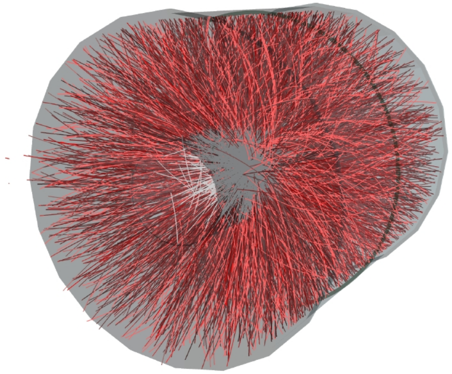


Run II (Online) Reconstruction and Calibration Strategy for the TPC

Mikolaj Krzewicki
Jochen Thäder

Research Division and ExtreMe Matter Institute EMMI, GSI Helmholtzzentrum für Schwerionenforschung

2013-11-06

Baseline strategy – online

- **Applications of the online calibration for Run2 for *PHYSICS* runs within the HLT**
  - Therefore, also online reconstruction and extensive online monitoring is needed
- **Pedestal runs, etc. will be done via DAQ Das**
  - They require only RAW data and are not needed immediately in on-line reconstruction
- **No new functionality foreseen for the DAQ DAs**
- **In case the output of DAQ DAs of *PHYSICS* runs, is needed during the reconstruction → The DAs need to be ported into the HLT**
  - Otherwise DAs can stay in the DAQ world
- **Online calibration should be able to handle PaC and RoE**
  - Our current strategy foresees this, with minimal implications
- **HLT TPC cluster compression will be used**

Baseline strategy – offline

- **The output from the online reconstruction is not intended for immediate use in physics analysis in Run2 → similar to current CPASS0**

- **Online created calibration objects will be used in offline reconstruction**

- **Online reconstruction output will be be used as input to offline reconstruction**
  - Usage of compressed clusters and/or tracks (pre-seeding)

- **An TPC offline reconstruction process has to have the following components:**
  - Possibility to join offline calibration pass (**CPASSx**) until we have understood all implications for Run II, and or in order to refine the calibration (plus QA)
  - Validation pass (**VPASS**) (plus QA)
  - Physics production pass (**PPASS**) → AOD filtering (plus QA)

Online calibration baseline

- **Online reconstruction is needed to obtain online calibration**
- **Online calibration is needed for online reconstruction**

  → **Feedback Loop**

- **Extensive online data and calibration monitoring needed during data-taking**

  → Quality Control **QC**

- **Fail-over / Re-running**
  - In case of problems during online calibration or re-running of improved algorithms, the same functionality with HLT offline framework can be used to run the calibration offline
  - The current CPASS0/CPASS1 framework will be still in place as well

Remarks

First: General Remarks on the Run II Strategy

Then: The specific remarks on PaC and RoE will be afterwards

Definitions for the Data-Flow (1)

- **Ideally data-taking should be *Fill* based**

  → natural interval for beam dependent parameters
- **However, trying to enlarge a *Run* comes close to it**

- **Sub-divide *Fill/long Run* in *Calibration Intervals* (CIs) , *which are periods of***
  - Stable detector conditions
  - Stable trigger conditions
  - Luminosity intervals

Calibration is bound to a CI !!!

→ **The online calibration procedure is bound to the CIs !!!!!**

Definitions for the Data-Flow (2)

- Implications
  - Change of TPC conditions (RoE) can be internally detected within HLT calibration
  - Further changing of detector conditions could be "tagged" by ECS → signal needed
  - Changing of luminosity conditions could be "tagged" by ECS → signal needed
- However in context of offline tasks:
  - The change of the CI should be always signaled by ECS
  - HLT triggers CI change via a message to ECS

Lets discuss this at the end!

