Offline Week, 5-8 November 2013

R. Shahoyan, C. Zampolli

CALIBRATION AND RECONSTRUCTION: PREPARATION FOR RUN2

Layout of the presentation

- Focus on Run2
  - Online implementations
  - Run-Over-Error
- Calibration
- Reconstruction

} Some points will be in common/shared

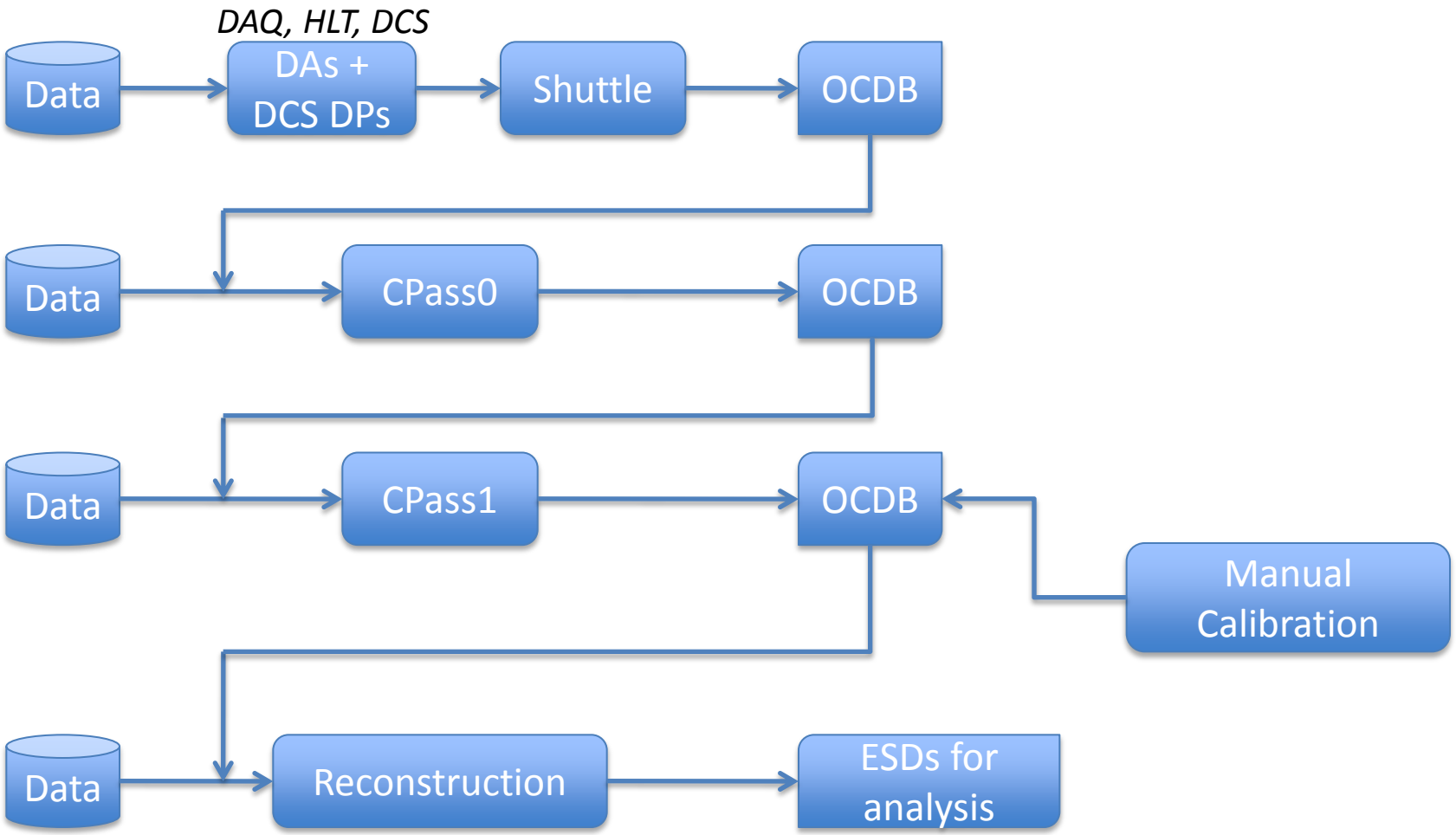
Calibration

What could we do in Run2

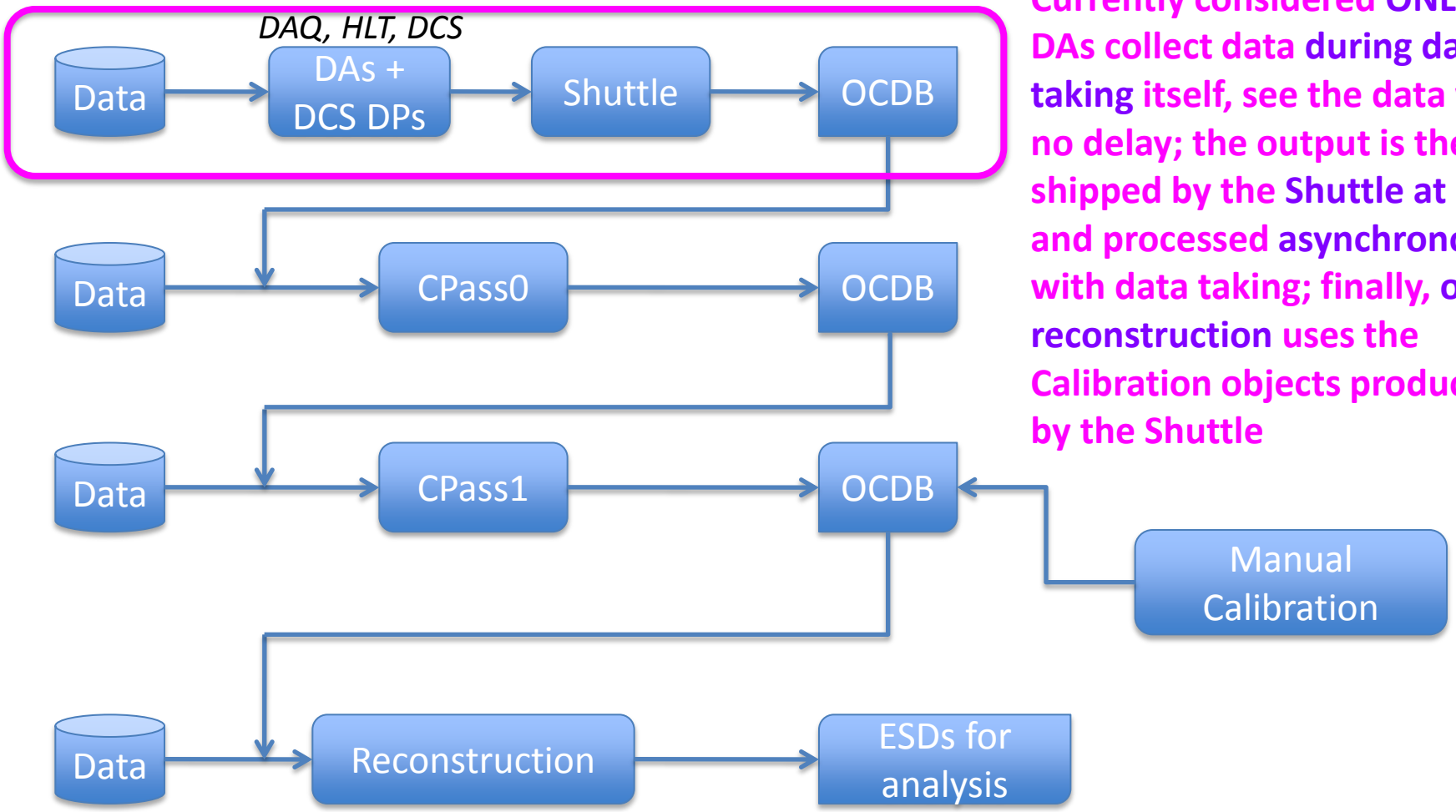
- Run2 could be used to test procedures for Run3, where calibration should be done online as much as possible to allow to reduce the data size
- Focus on calibrations that are done during PHYSICS runs, since those are to be obtained and used online in case of an online reconstruction
- Calibration summary table from Run1 used for the following slides as starting point
- Main focus for online calibration at Run2 on the detectors that will be there for Run3
  - No exclusion, only to not put useless pressure/ demands in the cases when it is not needed → if anybody wants to join the effort, be welcome!

