Introduction to the DQ analysis framework

Ionut-Cristian Arsene (University of Oslo)
on behalf of the O2 DQ group

O2 Analysis tutorial 4.0
16/10/2024

Overview

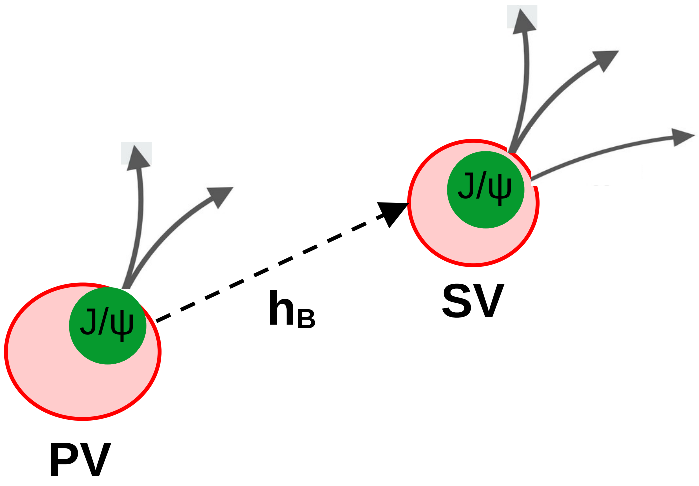
- Motivation and physics cases
- O2-DQ software structure
- Skimmed data model
- Workflows
  - Data filtering and skimming/slimming
  - Data analysis
- Pseudo-code example for using the framework utilities
- Summary

Motivation and physics cases

- Dedicated framework to work in particular with (di-)leptons, both in the MUON arm and Central Barrel
- Common tools and workflows prepared by experts in working with electrons in the barrel and muons with the MUON arm and MFT

Motivation and physics cases

- Dedicated framework to work in particular with (di-)leptons, both in the MUON arm and Central Barrel
- Common tools and workflows prepared by experts in working with electrons in the barrel and muons with the MUON arm and MFT
- Physics cases:
  - Quarkonia decaying to dileptons
  - Low mass dileptons
  - Heavy-flavour (single) leptons
  - $B^{\pm,0} \rightarrow J/\psi + K^{\pm,0}$, $B_c \rightarrow 3$ muons, $\chi_c \rightarrow J/\psi + \gamma$, $X(3872) \rightarrow J/\psi + \pi^+\pi^-$, etc.
  - Associated quarkonia production:
    - jets, charged hadrons, open HF hadrons, double quarkonia, etc.
  - Quarkonia and open heavy-flavour in ultra-peripheral collisions



Framework Overview

- **Code in O2Physics/PWGDQ/**

Framework Overview

- **Code in O2Physics/PWGDQ/**
- **PWG-DQ standalone data model (`ReducedInfoTables.h`)**

Framework Overview

- **Code in O2Physics/PWGDQ/**
- **PWG-DQ standalone data model (`ReducedInfoTables.h`)**
- **Configurable workflows:**
  - Event filtering for pp/p-Pb and Pb-Pb: `filter-pp, filter-pbpb`
  - Skimming/slimming of data/MC to a PWG-DQ standalone data model: `table-maker, table-maker-mc`
  - Analysis of skimmed data/MC: `table-reader, dq-efficiency`

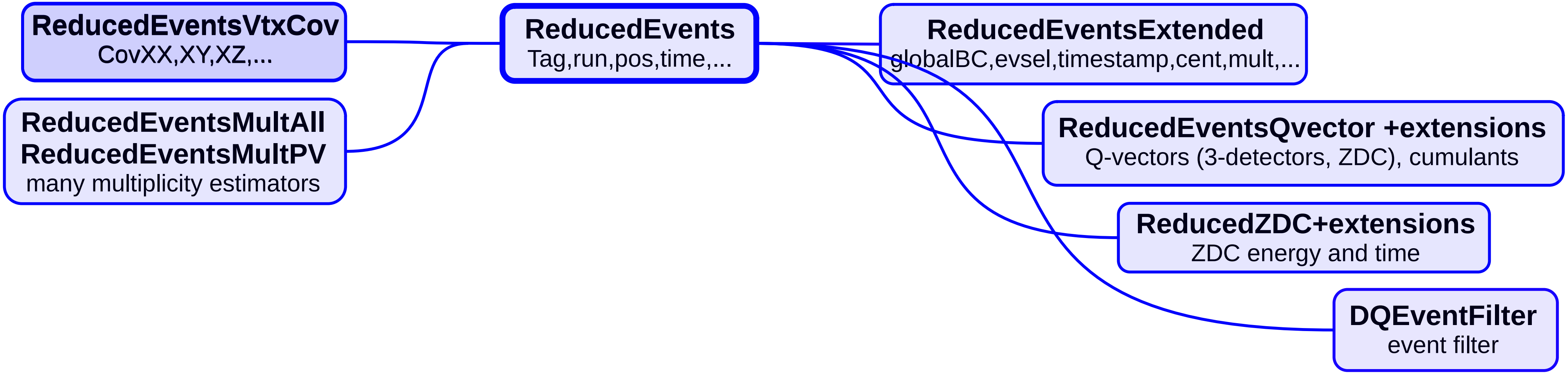
Framework Overview

- **Code in O2Physics/PWGDQ/**
- **PWG-DQ standalone data model (`ReducedInfoTables.h`)**
- **Configurable workflows:**
  - Event filtering for pp/p-Pb and Pb-Pb: `filter-pp, filter-pbpb`
  - Skimming/slimming of data/MC to a PWG-DQ standalone data model: `table-maker, table-maker-mc`
  - Analysis of skimmed data/MC: `table-reader, dq-efficiency`
  - Tag-and-probe for QA, calibration and data-driven efficiency
    - V0 selector for clean PID track samples: `v0-selector`
    - Dalitz electrons selector (primary electrons): `dalitz-selection`
  - MUON data driven efficiency workflows: `task-muon-mch-trk-eff`, etc.
  - Event Q-vector and other event-wise correlators creator: `dq-flow, dq-correlation`

