Status of Delphes Integration in FCCSW

Zbyněk Drásal

CERN

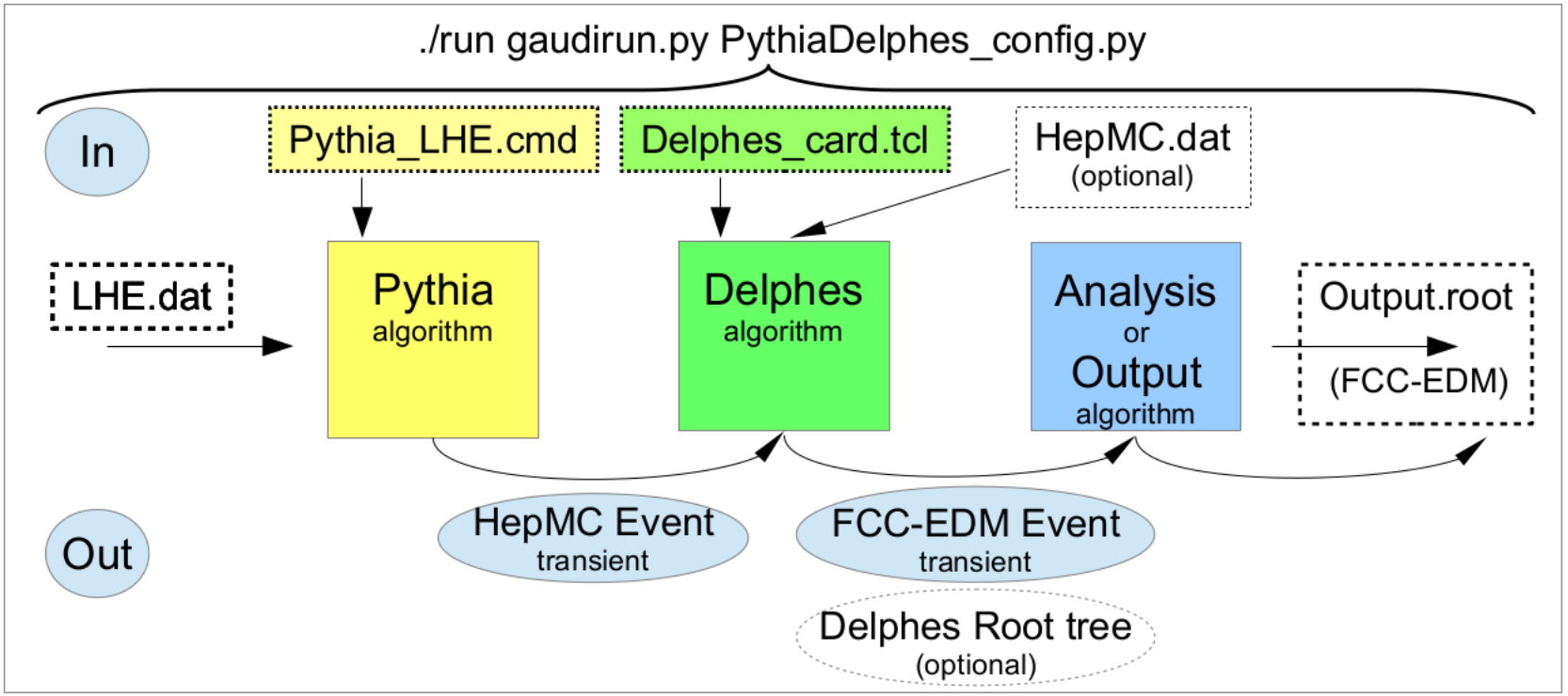
With: C. Helsens

Overview

- **Delphes within FCCSW**
  - Strategy & processing sequence
  - Input/Output using FCC-EDM (Event-Data model)
  - Configuration files
  - Documentation
- **Delphes card details**
  - Delphes 3.3 version & recommended FCC card versus official FCC-hh Delphes card

Strategy & Processing Sequence

- FCCSW → based on modular structure of Gaudi framework
- Strategy:
  - modularize Pythia & Delphes as Gaudi algorithms
  - use PythiaDelphes_config.py - FCCSW command file (python script) to run Pythia & Delphes → set all GAUDI parameters to run Pythia, Delphes & Output module