Obviously, implementing anything in HLT for Run2 needs the HLT help

Current Calibration schema



Current Calibration schema



Currently considered ONLINE: DAs collect data during data taking itself, see the data with no delay; the output is then shipped by the Shuttle at EOR, and processed asynchronously with data taking; finally, offline reconstruction uses the Calibration objects produced by the Shuttle

OCDB

- **Offline Condition DataBase**
- Stored in Alien
- Use of **Metadata** to identify an object
  - 3 level path
  - **Run number/range validity**
    - The Shuttle and CPass fill only objects valid for current run, at the end of the corresponding processing (i.e. EOR for the Shuttle, end of calibration procedure after the whole run has been processed for CPass)
  - Version number
    - Becomes subversion when you work locally
- Objects can be of any kind
  - Histograms, parameter, graphs, maps…
  - In the TPC case, already time-dependent, and plan is there to use HV… in simulations

CPass principle

- Introduced to allow calibrations that need **ESD as input**
  - Reconstructed data!
- Usage of **two passes** allows:
  - To resolve interdependencies between detectors
    - Run1: TOF from TPC; TRD from TPC
  - To refine calibrations
    - Run1: TPC, TRD
- Final reconstruction (→ aimed at Physics) is a third pass over the raw data
- **Quality Assurance** always included
  - In terms of QA train, and Validation Pass

What can we do at Run2

- Run2 can be used to:
  - Implement online procedures to be used at Run2 for Physics results → **"Run2 for Run2"**
  - Test online procedures, not aimed at Run2 Physics, but in view of Run3 → **"Run2 for Run3"**

- "Run2 for Run2":
  - **TPC online calibration**
    - Schema presented by Jochen Theader last spring
    - Would allow to remove CPass1
      - Unless TRD calibration refinement is needed
        - » Could TRD use the same approach as TPC?
    - Calibration objects created online can then be merged, put in the OCDB and used for CPass0

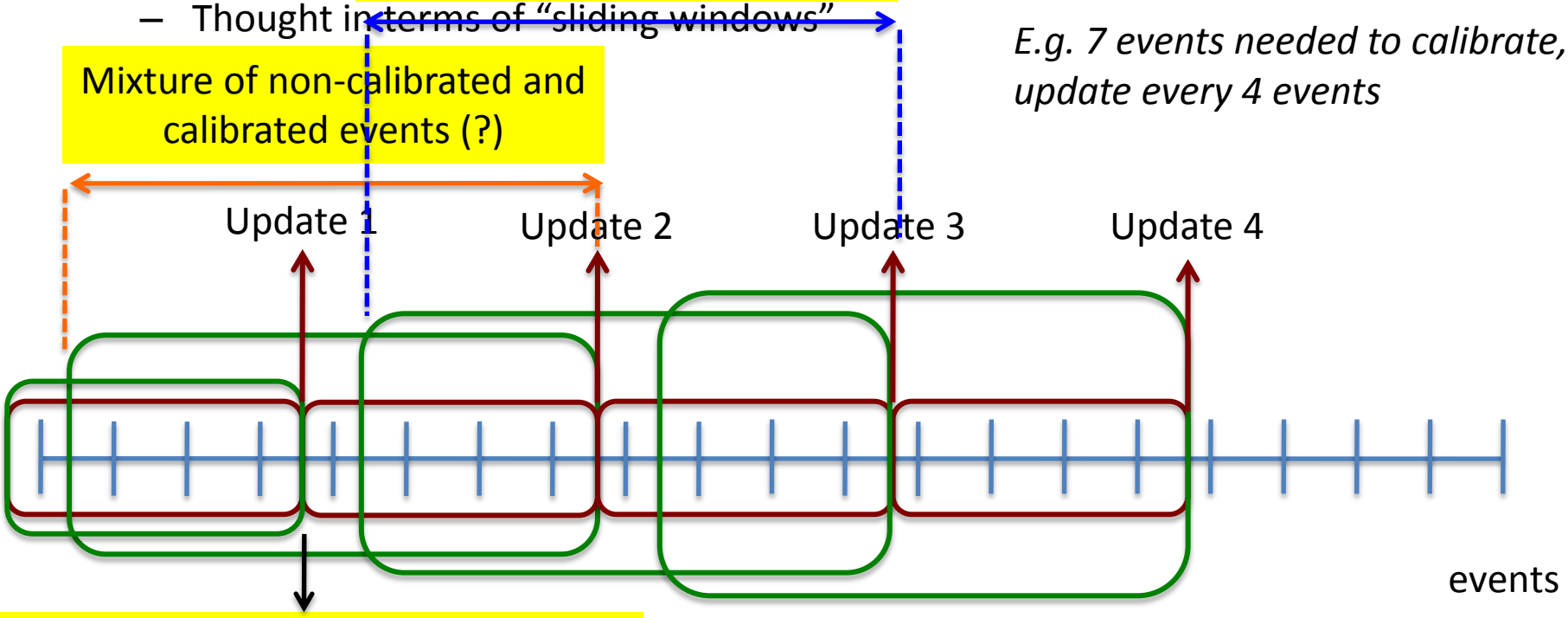
TPC Online Calibration: “Prediction Step”