Framework Overview

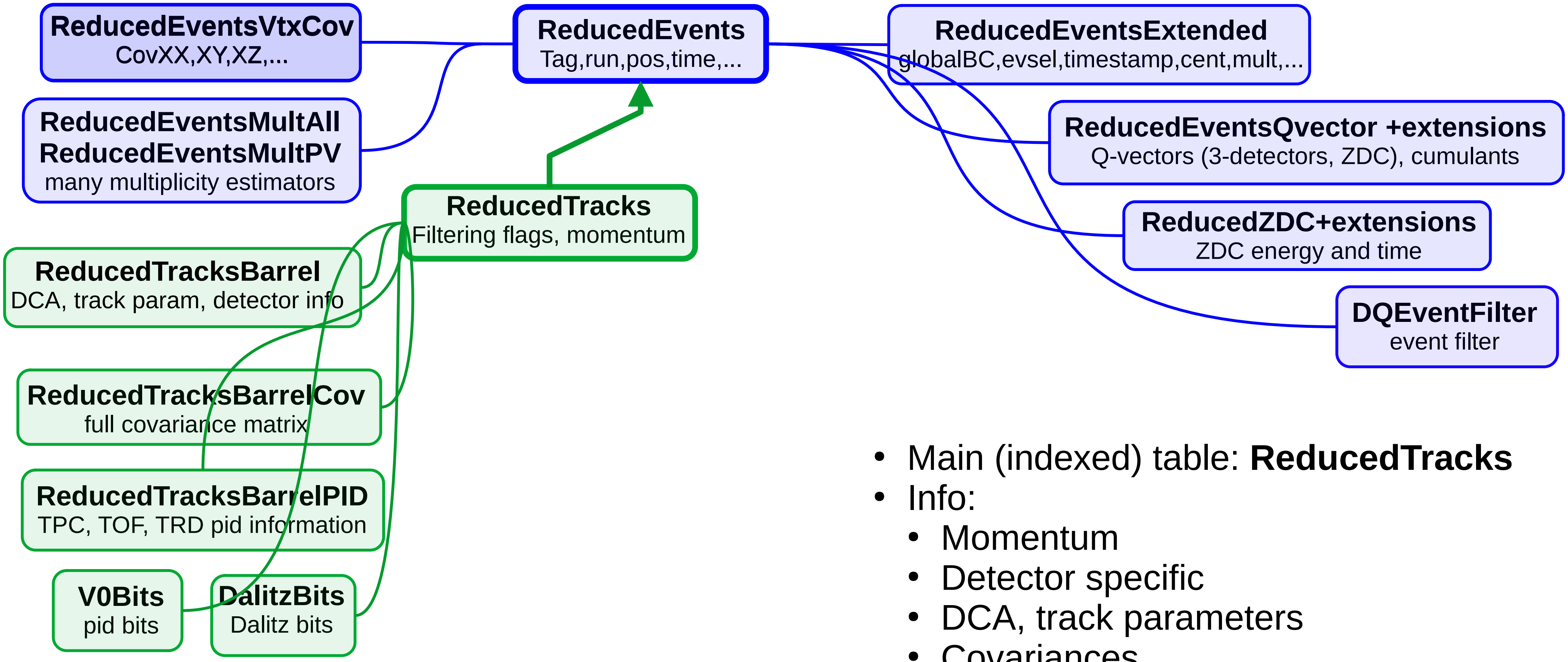
- Utilities
  - `VariableManager:` computes all quantities required in an analysis
  - `HistogramManager:` histogram definition, filling and handling
  - `AnalysisCut, AnalysisCompositeCut:` analysis cuts
  - `MCProng, MCSignal:` Monte-Carlo particle stack handling
  - `MixingHandler:` event mixing handling
  - Libraries: `HistogramsLibrary, CutsLibrary, MCSignalLibrary`

DQ derived data model: events



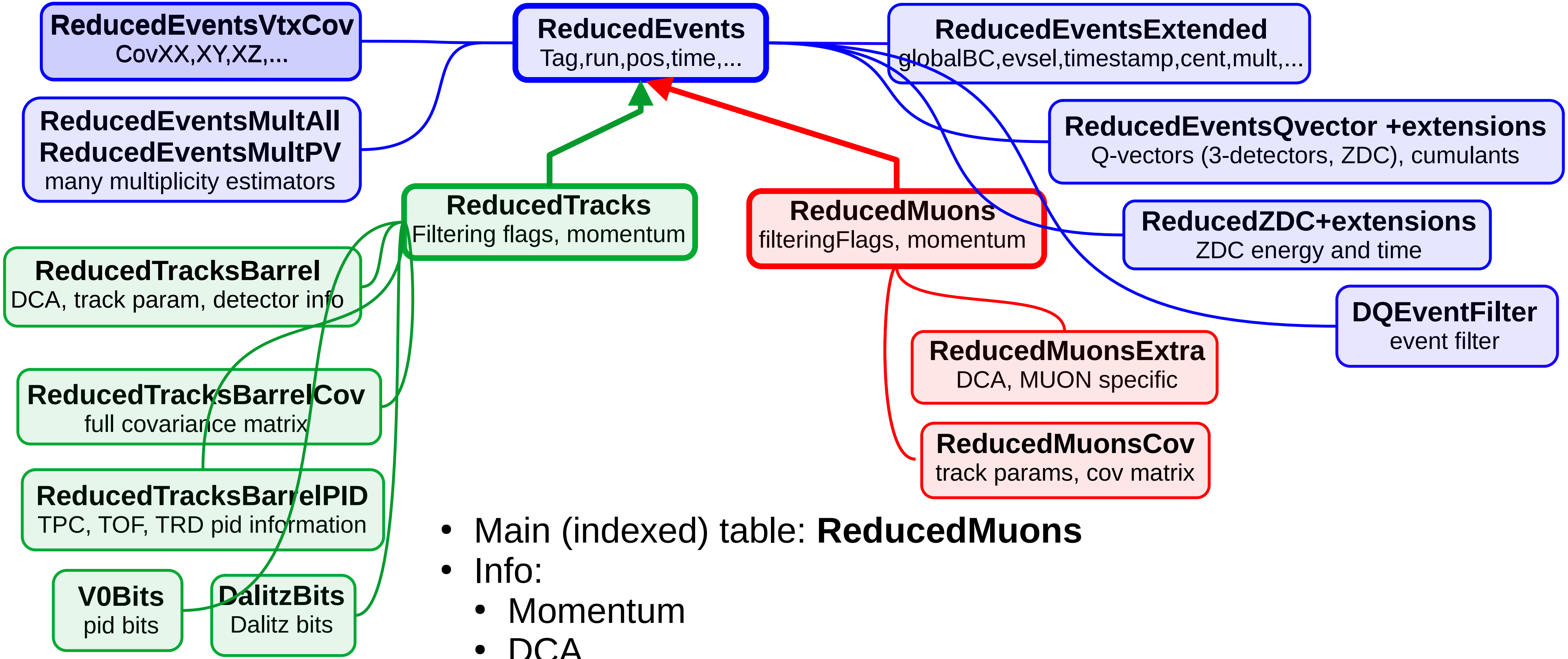
- Main (indexed) table: **ReducedEvents**
- Event information stored in several different tables
  - Main
  - Covariances
  - Multiplicities
  - Q-vector, etc.