- Add a "flag" to the ESDevent
  - Stable CI number (negative value indicates non-calibrated event)
- On start of a new CI
  - Increase the CI number
  - Reset calibration and QA

Definitions for the Data-Flow (3)

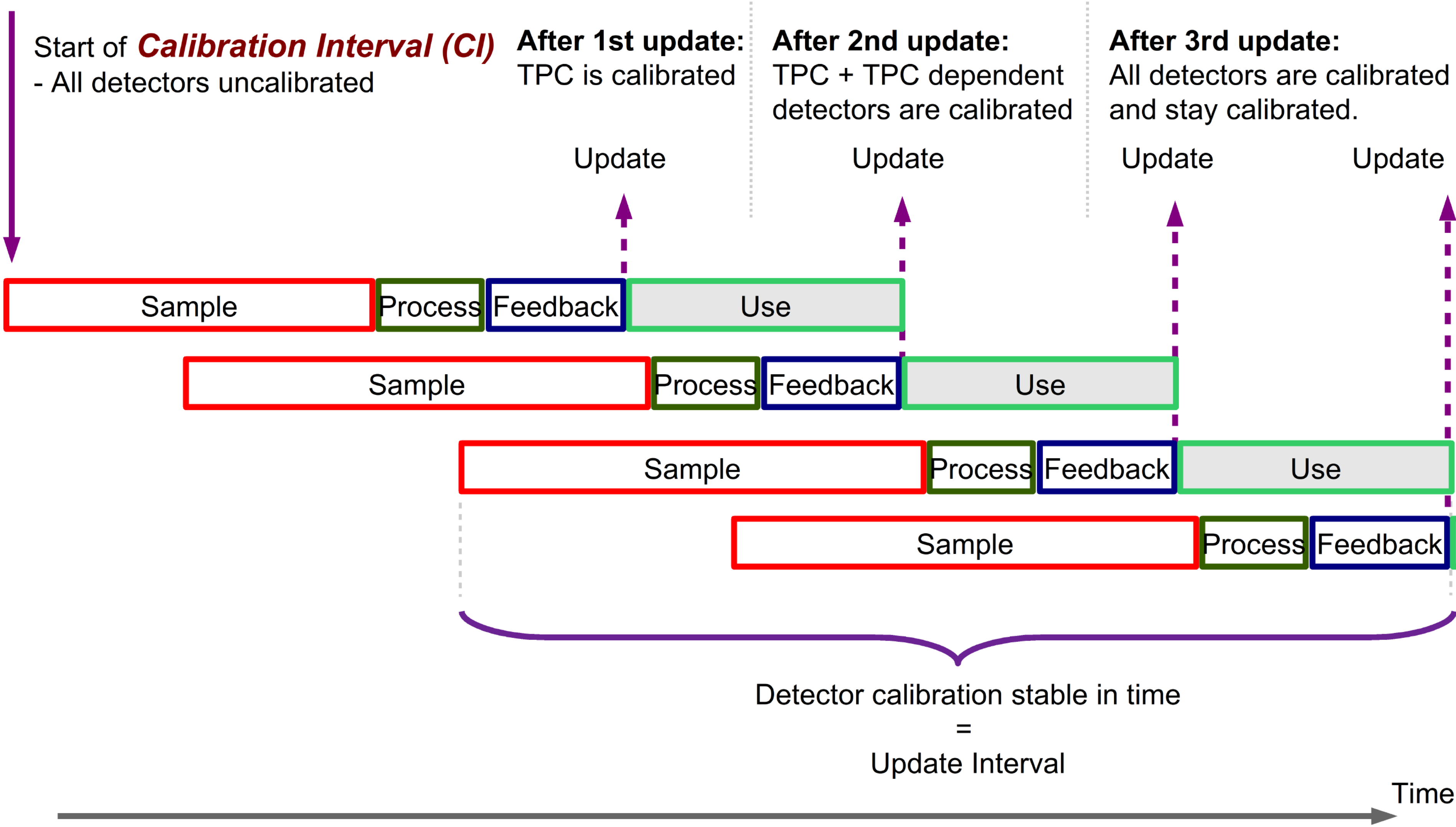
- **Follow the time-dependent calibration parameters in *Update Intervals* (UIs)**
  - Length of an *Update Interval* has to be optimized out of the following parameters
    - (Time-) stability of calibration
    - Number of tracks/clusters needed
    - Updating latency in the on-line system
    - Calibration processing and feedback latency