- No general p…en to update the calibration parameters
  - Thought in terms of “sliding windows”

All calibrated events used to produce new calibration

Mixture of non-calibrated and calibrated events (?)

E.g. 7 events needed to calibrate, update every 4 events



Events before here are NOT calibrated

Events used to produce calibration parameters

Update after event x will have been obtained from events in [x-7, x] and will be used in [x+1, x+4]

“Run2 for Run2” online calibration - ingredients

- To implement **TPC online calibration, online reconstruction** is needed:
  - TPC online reconstruction
    - TPC HLT reconstruction already there
    - Vertex reconstruction in HLT already there
    - SPD HLT reconstruction already there
    - SDD HLT reconstruction NOT there
      - Time-consuming clusterization, not straightforward
    - SSD HLT reconstruction already there
  - If **TRD** wants to join the TPC effort
    - TRD HLT reconstruction already there

“Run2 for Run2” online calibration – ***missing*** ingredients

- At present, reconstruction in **HLT** does not use the “Predictive step” approach: calibration and reconstruction are completely separate processes (components) which run event-by-event
  - Basic ingredients to start: one reconstruction process, one calibration processes (even dummy!) that uses the reco output, and produces an output itself to be used in the subsequent reco…

“Run2 for Run3”: other online calibrations

- Using **Run2 as playground for Run3**
  - **DA concept** can be moved to online reconstruction
    - No waiting for EOR and Shuttle before having the calibration objects
    - Possible approach:
      - Creating calibration objects as soon as the minimum required statistics is reached → time-dependent objects!
      - Using the calibration object to reconstruct the data they correspond to
      - The two processes should be implemented as components
    - Need implementation of a DA in HLT in the form of a component
      - Using running averages produced in the calib component to be passed to the reco component
        - » First events not calibrated, unless approximated calibration from previous run is used
      - Using average over previous 1000 events, for the subsequent 1000 events
      - Same object should be distributed to all “reconstruction nodes”
        - » 1 calibration node only
        - » 1 further component to merge the calibration data before reco
    - Similar approach could be used for DCS Data Points

