CMSSW의 소개

Jason Lee
University of Seoul

KCMS Lectures on Collider Physics
Season 1

What is CMSSW

https://github.com/cms-sw/cmssw

https://cms-opendata-workshop.github.io/workshop2021-lesson-cmssw/index.html

The CMS Software (CMSSW) is a collection of software libraries that the CMS experiment uses in order to acquire, produce, process and even analyze its data. The program is written in C++ but its configuration is manipulated using the Python language.

CMSSW is built around a Framework, an Event Data Model (EDM), and Services needed by the simulation, calibration and alignment, and reconstruction modules that process event data so that physicists can perform analysis. The primary goal of the Framework and EDM is to facilitate the development and deployment of reconstruction and analysis software.

workflow of CMSSW

1. Event Generation
2. Detector Simulation
3. Digitisation - Detector Response
4. Level 1 Trigger
5. Digi2Raw - packing data into DAQ format
6. High Level Trigger
7. Raw2Digi - unpacking DAQ data
8. Reconstruction
    1. Local Reconstruction
    2. Global Reconstruction
    3. Particle Identification
    4. Physics Analysis Tools

Setting up CMSSW

```
# log into lxplus or local tier 3
# setting up CMSSW environment
source /cvmfs/cms.cern.ch/cmsset_default.sh
# setting up a new CMSSW release
cmsrel CMSSW_12_1_0

cd CMSSW_12_1_0/src
# setting up the environment for the release
cmsenv
# if doing developement for CMSSW
git-cms-init

# list full simulation workflows
runTheMatrix.py -n -e -w upgrade

# running a simulation workflow - ttbar event
runTheMatrix.py -l 10024.0

# go into the workflow directory
cd
10024.0_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVES
TFakeHLT+ALCA+Nano
```

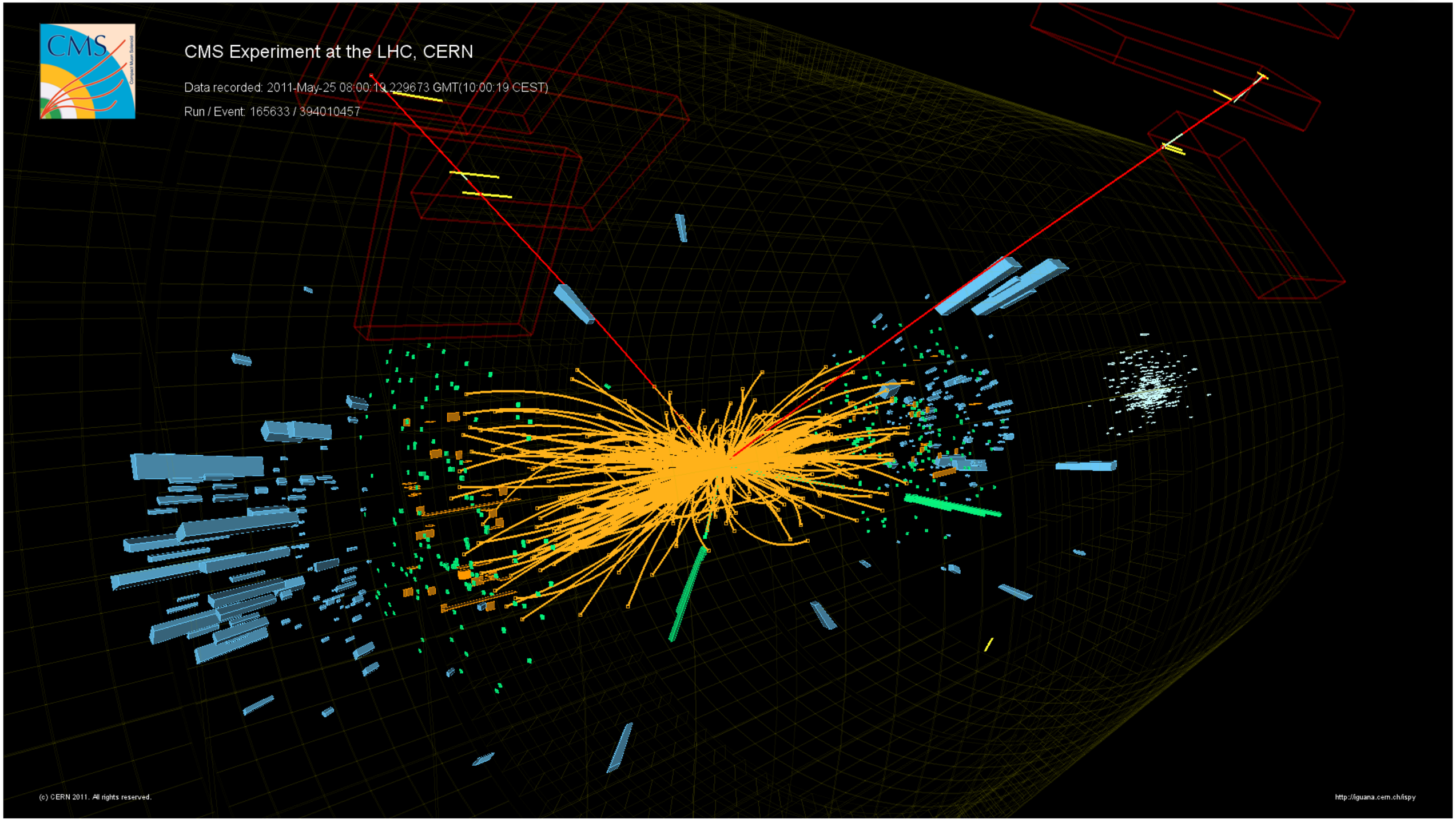
python driver

```
vi TTbar_13TeV_TuneCUETP8M1_cfi_GEN_SIM.py

# import cms module
import FWCore.ParameterSet.Config as cms

# Eras control the detector scenario
# phase2 cms has new detectors
from Configuration.Eras.Era_Run2_2017_cff import Run2_2017

# the Process
process = cms.Process('SIM',Run2_2017)
```