DQ derived data model: barrel tracks



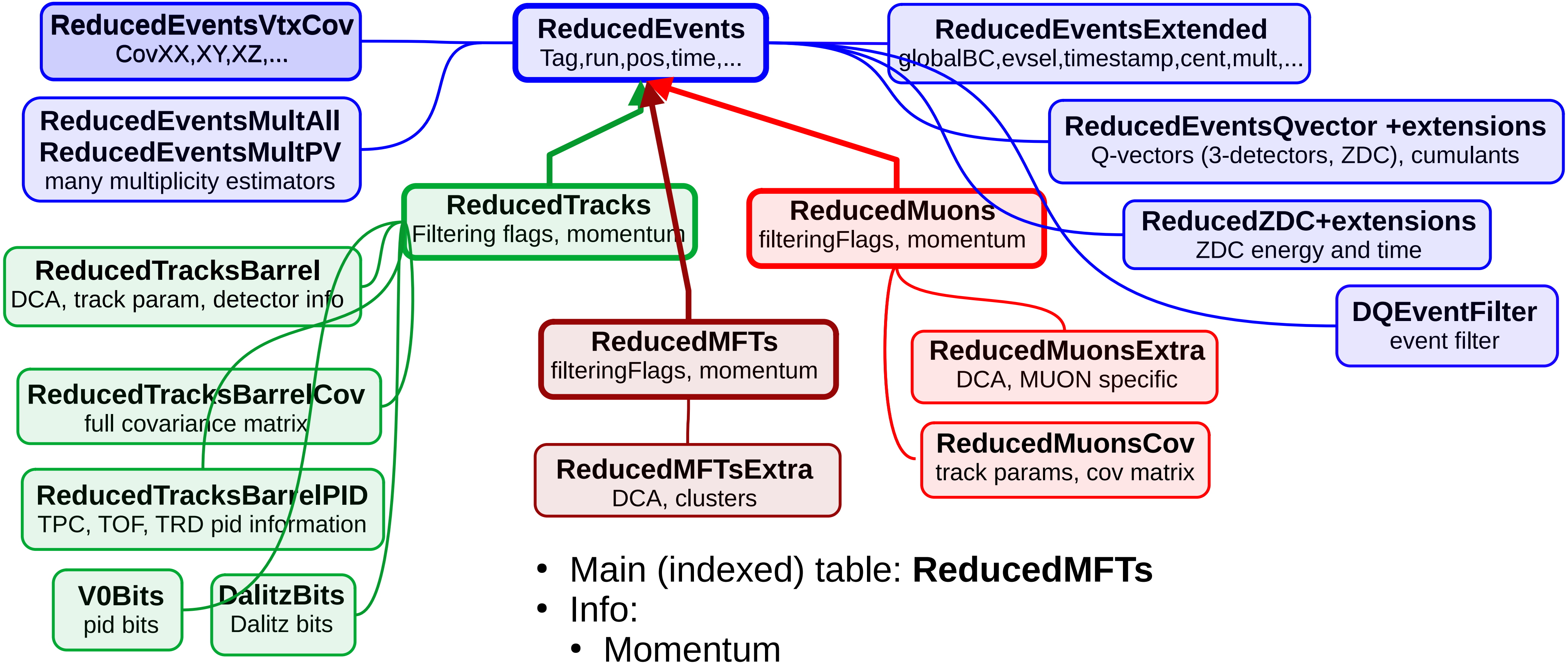
- Main (indexed) table: **ReducedTracks**
- Info:
  - Momentum
  - Detector specific
  - DCA, track parameters
  - Covariances
  - PID
  - V0 and Dalitz info

DQ derived data model: muon tracks



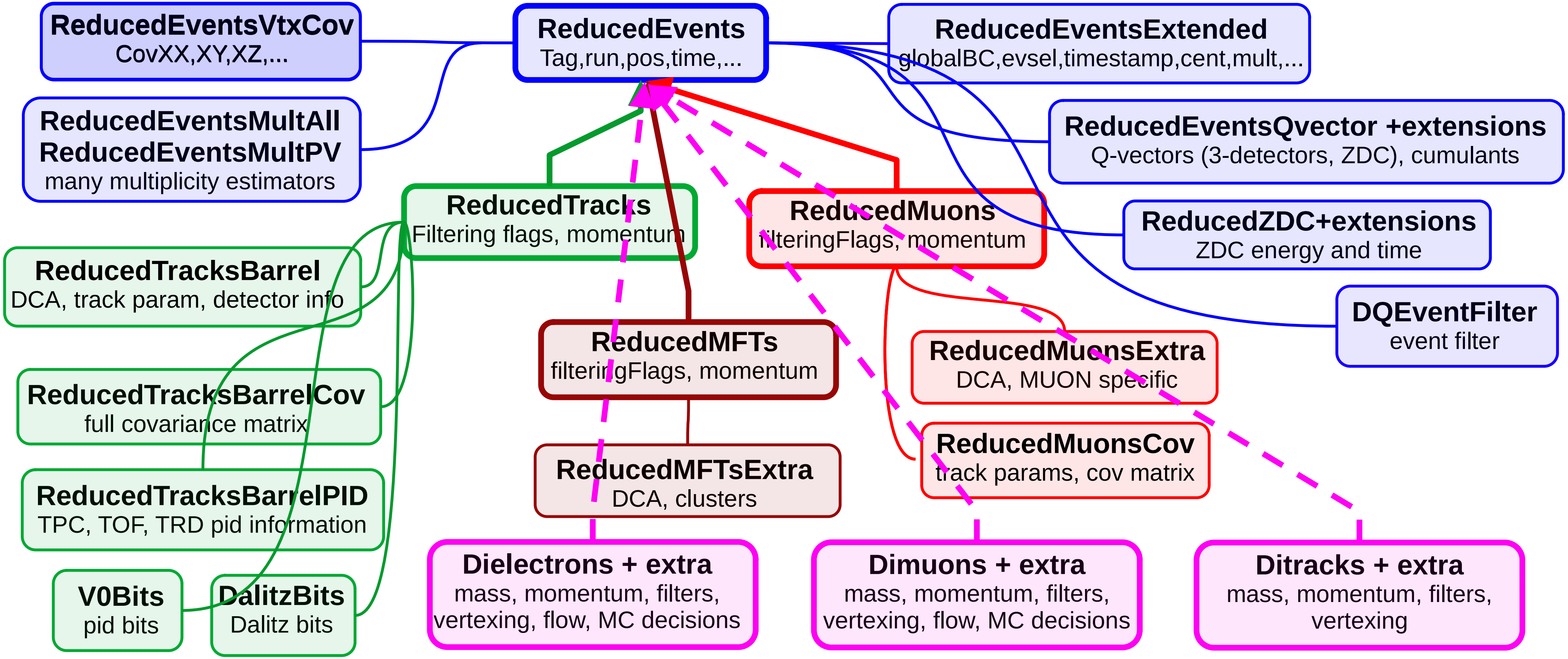
- Main (indexed) table: **ReducedMuons**
- Info:
  - Momentum
  - DCA,
  - track parameters, covariances
  - MUON specific