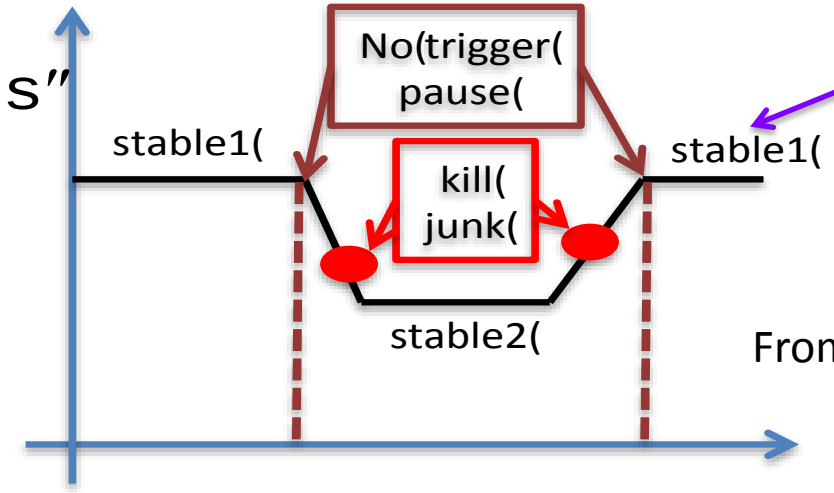
“Run2 for Run3”: another possible idea

- Could the HLT reconstruction be used for **data reduction**?
  - E.g.: removal of noise channels
    - Other kinds of data reduction/optimization: where? On DAQ? On HLT?
  - Useful if HLT output is then used for actual reconstruction “a’ la Run3”
  - Is there any interdependence between detectors that plays a role at data reduction stage?
  - Could be an overkill for people, if the reduction factor is not significant

- More in terms of reconstruction: an HLT reconstruction aimed at data reduction using calibrations could be used as an exercise to “simulate” what will happen on the EPNs
  - Should store compressed data, from which further reconstruction aimed at Physics could start

Run Over Error and Calibration

- **Run Over Error** aimed at increasing/improving ALICE data taking efficiency
- Data taking not in terms of "run" but more in terms of **"stability periods"**



Typically to be considered as different from before! → "stable 3"

From CTF meeting on 30 Oct

- Between "stable 1" and "stable 2", junk from "faulty" part of a detector is removed
  - It is as if you were already in "stable 2"
- At "stable 2", part of a detector is typically in a different (but stable) condition
  - Off, different voltage…
- How will this be propagated to raw data and ESDs?

Stability periods flag

- We suppose that no flags are propagated in the data to signal whether *each* event belonged to stability period 1, 2, 3… for detector X, Y, Z…
- We suppose that the data from different stability periods will be written in the raw data files as now
  - Mixed in different files
  - No time ordering

DAs and Shuttle

- DAs do not need to be modified
  - This supposes that differentiating the calibrations for the different stability periods is done at Shuttle level, and can be based on DCS information retrieved there
- The Shuttle will process all the data at the End of Run (as now!)
- The preprocessors should prepare calibration objects that are "time-dependent" with a minimum granularity corresponding to the duration of the stability periods (*)
  - Defined by the DCS values
- One file per calibration will be created at the end in the OCDB (as now!)

(*) for those detectors that are foreseen to change "stability" w/o stopping the run

CPass

- Events from different stability periods should be used to produce different calibrations
  - Even if a detector was not directly involved in the event that triggered the "stability period" change, it should take it into account
    - CPass is a calibration procedure that depends on globally reconstructed data, different from DAs + Shuttle
  - Possible reading flags stored in the data themselves
  - Merging job should be split internally in the detectors' code

GRP

- A new entry should contain the list of stability periods
  - Number of stability periods in a run
  - Start, end (timestamp or period/orbit/bc) of each stability period
  - Number of events in each stability period (or luminosity)
  - Detector that triggered the change
- Information could be retrieved from the logbook? Should it access some specific DCS information?
- Should help also to anchor properly MC simulations

Backward Compatibility

- Some OCDB objects that are currently not time-dependent could need to become time-dependent
  - One possible way: Encapsulate “sort-of-TObjArrays” in AliCDBEntry to make it transparent to the user
    - See Reconstruction part

Reconstruction

Reconstruction in Run2

Offline: no significant change wrt Run1

- Alignment to be redone after LS
- In preparation: global alignment/calibration schema (see next slide)
- TRD to be enabled in the "refit" track update (good alignment $\Rightarrow$ factor 2 in high p_T resolution)
- TOF works on improving reconstruction algorithm to decrease contamination by mismatches.
- "Run-over-error":
  - if each stable period leads to separate "mini-run": transparent for the reconstruction
  - if we switch to time-dependent calibration objects: trivial changes in reconstruction code:
    - ensure that all detectors update CDB objects at every event via AliCDBEntry::Get(timestamp)
    - can be done in a backward compatible way with minimal change in the OCDB framework:

```
TObject* AliCDBEntry::GetObject(timestamp=0) {
  if (fObject->IsA()==WrapArrayClass) return ((WrapArrayClass*)fObject)->Get(timestamp);
  else return fObject;
}
```

Online (HLT)

- ITS standalone tracker (w/o SDD): was asked by TPC for HLT calibration
  (but can leave also with current HLT TPC->ITS matching code)
  I.Kisel (Frankfurt Uni group) volunteered to adapt CBM based tracker (also as a benchmark for Run3 ITS online reconstruction), but no news so far.