Event Generation

```
vi TTbar_13TeV_TuneCUETP8M1_cfi_GEN_SIM.py

process.generator = cms.EDFilter("Pythia8ConcurrentGeneratorFilter",
    PythiaParameters = cms.PSet(
        parameterSets = cms.vstring(
            'pythia8CommonSettings',
            'pythia8CUEP8M1Settings',
            'processParameters'
        ),
        processParameters = cms.vstring(
            'Top:gg2ttbar = on ',
            'Top:qqbar2ttbar = on ',
            '6:m0 = 175 '
        ),
        pythia8CUEP8M1Settings = cms.vstring(
            'Tune:pp 14',
            'Tune:ee 7',
            'MultipartonInteractions:pT0Ref=2.4024',
            'MultipartonInteractions:ecmPow=0.25208',
            'MultipartonInteractions:expPow=1.6'
        ),
        pythia8CommonSettings = cms.vstring(
            'Tune:preferLHAPDF = 2',
            'Main:timesAllowErrors = 10000',
            'Check:epTolErr = 0.01',
            'Beams:setProductionScalesFromLHEF = off',
            'SLHA:minMassSM = 1000.',
            'ParticleDecays:limitTau0 = on',
            'ParticleDecays:tau0Max = 10',
            'ParticleDecays:allowPhotonRadiation = on'
        )
    ),
    comEnergy = cms.double(13000.0),
    filterEfficiency = cms.untracked.double(1.0),
    maxEventsToPrint = cms.untracked.int32(0),
    pythiaHepMCVerbosity = cms.untracked.bool(False),
    pythiaPylistVerbosity = cms.untracked.int32(0)
)
```

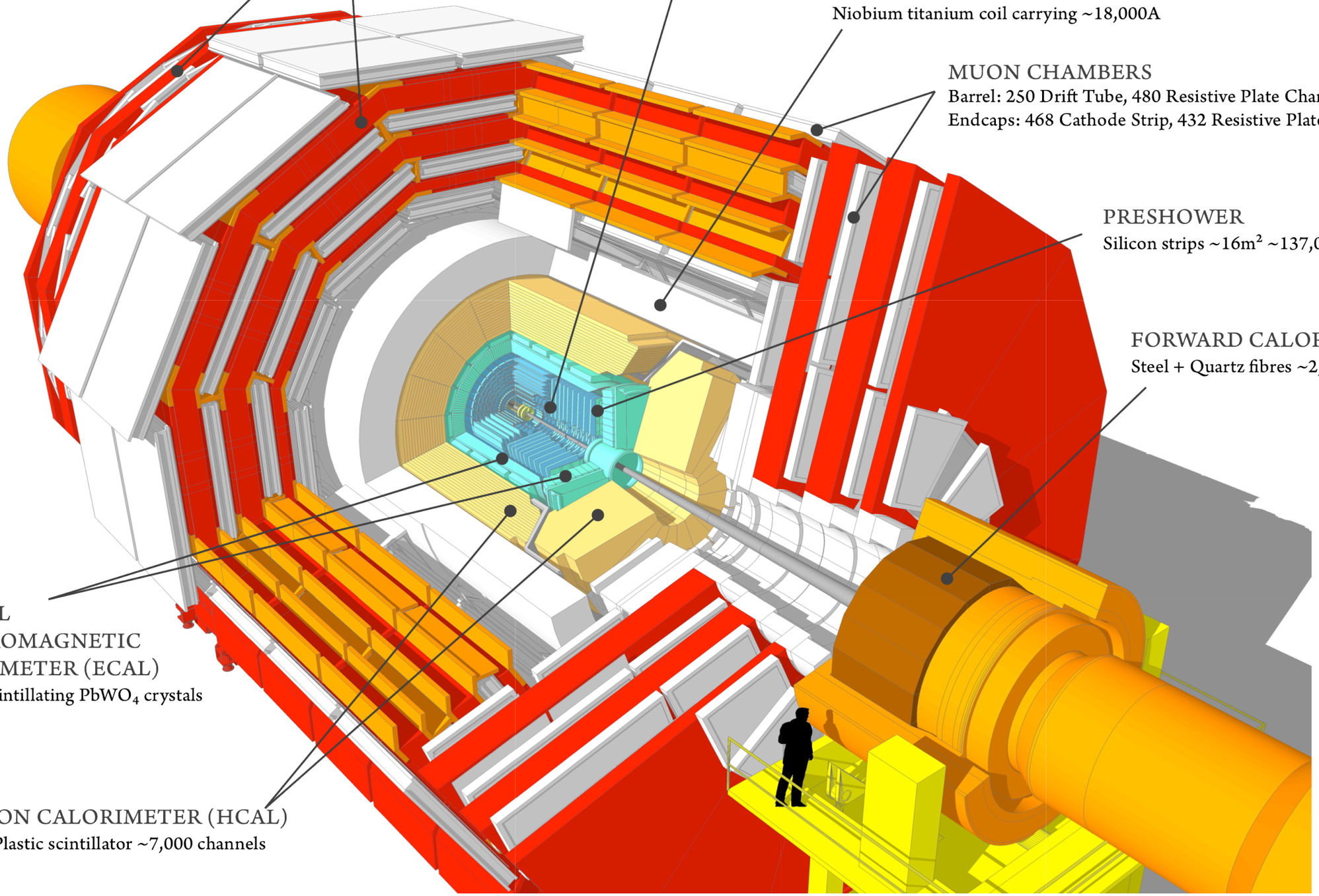
Event Generation



CMS experiment

CMS DETECTOR

Total weight	: 14,000 tonnes
Overall diameter	: 15.0 m
Overall length	: 28.7 m
Magnetic field	: 3.8 T



STEEL RETURN YOKE
12,500 tonnes

SILICON TRACKERS
Pixel (100x150 μm) ~16m² ~66M channels
Microstrips (80x180 μm) ~200m² ~9.6M channels

SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000A

MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 468 Cathode Strip, 432 Resistive Plate Chambers