    } Initial guess **1-2 min**
  - eg.: O(5.000) events needed for TPC

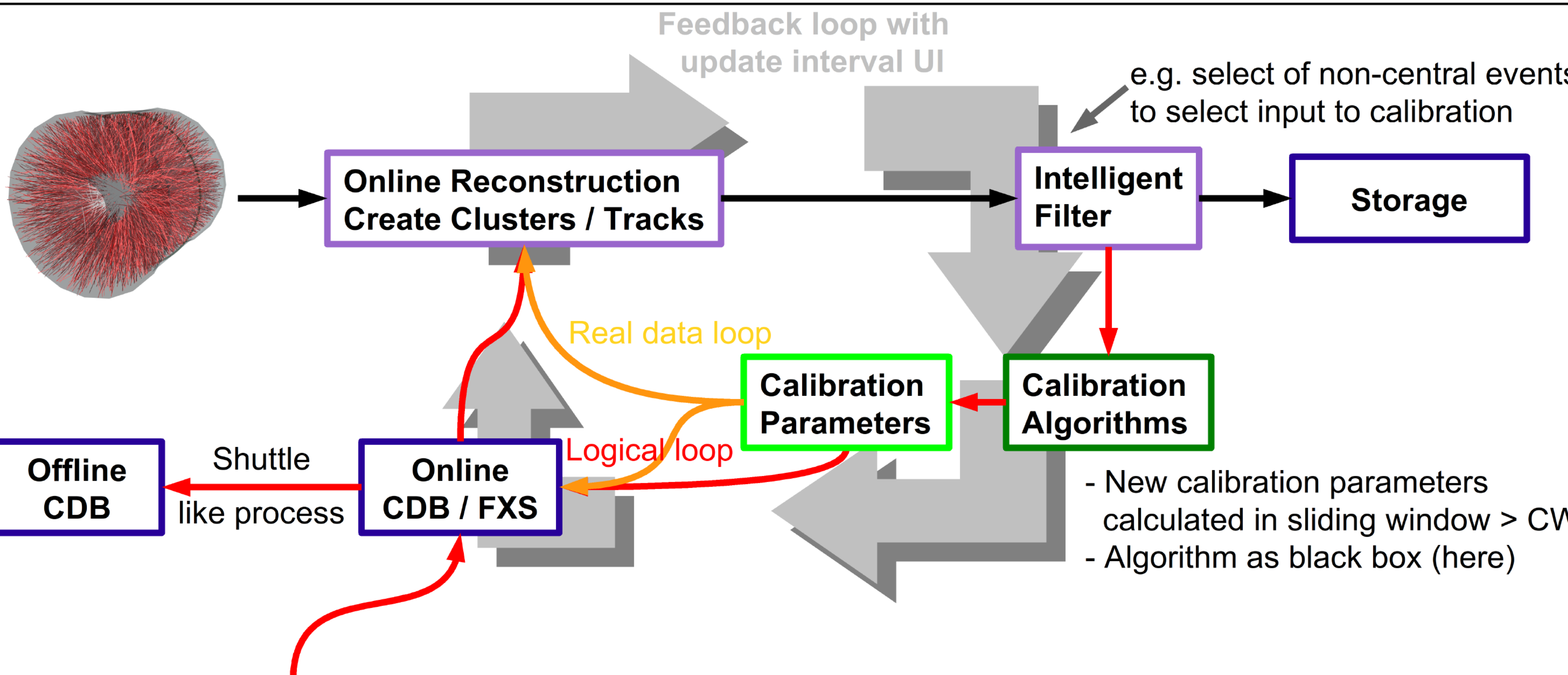
- **Detector calibration stable in time during whole *Update Interval***

- **Subdivide *Update Interval* in 4 steps**
  - Sampling Step (Length determined by calibration needs)
    - Continuous update of the calibration over a sliding window of reconstructed tracks and clusters → *Length(Sampling)* > *Length(Use)*
  - Process Step (minimize length)
    - Generate the calibration in different processes and merge in final calibration process
  - Feedback Step (minimize length)
  - Use Step (maximize length)

Definitions for the Data-Flow (4)



Schematic Data-Flow (1)
Feedback loop



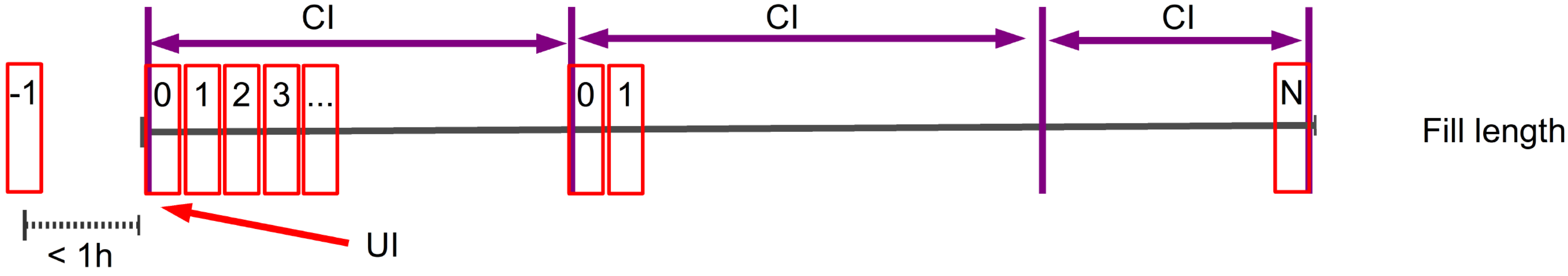
<u>DCS – Values</u>
- Parameters fixed for CI