Details of FCCSW Config File

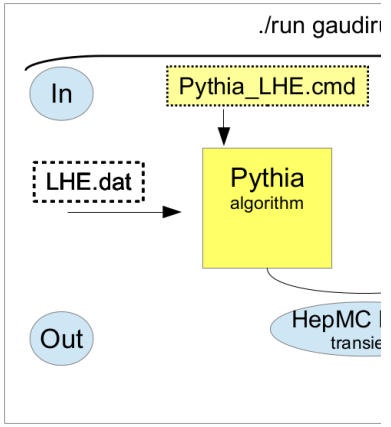
- **PythiaDelphes_config.py**:
  - Defines a run sequence of Gaudi modules (Pythia+Delphes) through a Python script
  - Variables to be arranged:
    - `nEvents` → Events to be simulated
    - `messageLevel` --> GAUDI messaging verbosity: ERROR, WARNING, INFO, DEBUG
    - `pythiaconfFile` → Pythia config file: `Pythia_LHEinput.cmd/Pythia_standard.cmd`
    - `delphesCard` → Delphes TCL configuration file (use official Delphes card)
    - `delphesHepMCInFile` → Delphes input file (use "" to read HepMC directly from Pythia module, i.e. from transient data store)
    - `delphesRootOutFile` → Delphes output file (use "" to output data to transient data store → FCC-EDM objects automatically written out through Gaudi `out` module!
    - `delphes???OutArray` → Define which Delphes module objects are processed as FCC-EDM (Event Data Model) particles, where `???` stands for `muons, electrons, photons` …

Pythia Input/Output & Configuration

- **Pythia Data Input:**

  1) **LHE (Les Houches Event)** data file (from e.g. Madgraph, ...)
     → Already simulated Les Houches Events are processed by Pythia
     → Pythia performs MPI, ISR, FSR, hadronization, decays …

  2) No input, **Pythia simulated events** directly used
     → Pythia simulates physics events & performs MPI, ISR, FSR, …

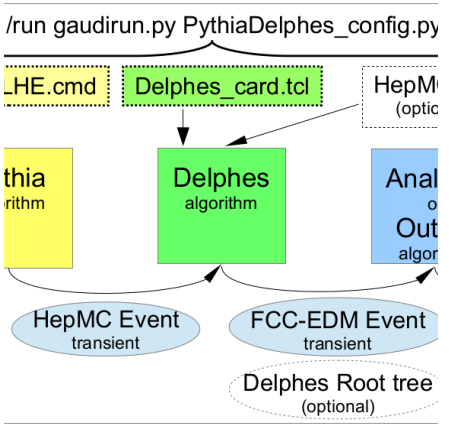


- **Pythia Data Output:**
  - Data output through transient memory data store using **HepMC event data format**

- **Pythia Configuration Files – 2 use cases prepared:**
  - **Pythia_LHEinput.cmd**: use Pythia-module to read-in the LHE file (generated by Madgraph, etc.), set Pythia run parameters
  - **Pythia_standard.cmd**: use Pythia-module to simulate physics events directly, define physics process to be simulated + Pythia run parameters

Delphes Input/Output & Configuration

- **Delphes Data Input:**
  - The Delphes data input is through the transient event data store in HepMC format (in principal, HepMC data file can be read-in too) → input read-in by `DelphesExtHepMCReader`
- **Delphes Output:**
  - The Delphes output objects (specified by `delphes???OutArray` variables in PythiaDelphes_config.py) are written out to FCC-EDM collections & associated collections (one-to-one relations)



- **`delphes???OutArrays`** (may be modified by user!):

```
delphesMuonsOutArray    ="MuonIsolation/muons"
delphesElectronsOutArray="ElectronIsolation/electrons"
delphesChargedOutArray  ="ChargedHadronMomentumSmearing/chargedHadrons"
delphesNeutralOutArray  ="Hcal/eflowNeutralHadrons"
delphesPhotonsOutArray  ="PhotonIsolation/photons"
delphesJetsOutArray     ="JetEnergyScale/jets"
delphesMETsOutArray     ="MissingET/momentum"
delphesSHTsOutArray     ="ScalarHT/energy"
```