PRESHOWER
Silicon strips ~16m² ~137,000 channels

FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels

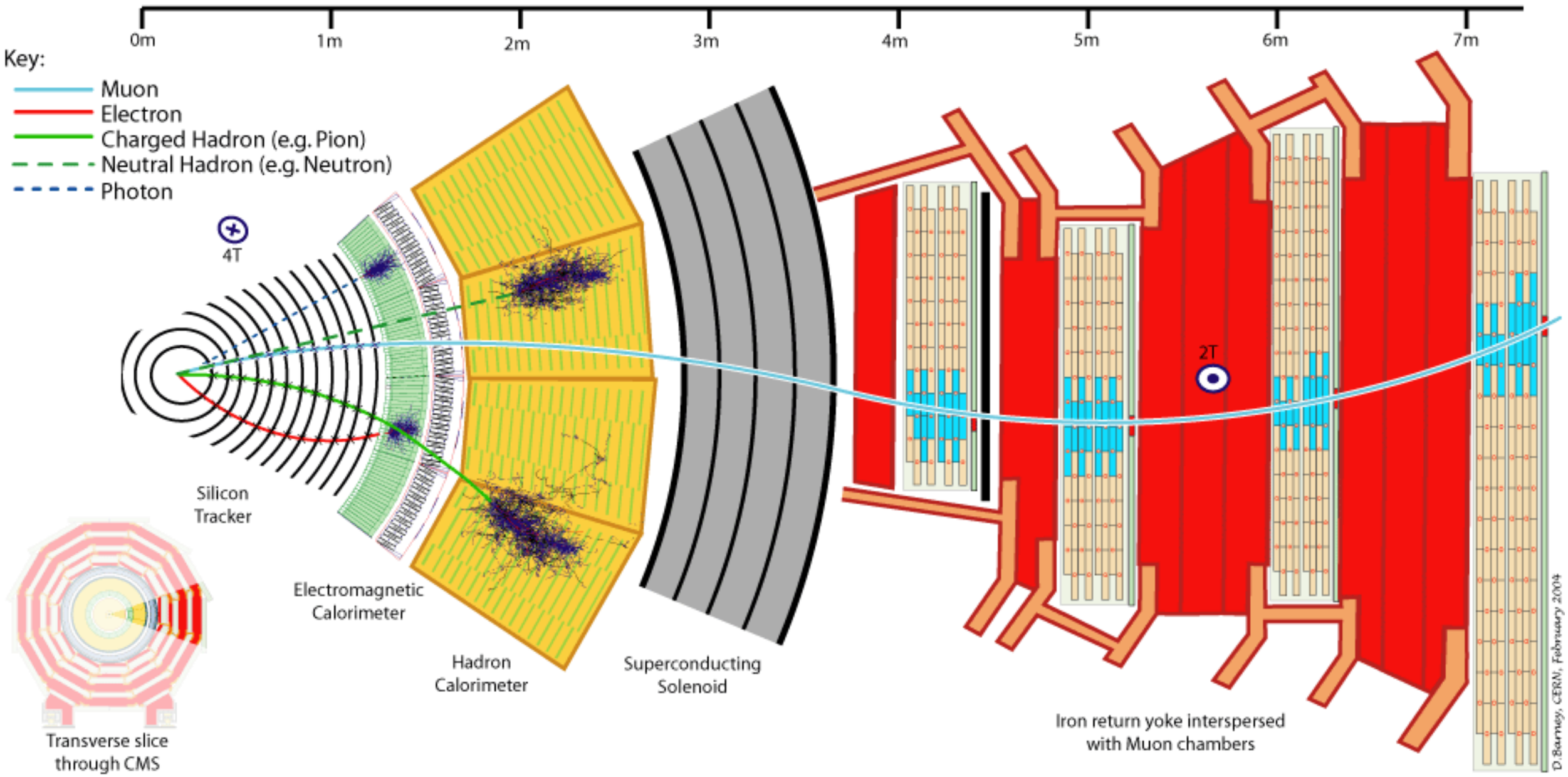
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating $PbWO_4$ crystals

HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

Detector Simulation

```
# GEANT simulation needs geometry and magnetic field
process.load('Configuration.StandardSequences.GeometryRecoDB_cff')
process.load('Configuration.StandardSequences.GeometrySimDB_cff')
process.load('Configuration.StandardSequences.MagneticField_cff')

# GEANT settings
process.load('Configuration.StandardSequences.SimIdeal_cff')

# geometry and magnetic are stored in database
# global tag needs to match the Era
process.load('Configuration.StandardSequences.FrontierConditions_GlobalTag_cf
f')
from Configuration.AlCa.GlobalTag import GlobalTag
process.GlobalTag = GlobalTag(process.GlobalTag,
'auto:phase1_2017_realistic', '')

# Schedule definition
process.simulation_step = cms.Path(process.psim)
process.schedule =
cms.Schedule(process.generation_step,process.genfiltersummary_step,process.si
mulation_step,process.endjob_step,process.FEVTDEBUGoutput_step)
```

Particle interaction with matter



Digitisation - Detector Response

```
vi step2_DIGI_L1TrackTrigger_L1_DIGI2RAW_HLT.py

# GEANT creates SimHits, SimHits are used to model the detector response
# Digis are the digitalised "SimHit"
process.load('Configuration.StandardSequences.Digi_cff')


python3 -i step2_DIGI_L1TrackTrigger_L1_DIGI2RAW_HLT.py
process.digitisation_step
cms.Path(generatorSmeared,generatorSmeared,simEcalTriggerPrimitiveDigis,simEcalDig
is,simEcalPreshowerDigis,simEcalEBTriggerPrimitiveDigis,simHcalTriggerPrimitiveDig
is,simHcalDigis,simHcalTTPDigis,simMuonCSCDigis,simMuonDTDigis,simMuonRPCDigis,sim
MuonGEMDigis,addPileupInfo,genPUProtons,genParticles,genParticlesForJets,genPartic
lesForJetsNoNu,ak4GenJets,ak8GenJets,ak4GenJetsNoNu,ak8GenJetsNoNu,genCandidatesFo
rMET,genParticlesForMETAllVisible,genMetCalo,genMetTrue,prunedTrackingParticles,pr
unedDigiSimLinks)

process.simMuonGEMDigis
cms.EDProducer("GEMDigiProducer",
    GE11ElecBkgParam0 = cms.double(406.249),
    GE11ElecBkgParam1 = cms.double(-2.90939),
    GE11ElecBkgParam2 = cms.double(0.00548191),
    fixedRollRadius = cms.bool(True),
    inputCollection = cms.string('g4SimHitsMuonGEMHits'),
    instLumi = cms.double(5),
    maxBunch = cms.int32(3),
    mightGet = cms.optional.untracked.vstring,
    minBunch = cms.int32(-5),
    mixLabel = cms.string('mix'),
```