DQ derived data model: MFT tracks



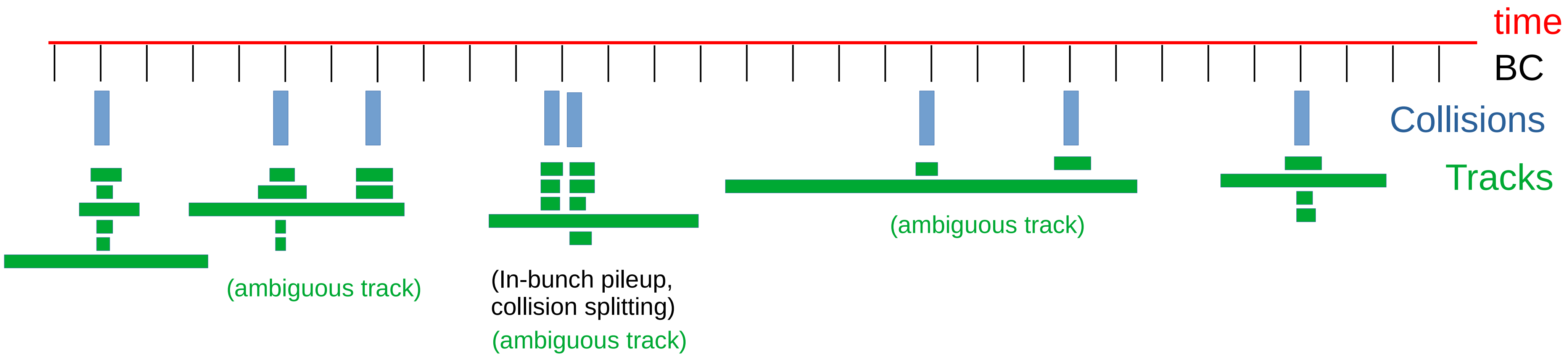
- Main (indexed) table: **ReducedMFTs**
- Info:
  - Momentum
  - DCA
  - clusters

DQ derived data model: high level tables



- Various high level tables produced during analysis

Track to collision ambiguity



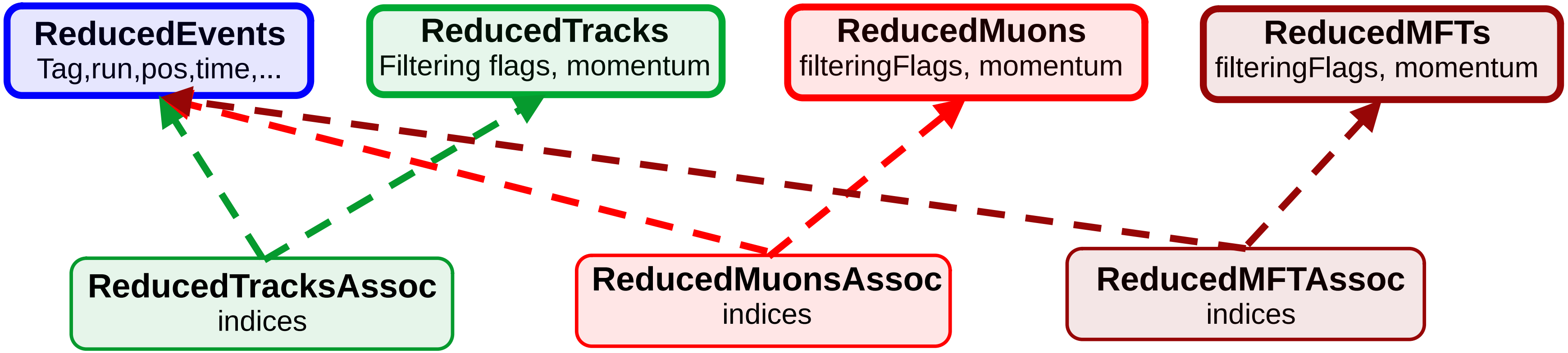
- **Ambiguous tracks**: track time resolution allows compatibility with more than one collision
- Time resolution scales:
  - TPC only: ~100 μs
  - ITS+TPC: 1-15 μs
  - ITS+TPC+TRD/TOF: ~1 ns
- **Orphan tracks** (tracks with no collision association)

Sources of ambiguity:

- Close in time out-of-bunch pileup
- In-bunch pileup
- Collision splitting, ...

Association tables can be written to disk as part of the derived data model

DQ derived data model: **associations**



- Association tables for barrel, muon and MFT tracks can be written in the data model
- One track may appear in more than one association (associated to different events)
- Some track or pair properties need to be recomputed for each association at analysis time:
  - DCA, secondary vertexing, ...
- **Analysis loops over associations instead of tracks!**
  - Ideal case: after analysis selections, only the “correct” association remains
  - Real life: some fraction of multiple-counting of tracks or signals (e.g. jpsi candidates)

DQ derived data model: MC event

(Reconstructed level)

(MC truth level)

ReducedMCEvents
vtx, time, weight, ...

- Top MC truth node: **ReducedMCEvents**
- MC event-wise information: vtx position, time, MC weight, etc.

DQ derived data model: MC tracks

(Reconstructed level)

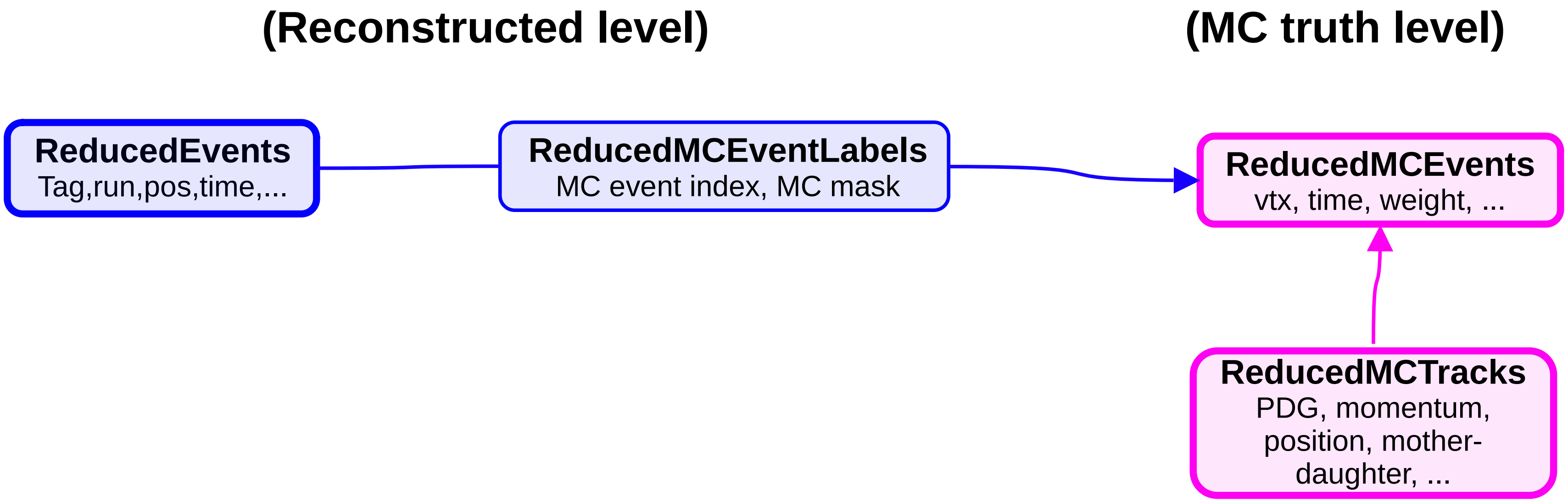
(MC truth level)

ReducedMCEvents
vtx, time, weight, ...