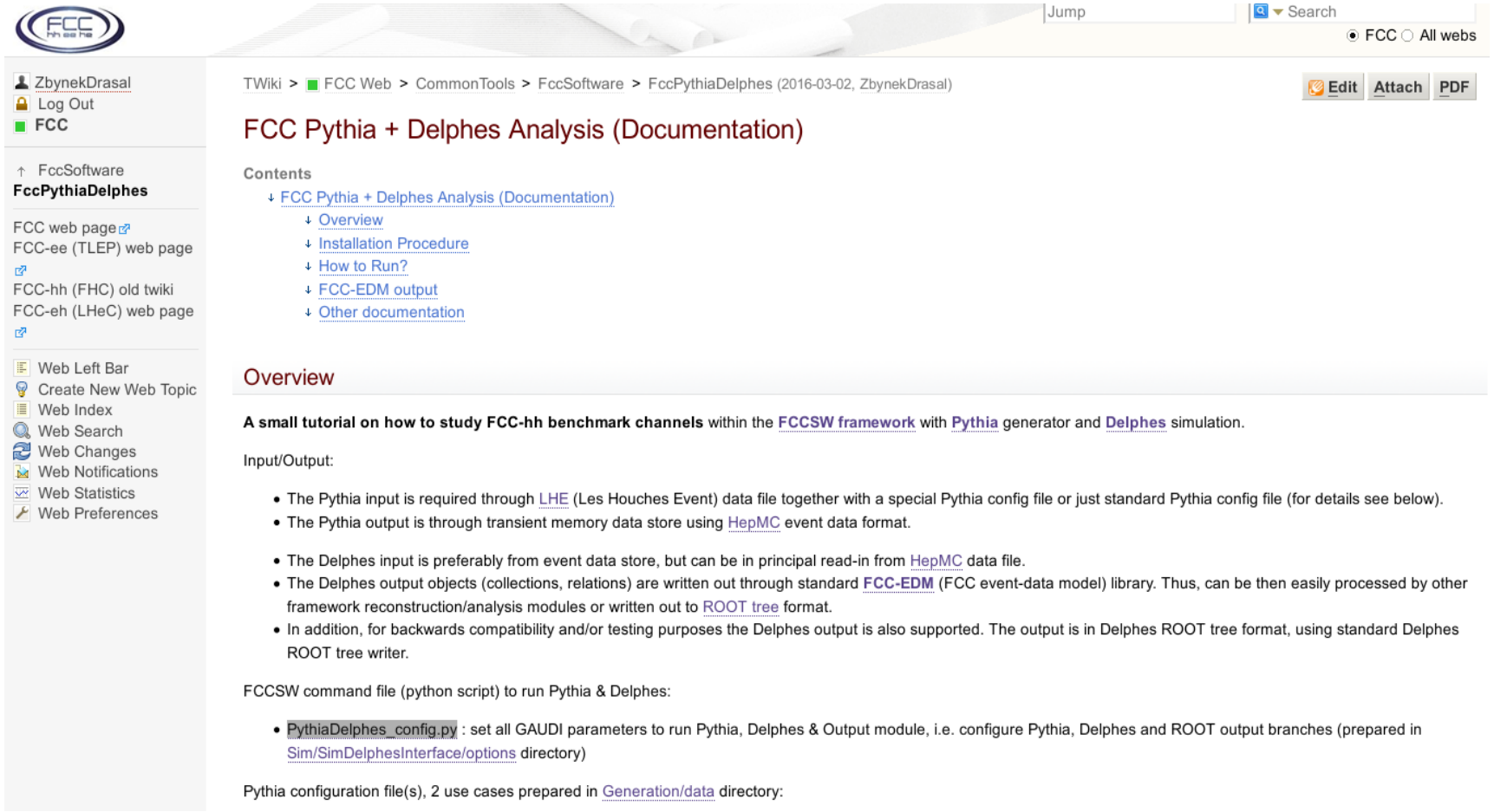
FCC-EDM Output

- **Collections:**
  - **`fcc::MCParticleCollection`** --> generated particles `[genParticles]`
  - **`fcc::GenVertexCollection`** --> generated vertices `[genVertices]`
  - **`fcc::GenJetCollection`** --> generated jets `[genJets]`
  - **`fcc::ParticleCollection`** --> reconstructed muons `[muons]`, electrons `[electrons]`, charged particles `[charged]`, neutral particles `[neutral]`, photons `[photons]` and jet constituents `[jetParts]`
  - **`fcc::JetCollection`** --> reconstructed jets `[jets]`
  - **`fcc::METCollection`** --> reconstructed missing Et `[met]`
  - **`fcc::IntTagCollection`** --> flavour tag of generated `[genJetsFlavor]` or reconstructed jets `[jetsFlavor]`, i.e. PDG of leading constituent
  - **`fcc::TagCollection`** --> reconstructed tags - b-tags, tau-tags for jets `[bTags, tauTags]`
  - **`fcc::TagCollection`** --> reconstructed isolation tag info for electrons, muons and photons `[muonITags, electronITags, photonITags]`

- **Relations:**
  - **`fcc::ParticleMCParticleAssociationCollection`** --> relations of reconstructed object to MC particle for muons `[muonsToMC]`, electrons `[electronsToMC]`, charged particles `[chargedToMC]`, neutral particles `[neutralToMC]` and photons `[photonsToMC]`
  - **`fcc::ParticleTagAssociationCollection`** --> relations of reconstructed particles: muons, electrons and photons to their isolation tag info `[muonsToITags, electronsToITags, photonsToITags]`
  - **`fcc::GenJetParticleAssociationCollection`** --> relations of generated jet to MC particle `[genJetsToMC]`
  - **`fcc::GenJetIntTagAssociationCollection`** --> relations of generated jets to the flavour (PDG of the leading constituent) `[genJetsToFlavor]`
  - **`fcc::JetParticleAssociationCollection`** --> relations of jet to particle constituents `[jetsToParts]`
  - **`fcc::JetIntTagAssociationCollection`** --> relations of jets to the flavour (PDG of the leading constituent) `[jetsToFlavor]`