Level 1 Trigger

```
# The L1 trigger is hardware, FPGA
# This is emulated using Digis
process.load('Configuration.StandardSequences.SimL1Emulator_cff')

process.L1simulation_step
cms.Path(simCaloStage2Layer1Digis,simCaloStage2Digis,simMuonGEMPadDigis,simMuonGEMPadD
igiClusters,simDtTriggerPrimitiveDigis,simCscTriggerPrimitiveDigis,simTwinMuxDigis,sim
BmtfDigis,simKBmtfStubs,simKBmtfDigis,simEmtfDigis,simOmtfDigis,simGmtCaloSumDigis,sim
GmtStage2Digis,simGtExtFakeStage2Digis,simGtStage2Digis,hgcalVFEProducer,hgcalConcentr
atorProducer,hgcalBackEndLayer1Producer,hgcalBackEndLayer2Producer,hgcalTowerMapProduc
er,hgcalTowerProducer,L1EGammaClusterEmuProducer,l1EGammaEEProducer,L1TkPrimaryVertex,
L1TkElectronsCrystal,L1TkElectronsLooseCrystal,L1TkElectronsEllipticMatchCrystal,L1TkI
soElectronsCrystal,L1TkPhotonsCrystal,L1TkElectronsHGC,L1TkElectronsEllipticMatchHGC,L
1TkIsoElectronsHGC,L1TkPhotonsHGC,L1TkMuons,pfClustersFromL1EGClusters,pfClustersFromC
ombinedCaloHCal,pfClustersFromCombinedCaloHF,pfClustersFromHGC3DClusters,pfTracksFromL
1TracksBarrel,l1pfProducerBarrel,pfTracksFromL1TracksHGCal,l1pfProducerHGCal,l1pfProdu
cerHGCalNoTK,l1pfProducerHF,l1pfCandidates,Phase1L1TJetProducer,Phase1L1TJetCalibrator
,l1PFMetCalo,l1PFMetPF,l1PFMetPuppi,l1NNTauProducer,l1NNTauProducerPuppi)
```

Digi2Raw - packing data into DAQ format

```
# the digi data is packed to look like the RAW data from CMS
process.load('Configuration.StandardSequences.DigiToRaw_cff')

process.digi2raw_step
cms.Path(caloLayer1RawFed1354,caloLayer1RawFed1356,caloLayer1RawFed1358,caloS
tage2Raw,bmtfStage2Raw,omtfStage2Raw,gmtStage2Raw,gtStage2Raw,SiStripDigiToRa
w,ecalPacker,esDigiToRaw,hcalRawDataVME,hcalRawDatauHTR,cscpacker,dtpacker,ra
wDataCollector)
```

```
# all data is collected by the Front-End Driver (FED)
# and stored in 64bit blocks
# each detector has readout chips
# readout chips are read out by modules (chambers)
# modules are readout in sectors
# sectors are readout in regions
# this is optimised for bandwidth and triggering

# digi has detector ID and hit information; strip, bx, energy, etc
# this must be converted to data from readout chip
# electronics mapping contains the mapping of the digi to the readout chip
```

High Level Trigger

```
# high level trigger
process.load('HLTrigger.Configuration.HLT_Fake2_cff')

process.HLTSchedule
cms.Schedule(*[ process.HLTriggerFirstPath, process.HLT_Physics_v1,
process.HLT_Random_v1, process.HLT_ZeroBias_v1, process.HLTriggerFinalPath,
process.HLTAnalyzerEndpath ])
```

```
process.hltGtStage2Digis
cms.EDProducer("L1TRawToDigi",
    CTP7 = cms.untracked.bool(False),
    DmxFWId = cms.uint32(0),
    FWId = cms.uint32(0),
    FWOverride = cms.bool(False),
    FedIds = cms.vint32(1404),
    InputLabel = cms.InputTag("rawDataCollector"),
```