ReducedMCTracks
PDG, momentum, position, mother-daughter, ...

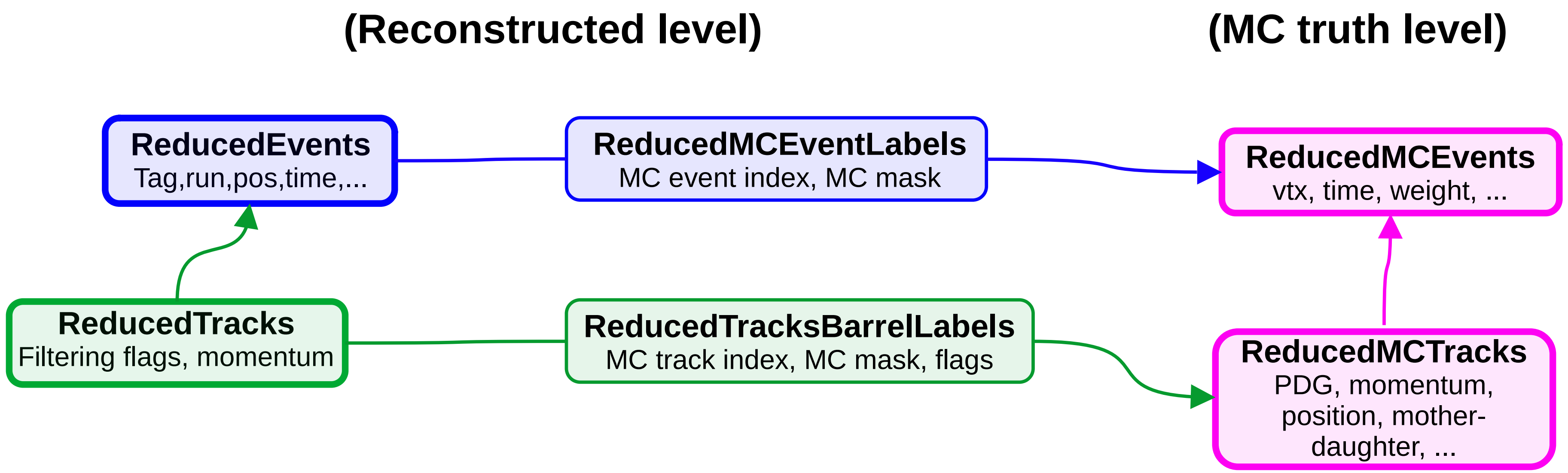
- MC truth tracks table: **ReducedMCTracks**
- MC truth tracks can be filtered using DQ framework utilities, such as **`MCSignal`**
- Mother-daughter relations recomputed at skimming time

DQ derived data model: event labels



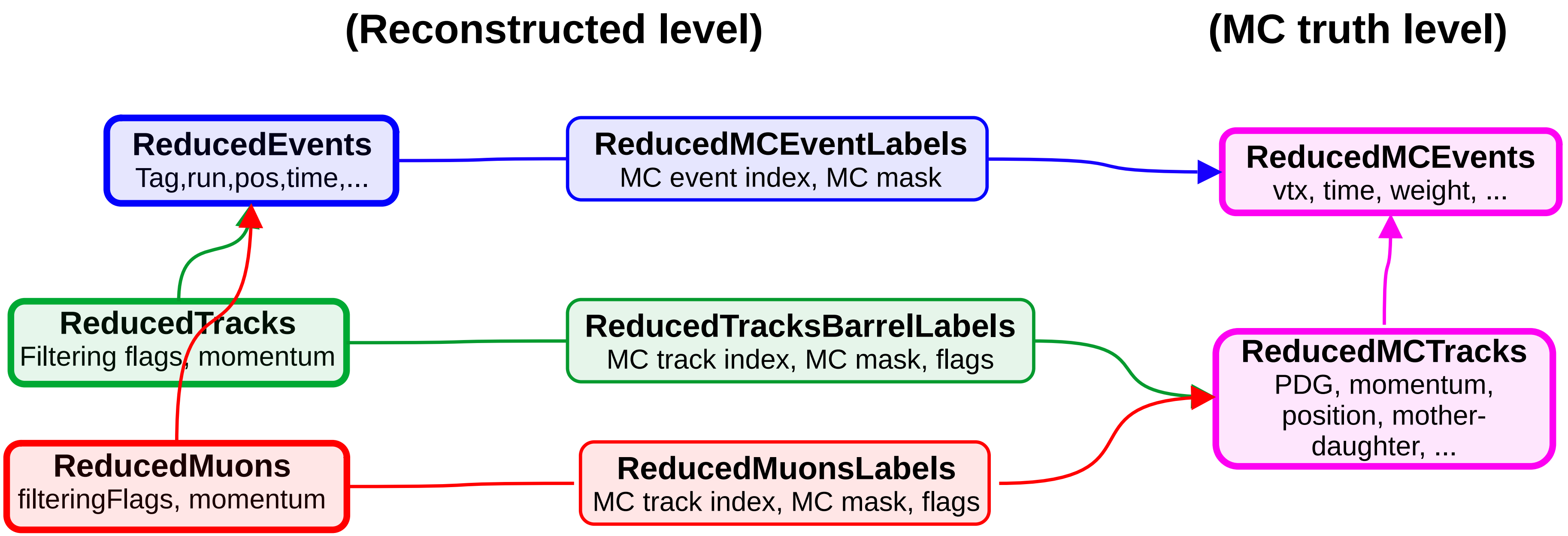
- The connection between the reconstructed-level tables and the MC truth tables is done via label tables, similar to the O2 framework

DQ derived data model: track labels



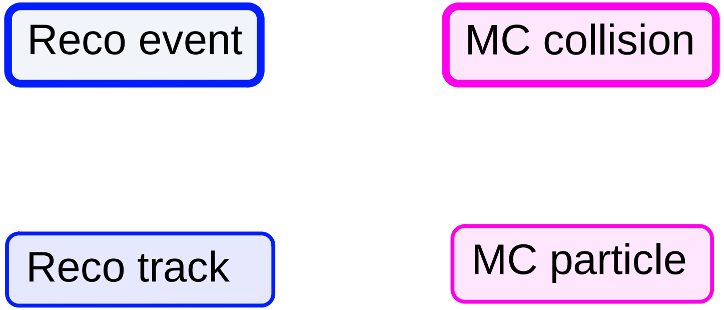
- The connection between the reconstructed-level tables and the MC truth tables is done via label tables, similar to the O2 framework