<u>GRP</u>
Split up GRP in CI parameters
- known at the beginning of CI
- known at the end of CI

Calibration parameters are created every UI and send <u>directly</u> to the reconstruction chain.

In parallel the parameters are filled into a time/eventID based graph, which is stored at the end of fill in the OCDB

Feedback Loop



- **Calibration is stable in 1 *UI***

- **Due to the feedback loop the calibration is improving in the first few *UIs***
  - After LS1: store those events (as they are flagged and might be off-line reconstructed)
  - After LS2: reject events (or flagged)

- **If needed extra *update interval* (*UI -1*) before data-taking**
  - Up to 1h before data-taking
  - Provide initial estimate for some calibration parameters
    - eg. Laser data-taking for the drift velocity

- **Data Quality Control needed to prevent oscillations of the calibration parameters**

QA / DQM → QC

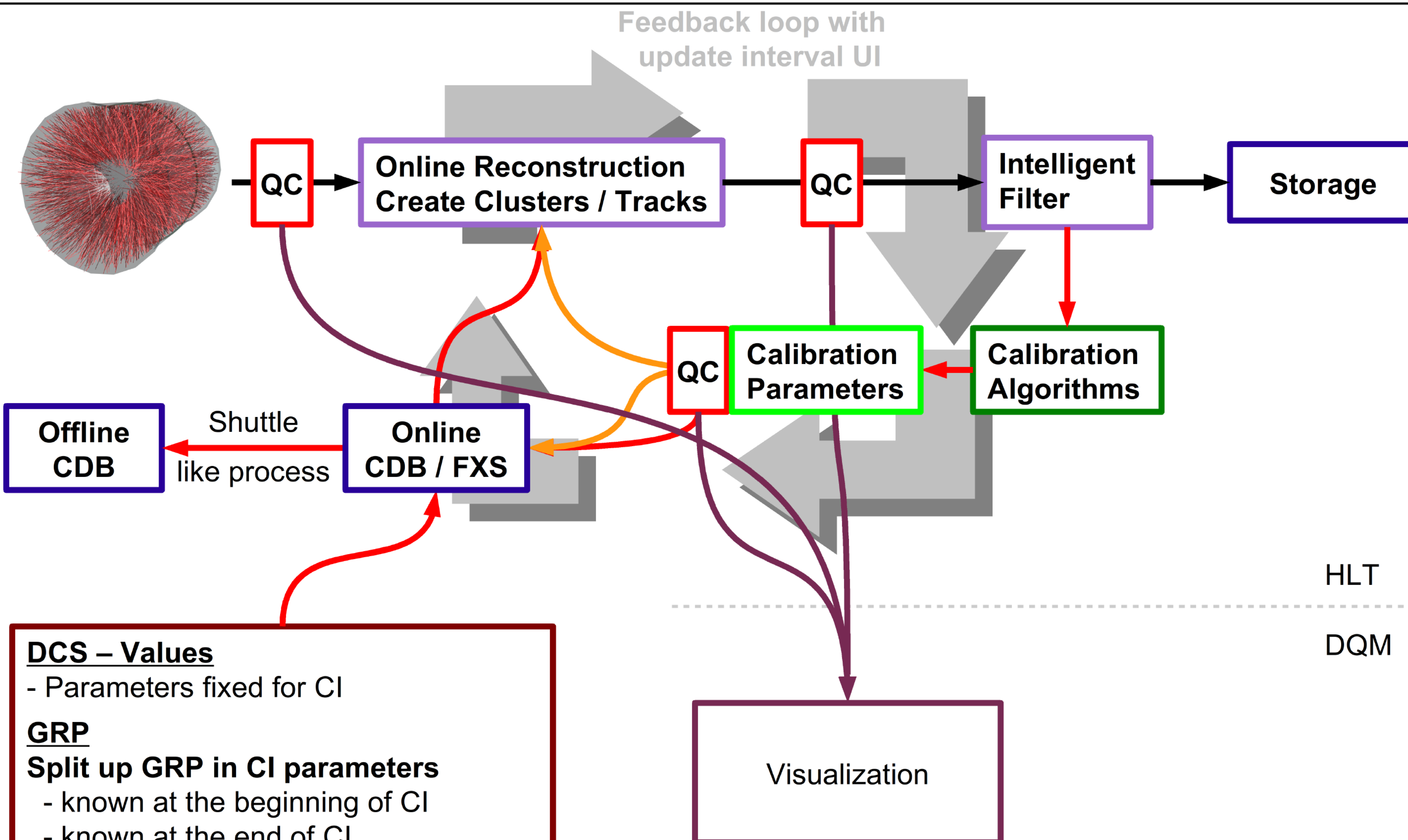
- **Topics**
  - Monitor time stability of calibration parameters
    - eg.: v_{Drift}, gain, ...
  - Monitor time stability of reconstruction performance
    - eg.: TPC-ITS, TPC-TOF, TPC-TRD matching
  - Monitor time stability of physics performance
    - eg.: K^0_S, Λ, PID performance