  - **`fcc::JetTagAssociationCollection`** --> relations of jets to reconstructed tags - b-tag, tau-tag `[jetsToBTags, jetsToTauTags]`

Need more Information?

- For additional information follow the HowTo? at FCC Twiki page:
  - **https://twiki.cern.ch/twiki/bin/view/FCC/FccPythiaDelphes**



Jump Search FCC All webs

ZbynekDrasal
Log Out
FCC

FccSoftware
FccPythiaDelphes

FCC web page
FCC-ee (TLEP) web page
FCC-hh (FHC) old twiki
FCC-eh (LHeC) web page

Web Left Bar
Create New Web Topic
Web Index
Web Search
Web Changes
Web Notifications
Web Statistics
Web Preferences

TWiki > FCC Web > CommonTools > FccSoftware > FccPythiaDelphes (2016-03-02, ZbynekDrasal)

Edit Attach PDF

FCC Pythia + Delphes Analysis (Documentation)

Contents

- FCC Pythia + Delphes Analysis (Documentation)
  - Overview
  - Installation Procedure
  - How to Run?
  - FCC-EDM output
  - Other documentation

Overview

A small tutorial on how to study FCC-hh benchmark channels within the FCCSW framework with Pythia generator and Delphes simulation.

Input/Output:

- The Pythia input is required through LHE (Les Houches Event) data file together with a special Pythia config file or just standard Pythia config file (for details see below).
- The Pythia output is through transient memory data store using HepMC event data format.

- The Delphes input is preferably from event data store, but can be in principal read-in from HepMC data file.
- The Delphes output objects (collections, relations) are written out through standard FCC-EDM (FCC event-data model) library. Thus, can be then easily processed by other framework reconstruction/analysis modules or written out to ROOT tree format.
- In addition, for backwards compatibility and/or testing purposes the Delphes output is also supported. The output is in Delphes ROOT tree format, using standard Delphes ROOT tree writer.