DQ derived data model: muon labels



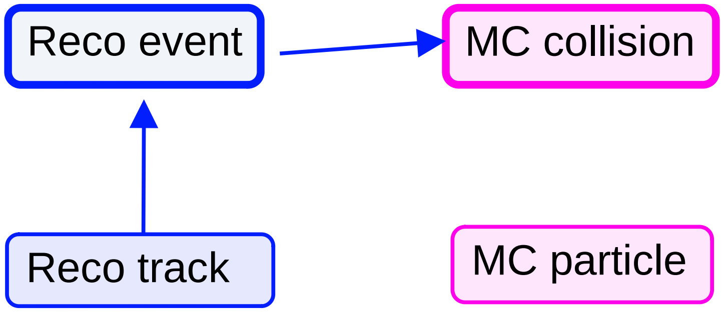
- The connection between the reconstructed-level tables and the MC truth tables is done via label tables, similar to the O2 framework

Reconstructed vs MC truth: “ideal case”



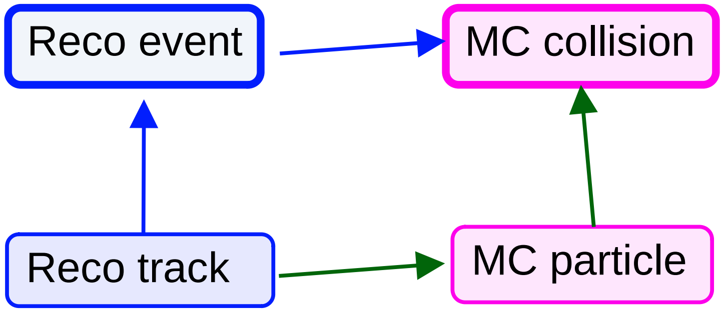
- Ideally, MC truth information should not be ambiguous, e.g.:
  - Retrieving the MC collision, starting from a reconstructed track

Reconstructed vs MC truth: “ideal case”



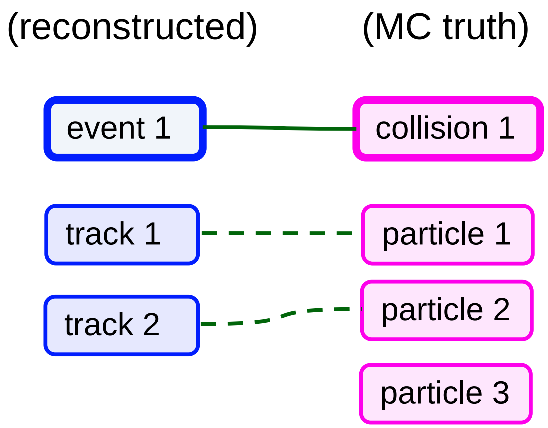
- A reconstructed track belongs to a corresponding reconstructed event, which in turn has a matched MC collision

Reconstructed vs MC truth: “ideal case”



- A reconstructed track has a matched MC particles, which in turn belongs to a MC collision
- Ideally, the blue and green paths would lead to the same MC collision

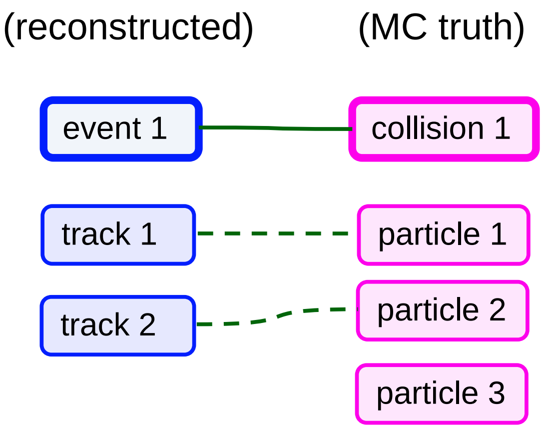
Reconstructed vs MC truth (a couple of examples)



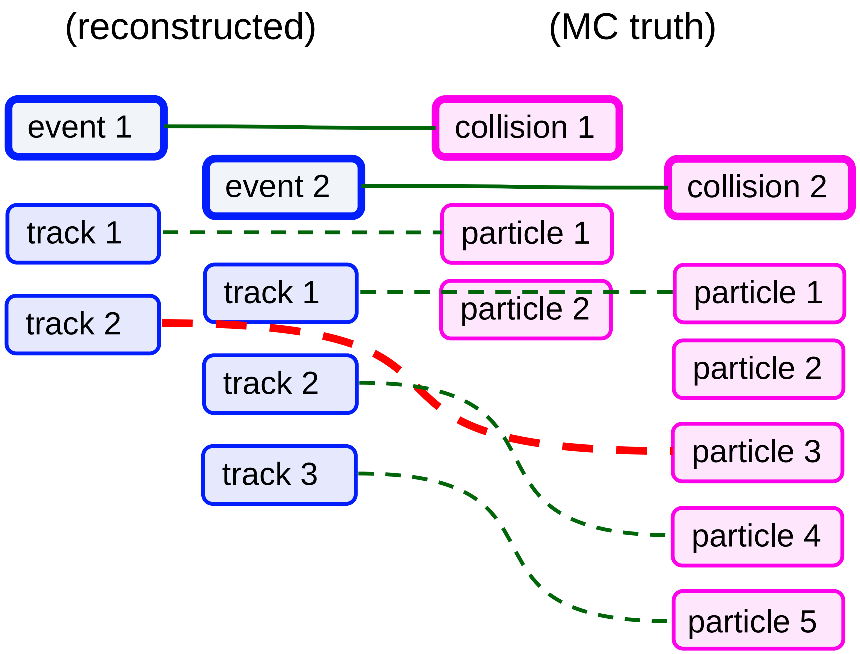
"Standard"/ideal case

- Standard case:
  - reconstructed event matched to an MC collision
  - Each reconstructed track is matched to an MC particle from the same MC collision as the reconstructed event match

Reconstructed vs MC truth (a couple of examples)



“Standard”/ideal case



Wrong track-collision association

- Wrong track-collision association
  - Reconstructed events correctly matched to MC collisions
  - One track from event 1 is matched to an MC particle from MC collision 2
    - Possible explanation: the two collisions are close in time, and one of the tracks has a large time uncertainty, so in reconstruction was assigned to the wrong reco event