- **Fast feedback on complex QA tasks is needed to steer the online calibration/data-taking**

- **QC needs to check and validate the calibration parameters before they are sent back into the feedback loop**
  - Apply similar validation step like now in the current calibration schema before updating the OCDB
  - These validations need to be automated within a framework

- **For Run2 we need online QC and offline QA**
  - Preferable the classes can be used in both cases
  - Standard interface needed between AnalysisTask ↔ HLTComponent

Schematic Data-Flow (2)
Quality Control



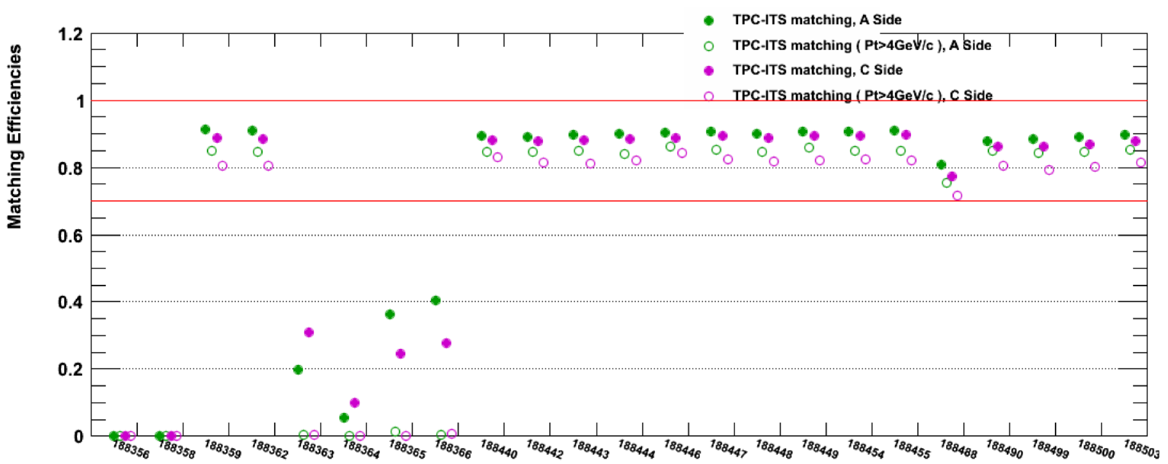
DCS – Values
- Parameters fixed for CI

GRP
Split up GRP in CI parameters
- known at the beginning of CI
- known at the end of CI

DA, QA, DQM … → QC (2)

- **Parameter Trending**
  - We need to see the stability of the physics performance during the course of the data-taking (within one LHC fill)
  - eg. Position of K_s^0 peak , dE/dx stability, TPC/ITS, TPC/TRD matching efficiency
  - Outer parameters (like pressure) could change and alter the detector calibration output (like the drift-velocity for the TPC), however the physics performance has to stay stable
  - **Browsable access must be available from ACR and for the world**