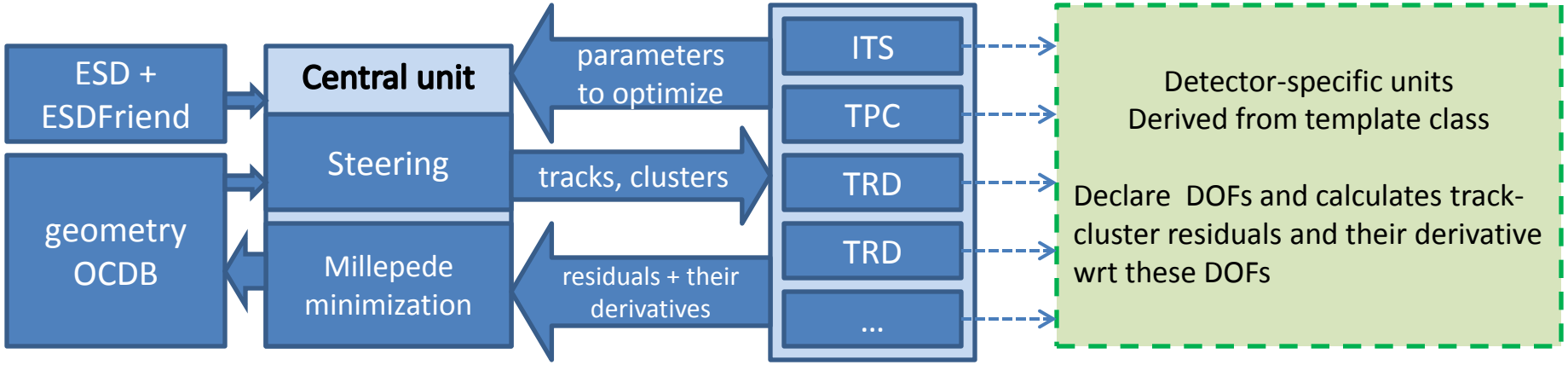
Plans for global alignment/calibration framework

- Currently ITS and TPC are aligned internally, then they are aligned globally one with respect to other
- Outer detectors aligned relying on TPC tracks, w/o any feedback on TPC alignment calibration
- At least for TRD this does not lead (after separate calibration) to target residuals, presumably due to:
  - the residual miscalibration of TPC tracks
  - entanglement of TRD alignment and calibration

  This prevents using the TRD info for the kinematics update in RefitInward

  $\Rightarrow$ Need global alignment/calibration

$\Rightarrow$ Use Millepede algorithm for simultaneous refitting of detector's alignment/calibration parameters and tracks (already used for ITS and Muon alignment).



Work started but is stuck due to the upgrade TDR's problems

Backup

Run2 Running Conditions

From F. Ronchetti, TB @ Wuhan

ALICE Running Conditions

Year	System	E [TeV]	Lumi [$cm^{-2}s^{-1}$]	Rate [kHz]	Level	Weeks	Trigger
2015	p-p	13	$1\text{-}2*10^{29}$	10-20	YES	24	MB
	Pb-Pb	5.1	10^{27}	8	YES	4	
2016	p-p	13	$0.5\text{-}1*10^{31}$	500	YES	24	RARE
	Pb-Pb	5.1	10^{27}	8	YES	4	
2017	p-p	13	$0.5\text{-}1*10^{31}$	500	YES	24	RARE
	p-Pb	8.2(?)	$0.5\text{-}1*10^{28}$	10-20	YES	2	MB
	p-Pb	8.2(?)	10^{29}	200	YES	2	RARE
2018	Long Shutdown 2						

- For 2015 we need **5 orders of magnitude luminosity reduction in IP2**
- **Almost all bunches (~2500) collide in ALICE** (except abort gap)