Skimming workflow: input data

Input data
& params

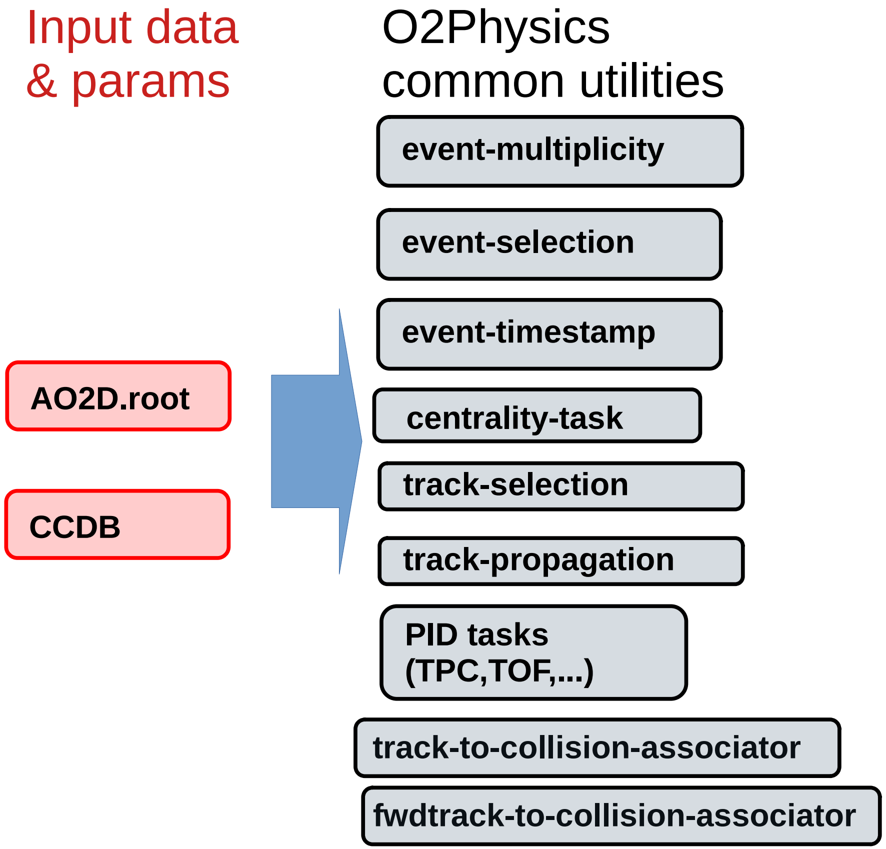
AO2D.root

CCDB

- Produces filtered and skimmed data
- Configurability for
  - selecting events, tracks, muons and MFT tracklets
  - amount of event/track/muon/MFT information
- Multiple parallel selections on tracks and muons (similar to AOD filter bits)
- Intended for general purpose analysis, not limited to DQ

- Input data: AO2D and CCDB

Skimming workflow: common utilities



- Aggregates on-the-fly information produced by the O2 common utilities
  - Event selection
  - Multiplicity, centrality
  - Timestamp
  - Track selection and propagation
  - PID
  - Track to collision associations

Skimming workflow: DQ utilities

Input data
& params

AO2D.root

CCDB

O2Physics
common utilities

event-multiplicity

event-selection

event-timestamp

centrality-task

track-selection

track-propagation

PID tasks (TPC,TOF,...)

track-to-collision-associator

fwdtrack-to-collision-associator

DQ framework

Filter-pp, filter-PbPb
Produces:
- *EventFiltering::DqFilters,*
- *DQ::DQEventFilter*
- *QA histograms*

v0-selector
Produces:
- *DQ::V0Bits*
- *QA histograms*

dalitz-selector
Produces:
- *DQ::DalitzBits*
- *QA histograms*

dq-flow
Produces:
- *DQ::Qvector*
- *QA histograms*

- DQ utilities
  - Event filtering
  - V0 and Dalitz tagging
  - Q-vector and correlations

Skimming workflow: `table-maker`

Input data & params

- **AO2D.root**
- **CCDB**

O2Physics common utilities

- **event-multiplicity**
- **event-selection**
- **event-timestamp**
- **centrality-task**
- **track-selection**
- **track-propagation**
- **PID tasks (TPC,TOF,...)**
- **track-to-collision-associator**
- **fwdtrack-to-collision-associator**

DQ framework

filter-pp
Produces:
- *EventFiltering::DqFilters,*
- *DQ::DQEventFilter*
- *QA histograms*

v0-selector
Produces:
- *DQ::V0Bits*
- *QA histograms*

dalitz-selector
Produces:
- *DQ::DalitzBits*
- *QA histograms*

dq-flow
Produces:
- *DQ::Qvector*
- *QA histograms*

table-maker
Produces:
- *Skimmed data tables*
- *Event stats*
- *Track & muon stats*
- *QA histograms*

- Produces a standalone derived data model

The analysis workflow: input data

Input data

DQ skims

or

AO2D.root

- Configurable workflow to run all analyses
- Uses DQ skims, but can be configured also Framework/AO2D
- Can replay event/track selections but also uses precomputed decisions from skimming time
- Produces higher level skims for “offline” (e.g. machine learning) applications
  - dileptons, triplets, etc.

The analysis workflow: event and track selection

Input data

DQ skims

or

AO2D.root

event-selection
Produces:
- *Selections*
- *Ev. mixing categories*
- *QA histograms*

track-selection(s)
Produces:
- *Selections*
- *QA histograms*

prefilter-selection(s)
Produces:
- *Selections*

muon-selection(s)
Produces:
- *Selections*
- *QA histograms*

- Event and track selection devices (analysis tasks)
  - Produce event/track selection bits
  - Event mixing categories
  - QA histograms

The analysis workflow: two-track combinatorics

Input data

DQ skims

or

AO2D.root

event-selection
Produces:
- *Selections*
- *Ev. mixing categories*
- *QA histograms*

track-selection(s)
Produces:
- *Selections*
- *QA histograms*

prefilter-selection(s)
Produces:
- *Selections*

muon-selection(s)
Produces:
- *Selections*
- *QA histograms*

Mixed-event-pairing
Produces:
- *Histograms*

Same-event pairing
Produces:
- *Histograms*
- *Dilepton tables*

asymmetric pairing
Produces:
- *Histograms*
- *Di-track tables*

High-level objects on disk

- Same event pairing
- Mixed event pairing
- Asymmetric pairing
- Produces di-lepton and di-track tables

The analysis workflow: higher level tasks

Input data

DQ skims

or

AO2D.root

event-selection
Produces:
- *Selections*
- *Ev. mixing categories*
- *QA histograms*

track-selection(s)
Produces:
- *Selections*
- *QA histograms*

prefilter-selection(s)
Produces:
- *Selections*

muon-selection(s)
Produces:
- *Selections*
- *QA histograms*

Mixed-event-pairing
Produces:
- *Histograms*

Same-event pairing
Produces:
- *Histograms*
- *Dilepton tables*

asymmetric pairing
Produces:
- *Histograms*
- *Di-track tables*

High-level objects on disk

Dilepton-track pairing
Produces:
- *Histograms*
- *Dilepton-track tables*

Dilepton-track-track
Produces:
- *Histograms*
- *Candidate tables*

High-level objects on disk

- High level analysis tasks
  - Dilepton + track: $B \rightarrow J/\psi + K$, J/ψ-hadron correlations
  - Dilepton + track + track: $\psi(2S)/X(3872) \rightarrow J/\psi+\pi\pi$
  - Ongoing: $\chi_c \rightarrow J/\psi + \gamma$, J/ψ in jets

O2 DQ analysis tools in ~20 lines of code

```
init() {
  histMan = new HistogramManager("analysisHistos", "Analysis histograms", VarManager::kNVars);
  dqhistograms::DefineHistograms(histMan, "EventHistograms", "event", "trigger,cent,mc");
  dqhistograms::DefineHistograms(histMan, "TrackHistograms", "track", "its,tpcpid,dca,tofpid,mc");
  dqhistograms::DefineHistograms(histMan, "TrackHistograms_MCmatch", "track", "its,tpcpid,dca,tofpid,mc");
}
```