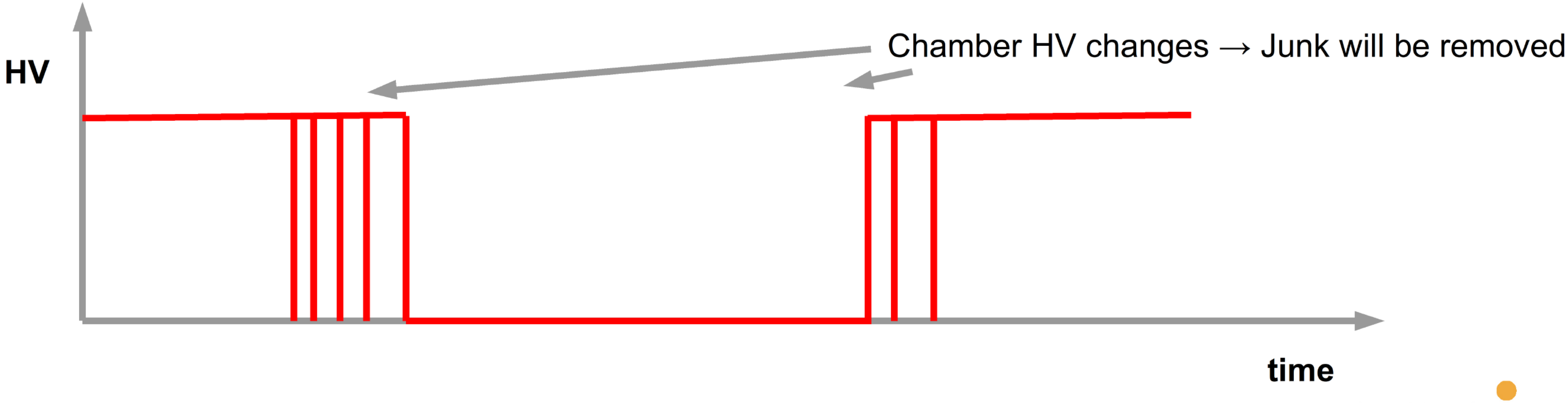
- **Trending is successfully used in TPC off-line QA**
  - Here on run-by-run basis
  - … then on much shorter timescales : O(update interval)



http://www-alice.gsi.de/TPC/PWG1train/data/2012/LHC12h/vpass1/StandardQA/

Handling new running paradigms

- **PaC (Pause and Configure)**
  - Not an issue for the online/offline system
- **RoE (Run over Error)**
  - Readout System will remove “junk” part on an equipment (DDL) in case of
    - Fighting TPC chambers
    - Tripping chambers
  - In case of removal: It is flagged in the trailer of the data per equipment
    - This bit has to be propagated into the ESD in order to recover the events from the first UI of the CI



Baseline strategy (1)

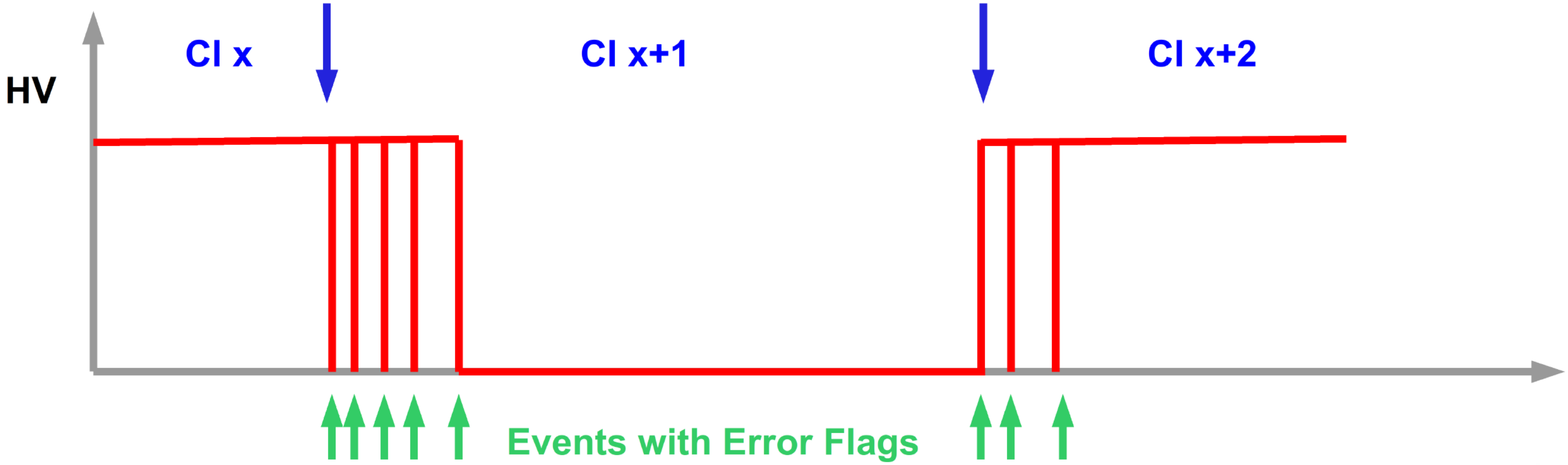
- **In case of one (or a to be defined number of events within a given time – to be studied) event arrives with an Error Flag in the trailer in on equipment**