FCCSW command file (python script) to run Pythia & Delphes:

- PythiaDelphes_config.py : set all GAUDI parameters to run Pythia, Delphes & Output module, i.e. configure Pythia, Delphes and ROOT output branches (prepared in Sim/SimDelphesInterface/options directory)

Pythia configuration file(s), 2 use cases prepared in Generation/data directory:

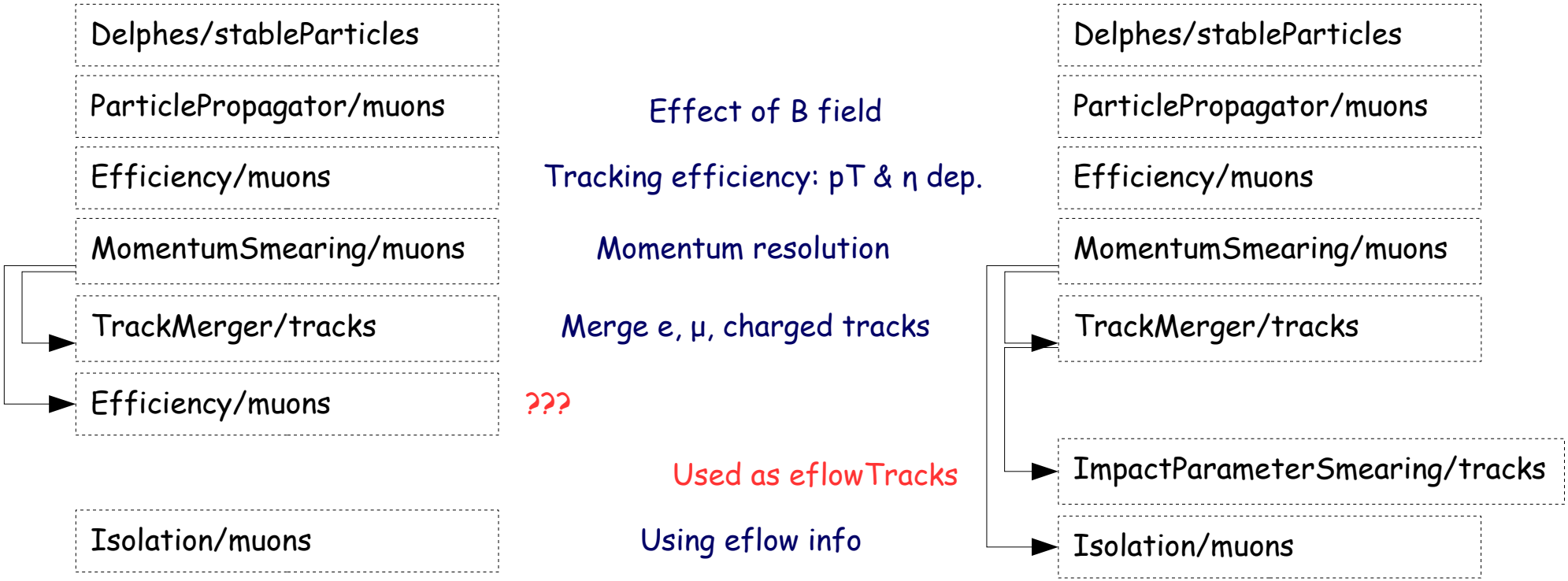
FCC Delphes Card – Flow Chart

- **Summary of Delphes flow chart** for individual particle objects
- Comparison of Delphes group recommended FCC card
  versus FCC-hh official card:
  - `FCC_Delphes.card` → Delphes card for Delphes version 3.3.2
  - `FCC-hh_official.card` → **current official FCC-hh Delphes card (done for Delphes 3.2)**

  → **want to address several "differences"**, mainly due to new version of Delphes used (3.3 versus 3.2)