} Define histograms

- Predefined histograms are loaded from the HistogramsLibrary for
  - Class ***event*** and sub-classes ***trigger,cent,mc***
  - Class ***track*** and sub-classes ***its,tpcpid,dca,tofpid,mc***
- Histograms will be available in the output file in the directories "*EventHistograms*", "*TrackHistograms*" and "*TrackHistograms_Mcmatch*"

- Histograms are "aware" of what information they require
  - Filling of histograms can be done "in bulk"
  - No call to each individual histogram in the analysis task
    - neat and compact code
    - Decreases the frequency of PRs which modify just the binning of a histogram and allows a task to be used by multiple users for different purposes

O2 DQ analysis tools in ~20 lines of code

```
init() {
  histMan = new HistogramManager("analysisHistos", "Analysis histograms", VarManager::kNVars);
  dqhistograms::DefineHistograms(histMan, “EventHistograms”, "event", "trigger,cent,mc");
  dqhistograms::DefineHistograms(histMan, “TrackHistograms”, "track", "its,tpcpid,dca,tofpid,mc");
  dqhistograms::DefineHistograms(histMan, “TrackHistograms_MCmatch”, "track", "its,tpcpid,dca,tofpid,mc");

  fEventCut = new AnalysisCompositeCut(true);
  fEventCut→AddCut(dqcuts::GetAnalysisCut(“eventStandard”));
  fTrackCut = new AnalysisCompositeCut(true);
  fTrackCut→AddCut(dqcuts::GetAnalysisCut(“jpsiPID1”));
}
```

Define histograms

Define cuts

- An event cut (***eventStandard***) and a track cut (***jpsiPID1***) loaded from the CutsLibrary are being initialized
- Define cuts in a library with “human readable names”
  - Easy usage as input parameters (configurables) of the task
  - Flexibility: select fixed or variable (TF1) ranges for VarManager defined variables
  - Can be combined with AND / OR logic to construct complicated selections
  - Sharing between analysers
  - Reproducibility

O2 DQ analysis tools in ~20 lines of code

```
init() {
  histMan = new HistogramManager("analysisHistos", "Analysis histograms", VarManager::kNVars);
  dqhistograms::DefineHistograms(histMan, “EventHistograms”, "event", "trigger,cent,mc");
  dqhistograms::DefineHistograms(histMan, “TrackHistograms”, "track", "its,tpcpid,dca,tofpid,mc");
  dqhistograms::DefineHistograms(histMan, “TrackHistograms_MCmatch”, "track", "its,tpcpid,dca,tofpid,mc");

  fEventCut = new AnalysisCompositeCut(true);
  fEventCut→AddCut(dqcuts::GetAnalysisCut(“eventStandard”));
  fTrackCut = new AnalysisCompositeCut(true);
  fTrackCut→AddCut(dqcuts::GetAnalysisCut(“jpsiPID1”));

  fMCsignal = dqmcsignals::GetMCSignal(“eFromJpsi”);
}
```

} Define histograms

} Define cuts

} Define MC signals

- A Monte-Carlo signal which will be used for matching is loaded from the MCSignalLibrary, e.g. ***eFromJpsi***
- Can be used as input parameters of the task
- Stored in a library shared with all analyzers
- Flexibility that covers a very wide range of use cases

O2 DQ analysis tools in ~20 lines of code

```
process(aod::ReducedEvent event, soa::Join<aod::ReducedTracks, aod::ReducedTracksBarrelLabels> tracks,
aod::ReducedMCTracks tracksMC) {
  VarManager::FillEvent<gkEventFillMap>(event);
  if (!fEventCut->IsSelected(VarManager::fgValues)) {
      return;
  }
  histMan→FillHistClass("EventHistograms", VarManager::fgValues);
}
```

Compute event quantities, apply selections & fill histograms

- Process function subscribing to the DQ skimmed collision, tracks and MC particles information
- Apply event selection and fill event histograms:
  - Compute all relevant event-wise quantities: *VarManager::FillEvent()*
  - Check that the event fulfills the selection cut: *fEventCut->IsSelected()*
  - Use the histogram manager to fill all the event wise histograms: *histMan->FillHistClass()*

O2 DQ analysis tools in ~20 lines of code

```
process(aod::ReducedEvent event, soa::Join<aod::ReducedTracks, aod::ReducedTracksBarrelLabels> tracks,
aod::ReducedMCTracks tracksMC) {
  VarManager::FillEvent<gkEventFillMap>(event);
  if (!fEventCut->IsSelected(VarManager::fgValues)) {
      return;
  }
  histMan→FillHistClass(“EventHistograms”, VarManager::fgValues);

  for(auto& track : tracks) {
    VarManager::FillTrack<gkTrackFillMap>(track); // compute track quantities
    VarManager::FillTrack<gkParticleMCFillMap>(track.reducedMCTrack());
    if( ! fTrackCut->IsSelected(VarManager::fgValues))
        continue;
    histMan→FillHistClass(“TrackHistograms”, VarManager::fgValues);
  }
}
```

} Compute event quantities, apply selections & fill histograms

} Compute track quantities, apply selections & fill histograms

- Loop over tracks
- Compute all relevant quantities: *VarManager::FillTrack()*
- Check that the track fulfills the selection cut: *fTrackCut→IsSelected()*
- Fill the histograms from the “TrackHistograms”: *histMan->FillHistClass()*

O2 DQ analysis tools in ~20 lines of code

```
process(aod::ReducedEvent event, soa::Join<aod::ReducedTracks, aod::ReducedTracksBarrelLabels> tracks,
aod::ReducedMCTracks tracksMC) {
  VarManager::FillEvent<gkEventFillMap>(event);
  if (!fEventCut->IsSelected(VarManager::fgValues)) {
     return;
  }
  histMan→FillHistClass("EventHistograms", VarManager::fgValues);

  for(auto& track : tracks) {
    VarManager::FillTrack<gkTrackFillMap>(track); // compute track quantities
    VarManager::FillTrack<gkParticleMCFillMap>(track.reducedMCTrack());
    if( ! fTrackCut->IsSelected(VarManager::fgValues))
       continue;
    histMan→FillHistClass("TrackHistograms", VarManager::fgValues);

    if(fMCsignal→CheckSignal(true,tracksMC,track.reducedMCTrack())) {
      histMan→FillHistClass("TrackHistograms_MCmatch", VarManager::fgValues);
    }
  }
}
```

Compute event quantities, apply selections & fill histograms

Compute track quantities, apply selections & fill histograms

Match MC signals & fill histograms

- Check that the current track matches the specified MC signal: *fMCSignal→CheckSignal()*
- Fill histograms for the tracks which match the specified signal

Summary

- A nearly complete set of analysis tools is available for single-lepton, di-lepton and higher level analyses (either in the barrel, muon or barrel-muon)
- Level of configurability allows for the same workflow to be used in a variety of analyses
- Ongoing debugging, understanding Run-3 data, implementing additional tools

Backup