→ **Start a new Calibration Interval (CI)**

- Increase the Calibration Interval counter
- Remember the 64-bit EventID

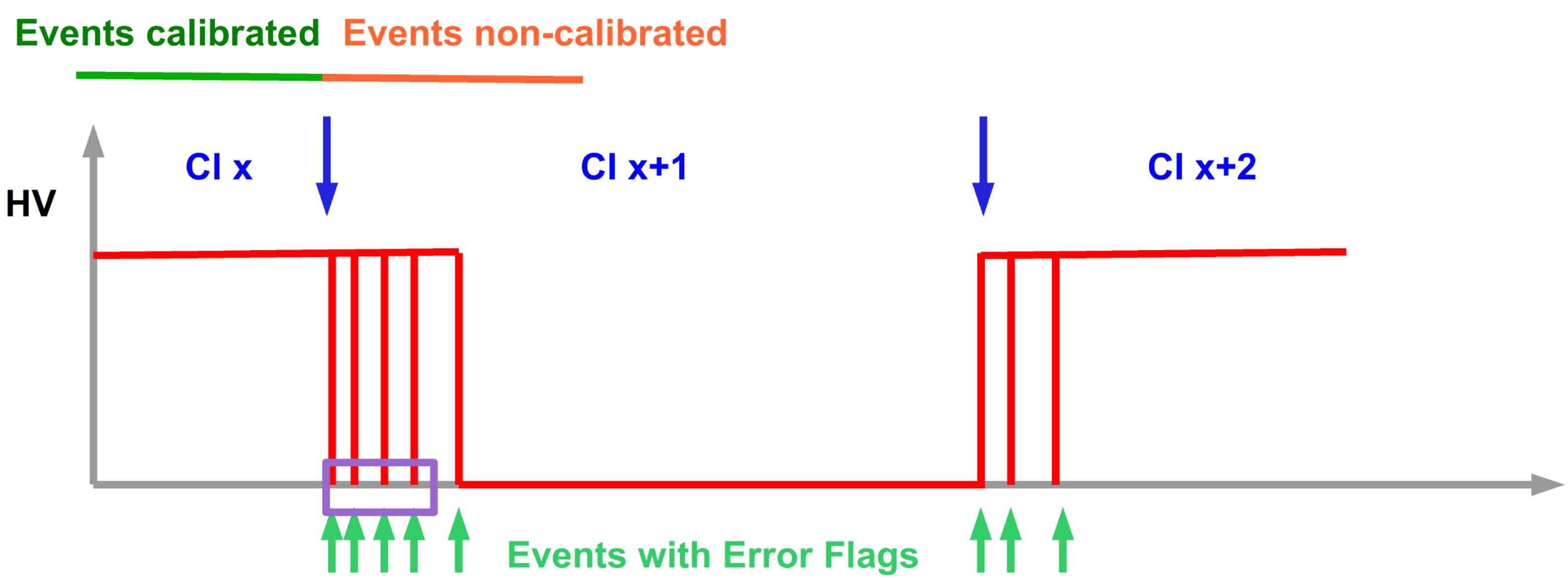
– The reconstruction and calibration is not stopped and just continues, however, the publishing of the sub-calibration objects is stopped ("Feedback" step) (to the merging process is stopped) and the calibration is reset

– All events from now on are flagged as non-calibrated



Baseline strategy (2)

- **Remember: The calibration will run in a sliding window mode**
  - sliding window = Update Interval (UI)
- **From previous slide: "A to be defined number of events within a given time"**
  - If an UI (time length is known) has more then eg. 10% of events with error flag.
- **Events are flagged as non-calibrated**