Raw2Digi - unpacking DAQ data

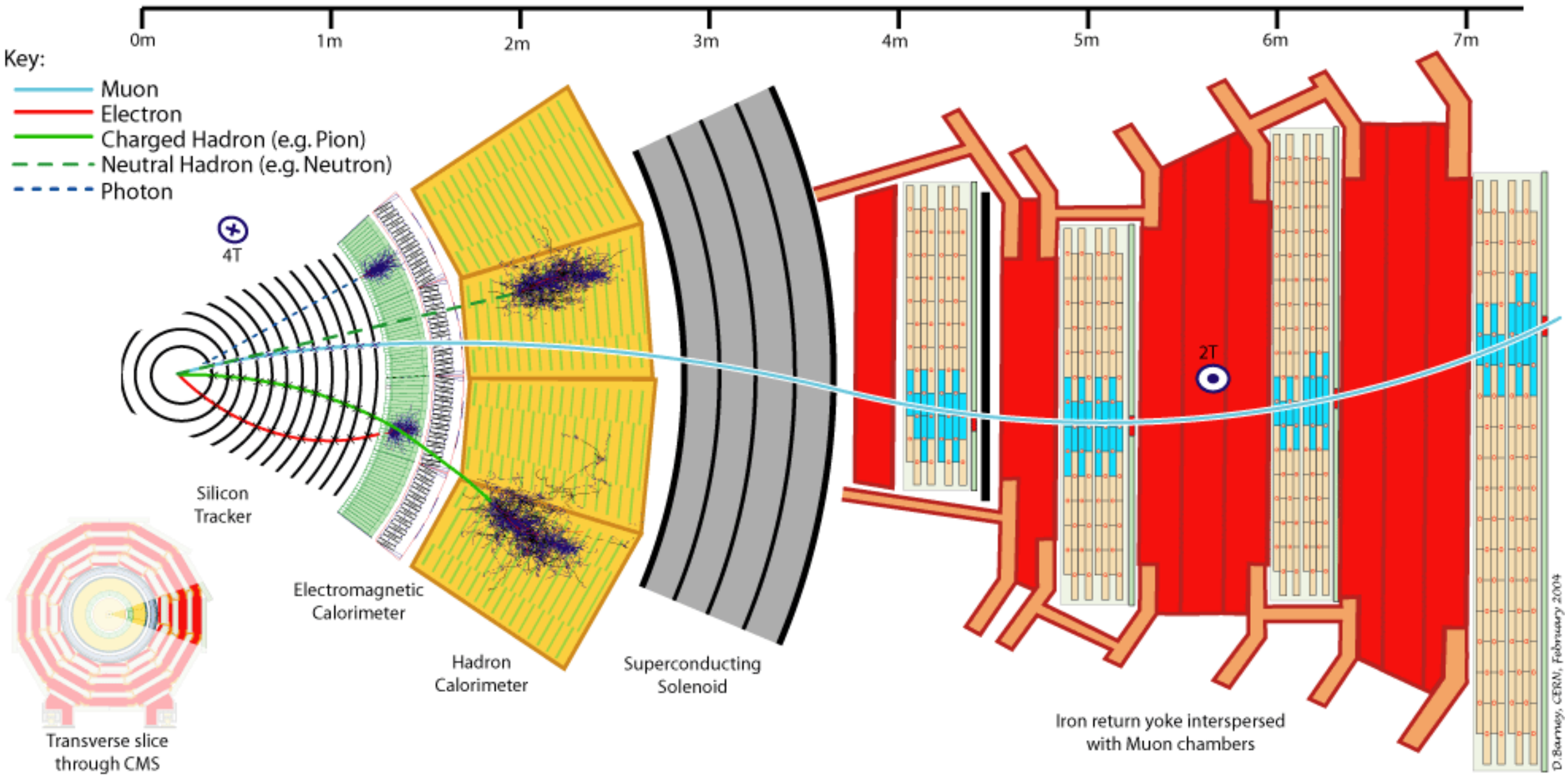
```
process.load('Configuration.StandardSequences.RawToDigi_cff')

process.raw2digi_step
cms.Path(csctfDigis,dttfDigis,gtDigis,caloStage1Digis,caloStage1FinalDigis,ca
loStage1LegacyFormatDigis,gctDigis,rpcTwinMuxRawToDigi,twinMuxStage2Digis,bmt
fDigis,omtfStage2Digis,rpcCPPFRawToDigi,emtfStage2Digis,caloLayer1Digis,caloS
tage2Digis,gmtStage2Digis,gtStage2Digis,siStripDigis,ecalDigis,ecalPreshowerD
igis,hcalDigis,muonCSCDigis,muonDTDigis,muonRPCDigis,scalersRawToDigi,tcdsDig
is,onlineMetaDataDigis,totemTriggerRawToDigi,totemRPRawToDigi,ctppsDiamondRaw
ToDigi,totemTimingRawToDigi,ctppsPixelDigis,hgcalDigis)
```

```
process.muonRPCDigis
cms.EDProducer("RPCUnpackingModule",
    InputLabel = cms.InputTag("rawDataCollector"),
    doSynchro = cms.bool(True),
    mightGet = cms.optional.untracked.vstring
)

# unpack the RAW data into digis
# For real data, this is the starting point
# For simulation, this digi and the previously digi should be identical
```

Reconstruction

```
process.load('Configuration.StandardSequences.Reconstruction_cff')
# Local Reconstruction
# digis are not directly used in reconstruction in most cases
# digis are clustered into reconstructed hits "rechits"
# for Muon detectors, rechits are reconstructed in to segments

# Reconstruction
# for Muons, segments are used to make stand alone muons (muon system only track)
# inner tracker rechits are used to make tracks
# inner tracker tracks and stand alone muons are matched and combined to make muons
# particle flow rechits are made for calorimeters
# all above is used to make particle flow objects, muon, electron, photon and charged/
neutron hadron
# particle flow objects are used to make jets

# Particle Identification
# once particles and jets are reconstructed, classification variables are created
# the classification variables are used to define loose, medium, tight

process.load('Configuration.StandardSequences.PATMC_cff')
# Physics Analysis Tools
# some selections are applied to physics objects
# isolation variables are calculated
# event filters are made, i.e. chargedHadronTrackResolutionFilter
# objects are slimmed (reduced in size) to store in smaller files for quicker analysis
```

Particle interaction with matter



when workflow is done

```
~/CMSSW_12_1_0/src/
10024.0_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano > ls

cmdLog
DQM_V0001_R000000001__Global__CMSSW_X_Y_Z__RECO.root
EcalESAlign.root
HcalCalHBHEMuonFilter.root
MuAlOverlaps.root
SiPixelCalSingleMuonLoose.root
SiPixelCalSingleMuonTight.root
SiStripCalMinBias.root
step1.root
step1_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano.log
step2_DIGI_L1_DIGI2RAW_HLT.py
step2.root
step2_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano.log
step3_inDQM.root
step3_inMINIAODSIM.root
step3_RAW2DIGI_L1Reco_RECO_RECOSIM_EI_PAT_VALIDATION_DQM.py
step3.root
step3_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano.log
step4_HARVESTING.py
step4_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano.log
step5_ALCA.py
step5_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano.log
step6_inDQM.root
step6_NANO_DQM.py
step6.root
step6_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+ALCA+Nano.log
TkAlDiMuonAndVertex.root
TkAlJpsiMuMu.root
TkAlMinBias.root
TkAlMuonIsolated.root
TkAlUpsilonMuMu.root
TkAlZMuMu.root
TTbar_13TeV_TuneCUETP8M1_cfi_GEN_SIM.py
```

