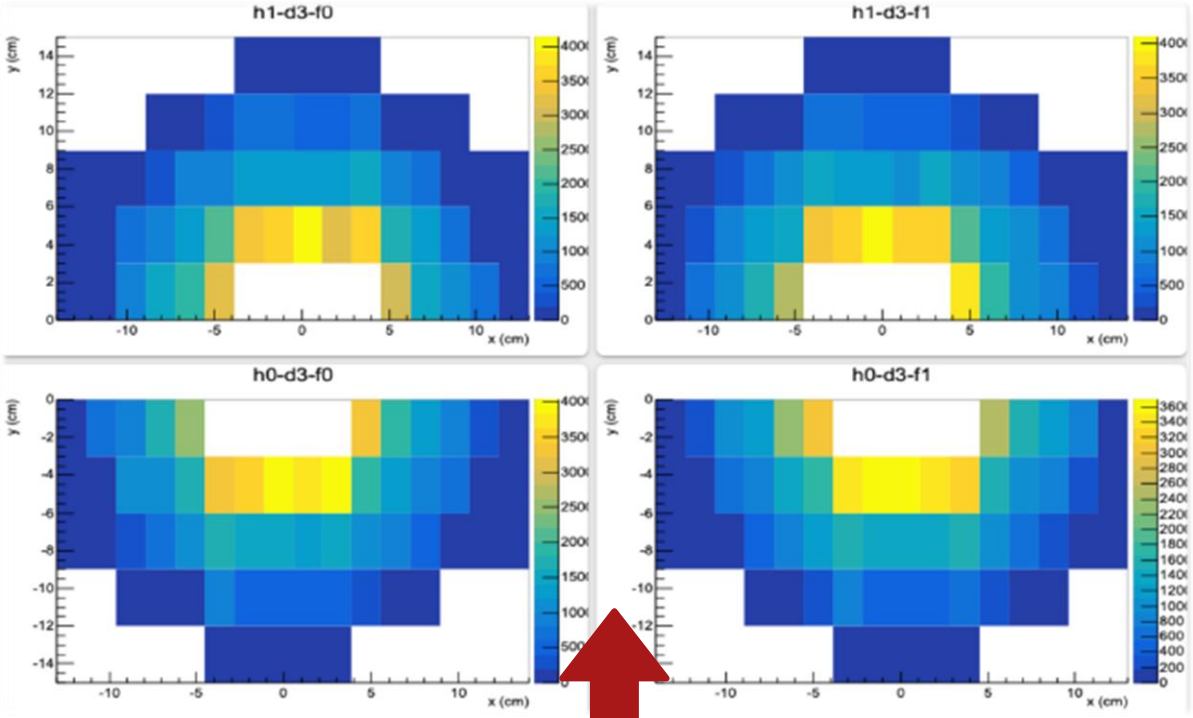
Digit QC Task

- 2D pixel hit maps



Access to pixels, double columns and regions

- Layout with chip occupancy



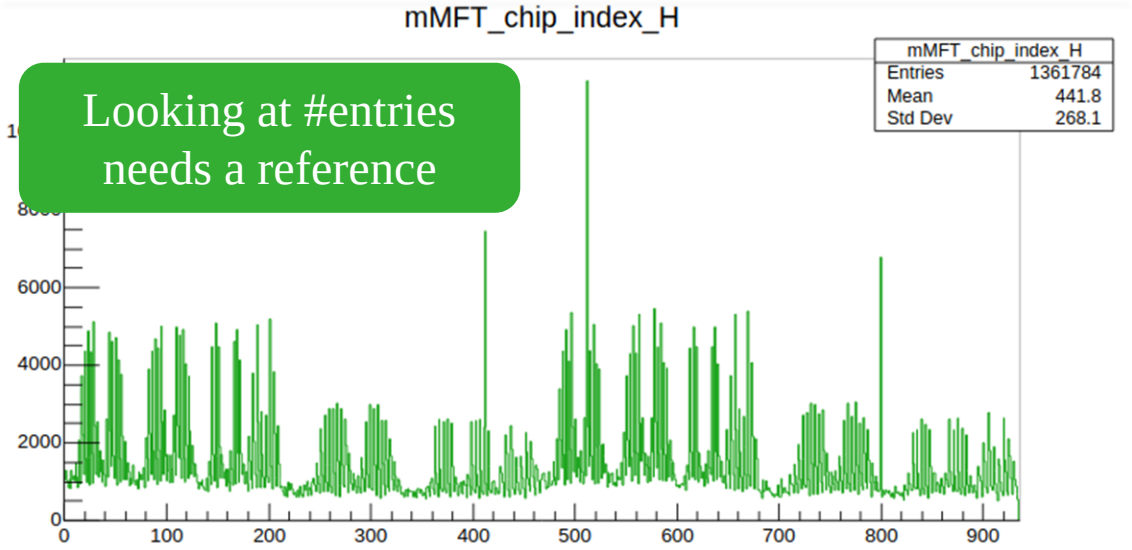
Access to chips, ladders and zones

Finding (very) hot pixels - Test

1. Generate Pythia8 events
2. Every ROF a digit with a specific pixel is added in the data producer (in our test: 413, 513 and 800)
3. Check number of entries and standard deviation of pixel hit maps