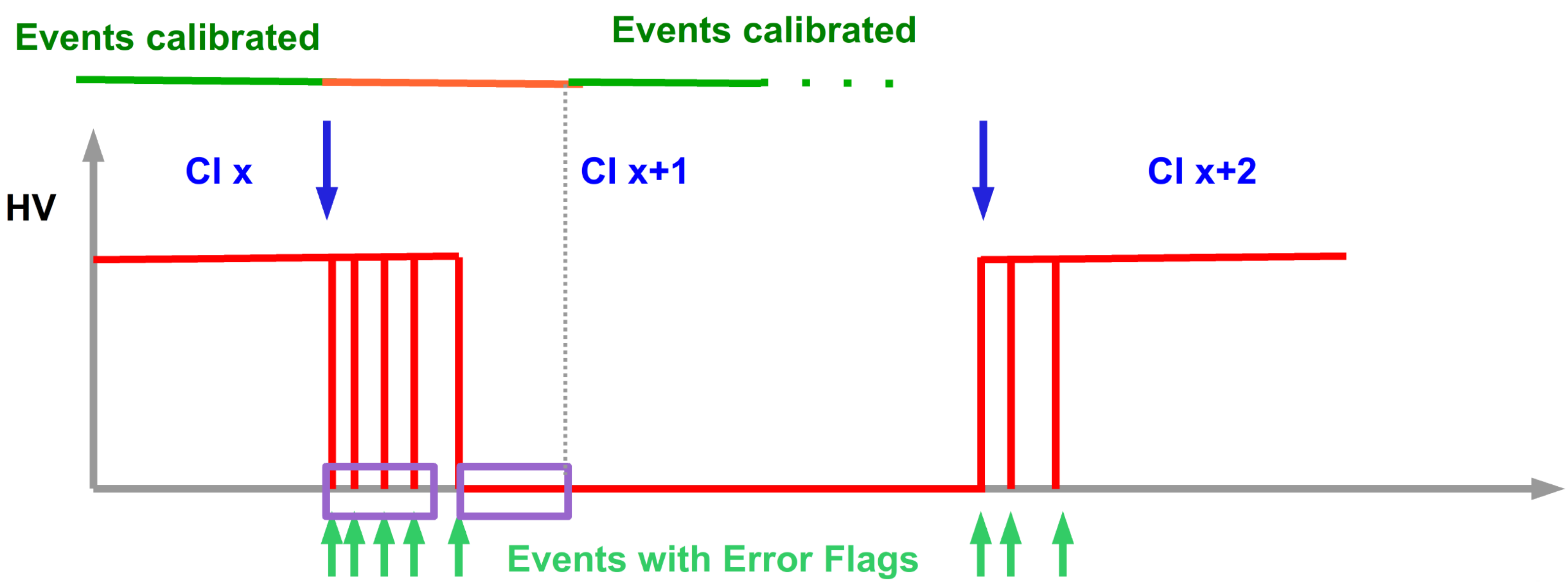
Baseline strategy (3)

- If in an UI no more events with an Error Flag are contained

→ **The calibration can be seen as stable again**

  - Calibration objects are pushed out again for the reconstruction ("Feedback" step)

    → This is the normal behavior of the Feedback loop, with the additional condition



Bookkeeping (in the OCDB)

- **The change of every CI will be stored in a time-based maps indicating:**
  - When (EventID) and which calibration interval (CInumber)
  - The equipment(s) / chamber sending the Error Flag

- **Both above can be filled online**

- **Add the chamber HV values in the preprocesser after the run to indicate the individual chamber status**

Offline processing

- **Reconstruction**
  - No direct implication, as TPC uses time-based calibration objects
- **QA**
  - Needs to be aware of different CI's as the have to be QA'ed separately
    → **Which can create severe problems (memory, merging, etc...)**
- **Simulation**
  - The simulation has to be anchored to the CI's
  - The relative distribution of statistics of the different CI's has to be taken into account
- **Analysis**
  - Non-calibrated events are flags online (negative CInumber) and removed by PhysicsSelection
  - Analysis should be done CI-wise

Remarks on CI vs Run

- **The procedure for the TPC online calibration and reconstruction is now independent of this**
- **BUT offline QA is not**
- **For now, we would prefer a on-the-fly change of the "run number"**
  - The relevant item is the separate storage of raw data per "run" / CI