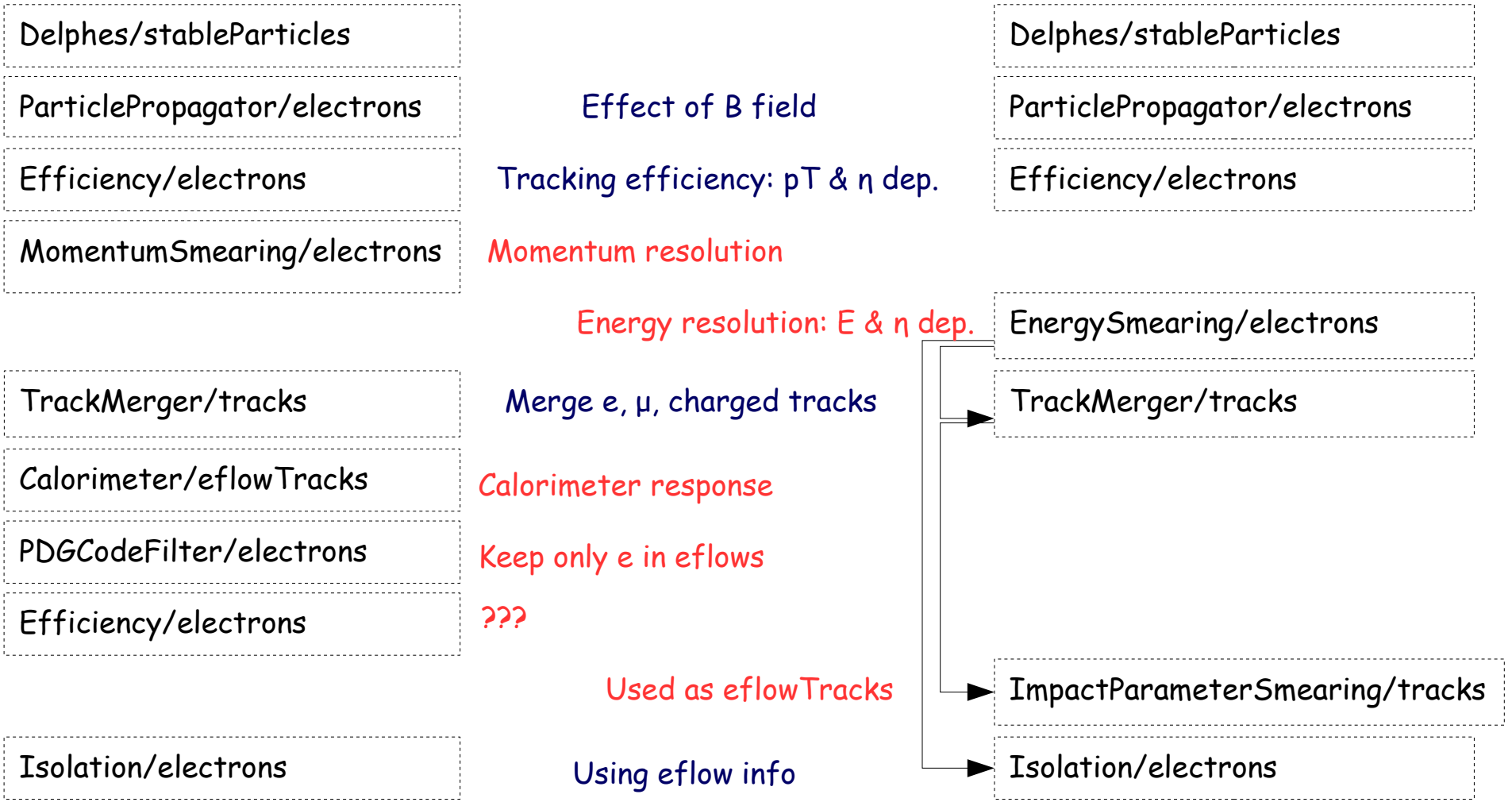
Flow Chart: Muons

- **Delphes x FCC-hh:** DelphesModule/objectName



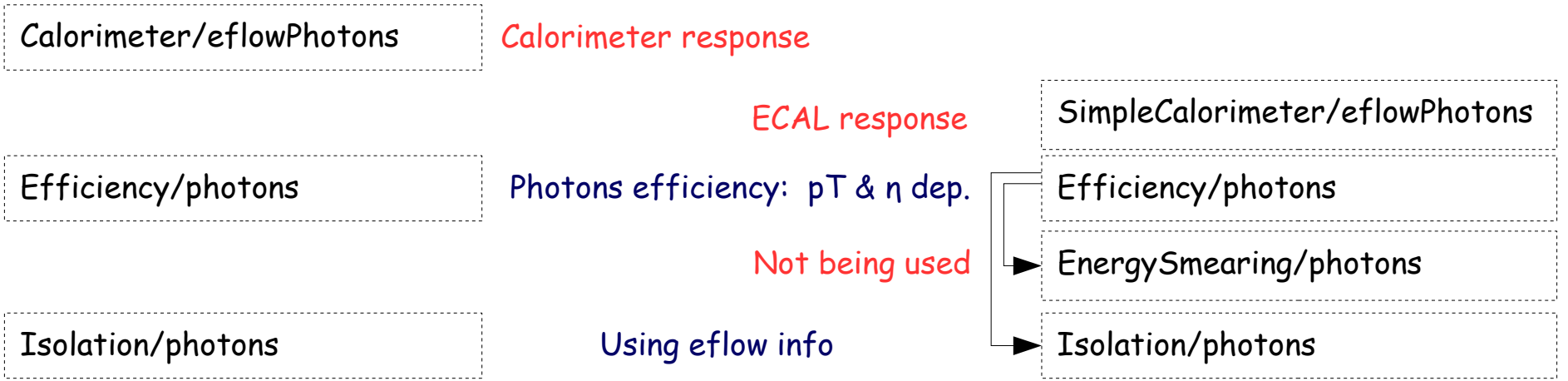
→ **OK?**

Flow Chart: Electrons



→ ???

Flow Chart: Photons



→ **OK**

Flow Chart: Charged Hadrons

Delphes/stableParticles		Delphes/stableParticles
ParticlePropagator/chargedHadrons	Effect of B field	ParticlePropagator/chargedHadrons
Efficiency/chargedHadrons	Tracking efficiency: pT & η	Efficiency/chargedHadrons
MomentumSmearing/chargedHadrons	Momentum resolution	MomentumSmearing/chargedHadrons
TrackMerger/tracks	Merge e, μ, charged tracks	TrackMerger/tracks

→ **OK**

Flow Chart: GenJets

Delphes/stableParticles		Delphes/stableParticles
PDGCodeFilter/filteredParticles	Filter out (anti)neutrinos	
FastJetFinder/jets	Find jets	FastJetFinder/jets

→ **OK? (needs to add neutrino filter)**

Flow Chart: Jets

Calorimeter/eflowTracks Calorimeter/eflowPhotons Calorimeter/eflowNeutralHadrons	Calorimeter	ImpactParameterSmearing/eflowTracks SimpleCalorimeter/flowPhotons (ECAL) SimpleCalorimeter/eflowNeutralHadrons
Merger/eflow	Merge eflow	Merger/eflow
FastJetFinder/jets	Find jets	FastJetFinder/jets
EnergyScale/jets	Jet energy scale	EnergyScale/jets
JetFlavorAssociation	PDG of leading constituent	
BTagging	b-tagging	Btagging → b-tag
	b-tagging	Btagging → c-tag
TauTagging	tau-tagging	TauTagging

→ **OK? (missing flavor tag)**

Summary & Outlook

- **Summary – Delphes integration:**
  - Pythia & Delphes implemented as Gaudi algorithms into FCCSW framework with standard FCC-EDM output (using FCC-EDM & PODIO library)
  - For instructions on how to analyze the output, follow the talk by C.Helsens
  - Further information & detailed documentation can be found on FCCSW Twiki: `https://twiki.cern.ch/twiki/bin/view/FCC/FccPythiaDelphes`
- **Summary – Delphes card:**
  - FCCSW uses a new 3.3 version of Delphes
  - Addressing the differences between the recommended Delphes modules flow for FCC by Delphes collaboration and the current official FCC-hh Delphes card → **the official card needs to be updated in terms of modules flow for electrons, genJets & jets**