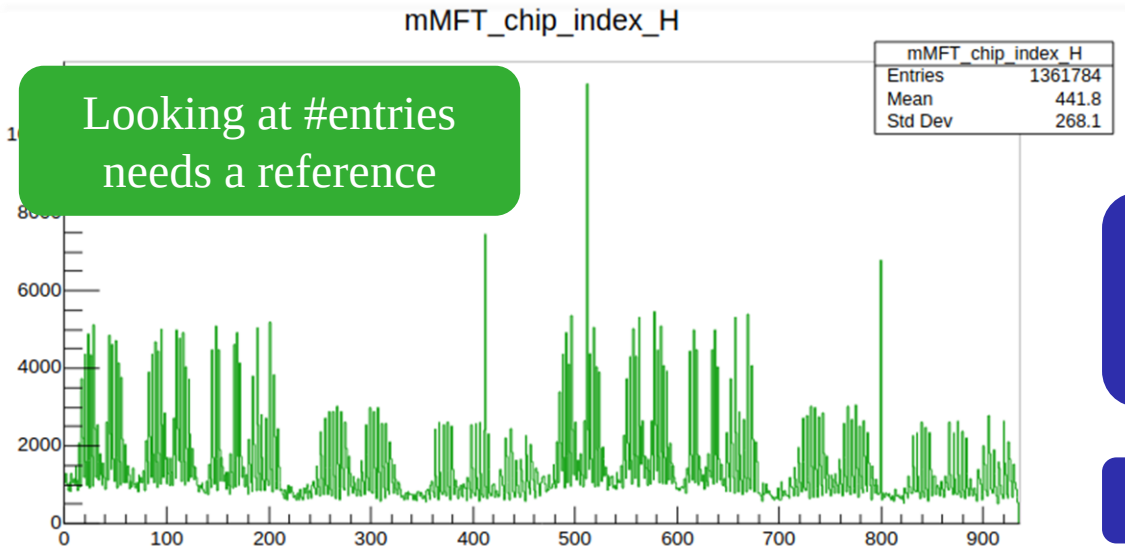
Finding (very) hot pixels - Test

1. Generate Pythia8 events
2. Every ROF a digit with a specific pixel is added in the data producer (in our test: 413, 513 and 800)
3. Check number of entries and standard deviation of pixel hit maps

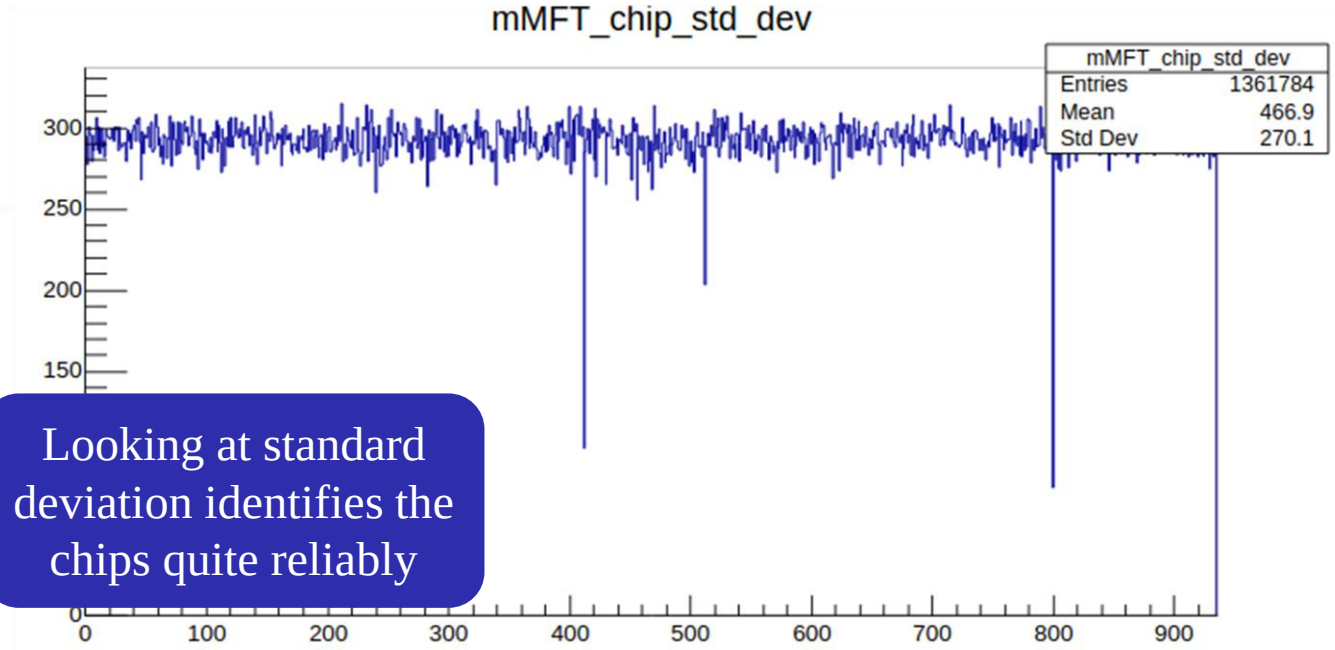


Finding (very) hot pixels - Test

1. Generate Pythia8 events
2. Every ROF a digit with a specific pixel is added in the data producer (in our test: 413, 513 and 800)
3. Check number of entries and standard deviation of pixel hit maps



Looking at #entries needs a reference



Looking at standard deviation identifies the chips quite reliably

Specific pixel has to be identified in the pixel hit map

Next steps

- Finding not so hot pixels (firing every ROF is an extreme case)
- Exploring the effect of several hot pixels in one chip
- Finding dead pixels (if possible)
- Add more MOs to Cluster and Tracks tasks
- Develop trending, physics, calibration tasks
- Develop warning scheme/layout for shifters
- Start using real data
- Start using the real hardware which will run the tasks (FLP/EPN)
- Implement communication with the DCS database
- …

Thank you for your attention