checking contents of file

```
edmDumpEventContent step3.root
  Type                                    Module                          Label             Process
  ------------------------------------------------------------------------------------------------
  GenEventInfoProduct                     "generator"                     ""                "SIM"
  edm::HepMCProduct                       "generatorSmeared"              ""                "SIM"
  edm::TriggerResults                     "TriggerResults"                ""                "SIM"
  vector<SimTrack>                        "g4SimHits"                     ""                "SIM"
  vector<SimVertex>                       "g4SimHits"                     ""                "SIM"
  GlobalObjectMapRecord                   "hltGtStage2ObjectMap"          ""                "HLT"
  MuonDigiCollection<DTLayerId,DTDigiSimLink>    "simMuonDTDigis"                 ""                "HLT"
  ROOT::Math::PositionVector3D<ROOT::Math::Cartesian3D<float>,ROOT::Math::DefaultCoordinateSystemTag>
"genParticles"                "xyz0"                "HLT"
  double                                  "ak4GenJets"                    "rho"             "HLT"
  double                                  "ak4GenJetsNoNu"                "rho"             "HLT"
  double                                  "ak8GenJets"                    "rho"             "HLT"
  double                                  "ak8GenJetsNoNu"                "rho"             "HLT"
  double                                  "ak4GenJets"                    "sigma"           "HLT"
  double                                  "ak4GenJetsNoNu"                "sigma"           "HLT"
  double                                  "ak8GenJets"                    "sigma"           "HLT"
  double                                  "ak8GenJetsNoNu"                "sigma"           "HLT"
  edm::DetSetVector<GEMDigiSimLink>       "simMuonGEMDigis"               "GEM"             "HLT"
  edm::DetSetVector<RPCDigiSimLink>       "simMuonRPCDigis"               "RPCDigiSimLink"   "HLT"
  edm::DetSetVector<StripDigiSimLink>      "simMuonCSCDigis"               "MuonCSCStripDigiSimLinks"   "HLT"
  edm::DetSetVector<StripDigiSimLink>      "simMuonCSCDigis"               "MuonCSCWireDigiSimLinks"   "HLT"
  edm::HepMCProduct                       "generatorSmeared"              ""                "HLT"
  edm::TriggerResults                     "TriggerResults"                ""                "HLT"
  float                                   "genParticles"                  "t0"              "HLT"
  int                                     "addPileupInfo"                 "bunchSpacing"    "HLT"
  vector<DcsStatus>                       "hltScalersRawToDigi"           ""                "HLT"
  vector<PileupSummaryInfo>               "addPileupInfo"                 ""                "HLT"
  vector<double>                          "ak4GenJets"                    "rhos"            "HLT"
  vector<double>                          "ak4GenJetsNoNu"                "rhos"            "HLT"
  vector<double>                          "ak8GenJets"                    "rhos"            "HLT"
  vector<double>                          "ak8GenJetsNoNu"                "rhos"            "HLT"
  vector<double>                          "ak4GenJets"                    "sigmas"          "HLT"
  vector<double>                          "ak4GenJetsNoNu"                "sigmas"          "HLT"
  vector<double>                          "ak8GenJets"                    "sigmas"          "HLT"
  vector<double>                          "ak8GenJetsNoNu"                "sigmas"          "HLT"
  vector<int>                             "genParticles"                  ""                "HLT"
  vector<reco::GenJet>                    "ak4GenJets"                    ""                "HLT"
  vector<reco::GenJet>                    "ak4GenJetsNoNu"                ""                "HLT"
  vector<reco::Track>                     "generalTracks"                 ""                "RECO"
```

make ED Analyzer package

```
gate:~/CMSSW_12_1_0/src > mkdir Demo
gate:~/CMSSW_12_1_0/src > cd Demo
gate:~/CMSSW_12_1_0/src/Demo > mkedanlzr
Please enter edanalyzer name: DemoAnalyzer
New package "DemoAnalyzer" of EDAnalyzer type is successfully generated
DemoAnalyzer/
|  plugins/
|  |-- DemoAnalyzer.cc
|  |-- BuildFile.xml
|  test/
|  |-- test_catch2_main.cc
|  |-- BuildFile.xml
|  |-- test_catch2_DemoAnalyzer.cc
|  python/
Total: 3 directories, 5 files
gate:~/CMSSW_12_1_0/src/Demo > cd ..
```

Compiling the package

```
gate:~/CMSSW_12_1_0/src > scram b
>> Local Products Rules ..... started
>> Local Products Rules ..... done
>> Entering Package gemsw/Analysis
>> Leaving Package gemsw/Analysis
>> Package gemsw/Analysis built
>> Entering Package gemsw/Geometry
>> Leaving Package gemsw/Geometry
>> Package gemsw/Geometry built
>> Entering Package gemsw/EventFilter
>> Creating project symlinks
>> Leaving Package gemsw/EventFilter
>> Package gemsw/EventFilter built
>> Subsystem gemsw built
>> Entering Package Geometry/GEMGeometry
>> Leaving Package Geometry/GEMGeometry
>> Package Geometry/GEMGeometry built
>> Subsystem Geometry built
>> Entering Package SimG4Core/Application
>> Leaving Package SimG4Core/Application
>> Package SimG4Core/Application built
>> Subsystem SimG4Core built
>> Entering Package Fireworks/Geometry
>> Leaving Package Fireworks/Geometry
>> Package Fireworks/Geometry built
>> Subsystem Fireworks built
>> Entering Package Demo/DemoAnalyzer
Entering library rule at src/Demo/DemoAnalyzer/plugins
>> Compiling edm plugin /pad/jlee/CMSSW_12_1_0/src/Demo/DemoAnalyzer/plugins/
DemoAnalyzer.cc
```

includes

```
vi Demo/DemoAnalyzer/plugins/DemoAnalyzer.cc

// framework
#include "FWCore/Framework/interface/Frameworkfwd.h"
// analyzer base class
#include "FWCore/Framework/interface/one/EDAnalyzer.h"
// event information
#include "FWCore/Framework/interface/Event.h"
// making into module
#include "FWCore/Framework/interface/MakerMacros.h"
// added parameters from python
#include "FWCore/ParameterSet/interface/ParameterSet.h"
// input data names
#include "FWCore/Utilities/interface/InputTag.h"
// track class, to get track data
#include "DataFormats/TrackReco/interface/Track.h"
#include "DataFormats/TrackReco/interface/TrackFwd.h"
```

class definition

```
class DemoAnalyzer : public edm::one::EDAnalyzer<edm::one::SharedResources> {
public:
  explicit DemoAnalyzer(const edm::ParameterSet&);
  ~DemoAnalyzer();

  static void fillDescriptions(edm::ConfigurationDescriptions& descriptions);

private:
  void beginJob() override;
  void analyze(const edm::Event&, const edm::EventSetup&) override;
  void endJob() override;

  // ----------member data ---------------------------
  edm::EDGetTokenT<TrackCollection> tracksToken_;  //used to select what tracks to
read from configuration file
#ifdef THIS_IS_AN_EVENTSETUP_EXAMPLE
  edm::ESGetToken<SetupData, SetupRecord> setupToken_;
#endif
};
```

constructor and destructor

```
DemoAnalyzer::DemoAnalyzer(const edm::ParameterSet& iConfig)
    :
tracksToken_(consumes<TrackCollection>(iConfig.getUntrackedParameter<edm::InputTag>("t
racks"))) {
#ifdef THIS_IS_AN_EVENTSETUP_EXAMPLE
  setupDataToken_ = esConsumes<SetupData, SetupRecord>();
#endif
  //now do what ever initialization is needed
}

DemoAnalyzer::~DemoAnalyzer() {
  // do anything here that needs to be done at desctruction time
  // (e.g. close files, deallocate resources etc.)
  //
  // please remove this method altogether if it would be left empty
}
```

analyze function

```
// ------------ method called for each event  ------------
void DemoAnalyzer::analyze(const edm::Event& iEvent, const edm::EventSetup& iSetup) {
  using namespace edm;

  for (const auto& track : iEvent.get(tracksToken_)) {
    // do something with track parameters, e.g, plot the charge.
    // int charge = track.charge();
  }

#ifdef THIS_IS_AN_EVENTSETUP_EXAMPLE
  // if the SetupData is always needed
  auto setup = iSetup.getData(setupToken_);
  // if need the ESHandle to check if the SetupData was there or not
  auto pSetup = iSetup.getHandle(setupToken_);
#endif
}
```

other functions

```
// ------------ method called once each job just before starting event loop
void DemoAnalyzer::beginJob() {
  // please remove this method if not needed
}

// ------------ method called once each job just after ending the event loop
void DemoAnalyzer::endJob() {
  // please remove this method if not needed
}

// ------------ method fills 'descriptions' with the allowed parameters for the module
void DemoAnalyzer::fillDescriptions(edm::ConfigurationDescriptions& descriptions) {
  //The following says we do not know what parameters are allowed so do no validation
  // Please change this to state exactly what you do use, even if it is no parameters
  edm::ParameterSetDescription desc;
  desc.setUnknown();
  descriptions.addDefault(desc);

  //Specify that only 'tracks' is allowed
  //To use, remove the default given above and uncomment below
  //ParameterSetDescription desc;
  //desc.addUntracked<edm::InputTag>("tracks","ctfWithMaterialTracks");
  //descriptions.addWithDefaultLabel(desc);
}
```

looking at track.h class

https://github.com/cms-sw/cmssw/blob/master/DataFormats/TrackReco/interface/Track.h

```
namespace reco {

  class Track : public TrackBase {
  public:
    /// default constructor
    Track() {}
```

https://github.com/cms-sw/cmssw/blob/master/DataFormats/TrackReco/interface/TrackBase.h

```
    /// track electric charge
    int charge() const;
    /// q / p
    double qoverp() const;
    /// polar angle
    double theta() const;
    /// dxy parameter in perigee convention (d0 = -dxy)
    double d0() const;
    /// dz parameter (= dsz/cos(lambda)). This is the track z0 w.r.t (0,0,0) only if the refPoint is close to (0,0,0). See also function dz(myBeamSpot)
    double dz() const;
    /// track transverse momentum
    double pt() const;
```

cout pt

```
void DemoAnalyzer::analyze(const edm::Event& iEvent, const edm::EventSetup&
iSetup) {
    using namespace edm;

    for (const auto& track : iEvent.get(tracksToken_)) {
      // do something with track parameters, e.g, plot the charge.
      // int charge = track.charge();
      std::cout << "track pt="<< track.pt() << std::endl;
    }

  #ifdef THIS_IS_AN_EVENTSETUP_EXAMPLE
    // if the SetupData is always needed
    auto setup = iSetup.getData(setupToken_);
    // if need the ESHandle to check if the SetupData was there or not
    auto pSetup = iSetup.getHandle(setupToken_);
  #endif
}
```

runDemoAnalyzer.py

```
import FWCore.ParameterSet.Config as cms

process = cms.Process("Demo")

process.maxEvents = cms.untracked.PSet( input =
cms.untracked.int32(10) )

process.source = cms.Source("PoolSource",
    fileNames = cms.untracked.vstring('file:/pad/jlee/CMSSW_12_1_0/
src/
10024.0_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFake
HLT+HARVESTFakeHLT+ALCA+Nano/step3.root' )
)

process.demo = cms.EDAnalyzer('DemoAnalyzer',
    tracks = cms.untracked.InputTag("generalTracks")
)

process.p = cms.Path(process.demo)
```

cmsRun runDemoAnalyzer.py

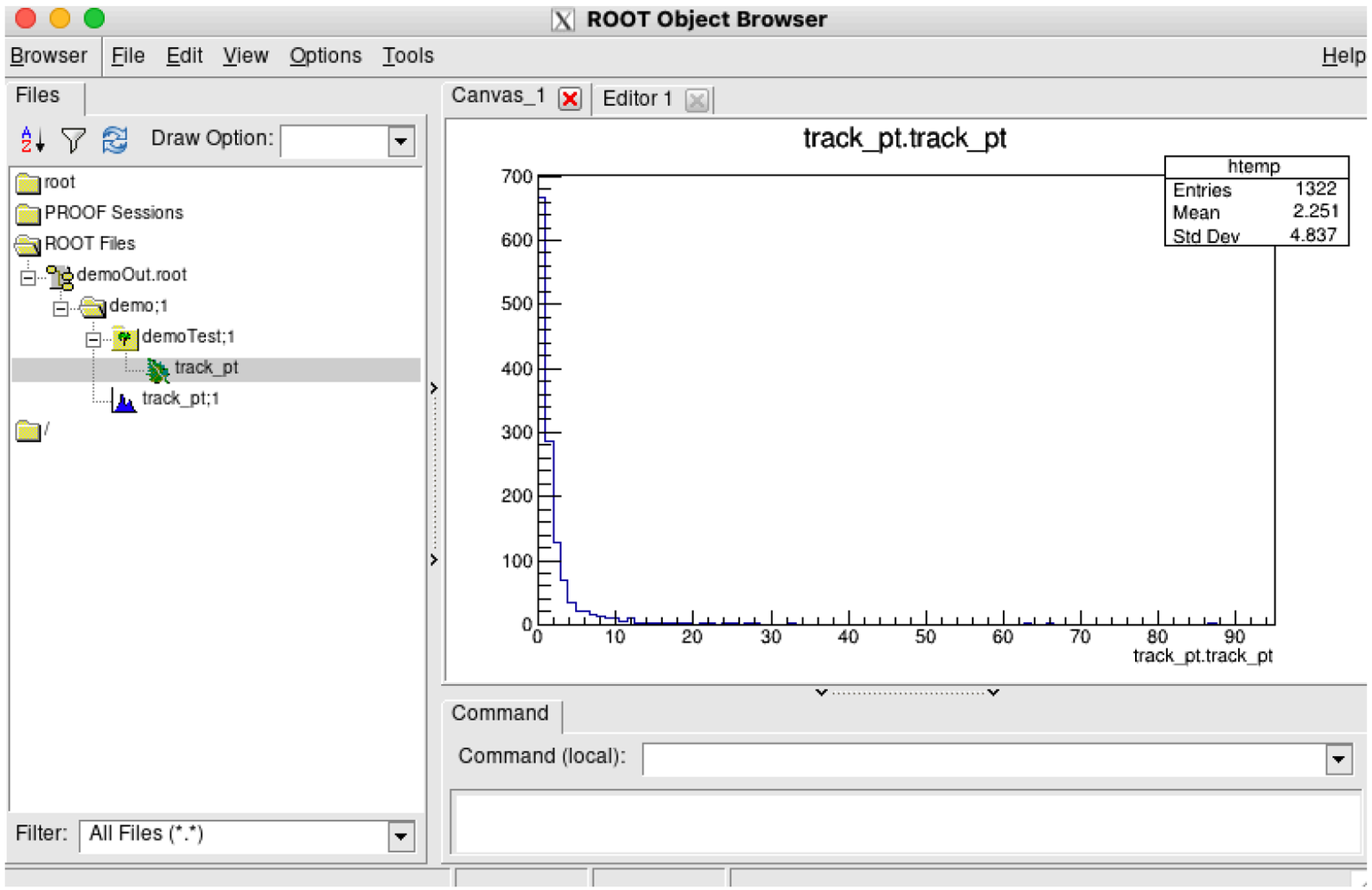
```
16-Nov-2021 07:59:20 KST  Initiating request to open file file:/pad/jlee/CMSSW_12_1_0/
src/
10024.0_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+
ALCA+Nano/step3.root
16-Nov-2021 07:59:26 KST  Successfully opened file file:/pad/jlee/CMSSW_12_1_0/src/
10024.0_TTbar_13+2017+TTbar_13TeV_TuneCUETP8M1_GenSim+Digi+RecoFakeHLT+HARVESTFakeHLT+
ALCA+Nano/step3.root
Begin processing the 1st record. Run 1, Event 1, LumiSection 1 on stream 0 at 16-
Nov-2021 07:59:30.538 KST
track pt=10.2573
track pt=1.1159
track pt=0.382952
track pt=0.38969
track pt=0.678574
track pt=0.377063
track pt=1.30795
track pt=1.0383
track pt=7.51442
track pt=1.50332
track pt=0.760519
track pt=0.703714
track pt=5.71689
track pt=8.3916
track pt=0.970788
track pt=1.69088
track pt=1.04599
track pt=8.21291
track pt=0.745155
track pt=23.8043
```

Saving to ttree

```
// add in Demo/DemoAnalyzer/plugins/BuildFile.xml
<use name="FWCore/ServiceRegistry"/>
<use name="CommonTools/UtilAlgos"/>


// add in runDemoAnalyzer.py
process.TFileService = cms.Service("TFileService",
    fileName = cms.string('demoOut.root')
)


// add includes
#include "FWCore/ServiceRegistry/interface/Service.h"
#include "CommonTools/UtilAlgos/interface/TFileService.h"
#include "TFile.h"
#include "TH1.h"
#include "TTree.h"

// add in class def
edm::Service<TFileService> fs_;
TTree* tree_;
std::vector<float> track_pt;
TH1D* h_track_pt;

// add in constructor
tree_ = fs_->make<TTree>("demoTest", "demoTest");
tree_->Branch("track_pt", &track_pt);
h_track_pt = fs_->make<TH1D>("track_pt", "track_pt", 100, 0, 100);
```

Saving to ttree

```
void DemoAnalyzer::analyze(const edm::Event& iEvent, const edm::EventSetup&
iSetup) {
  using namespace edm;
  track_pt.clear();
  for (const auto& track : iEvent.get(tracksToken_)) {
    std::cout << "track pt="<< track.pt() << std::endl;
    track_pt.push_back(track.pt());
    h_track_pt->Fill(track.pt());
  }
  tree_->Fill();
}
```



EDProducer

1. EDProducer produces EDM objects
2. gen-sim-reco chain is built with EDProducers
3. They are stored in files with PoolOutputModule
4. RECO, AOD, MINIAOD are different tiers of EDM

PoolOutput

```
process.RECOSIMoutput = cms.OutputModule("PoolOutputModule",
    dataset = cms.untracked.PSet(
        dataTier = cms.untracked.string('GEN-SIM-RECO'),
        filterName = cms.untracked.string('')
    ),
    fileName = cms.untracked.string('file:step3.root'),
    outputCommands = process.RECOSIMEventContent.outputCommands,
    splitLevel = cms.untracked.int32(0)
)
```

Exercise

get the vector<reco::PFJet> "ak4PFJets" object and store into ttree the pt, eta, phi, mass as well as charged/neutral hadron multiplicities, photon and electron multiplicities.