

FCCAnalyses and RootDataFrame in a nutshell

Common analysis framework

The need

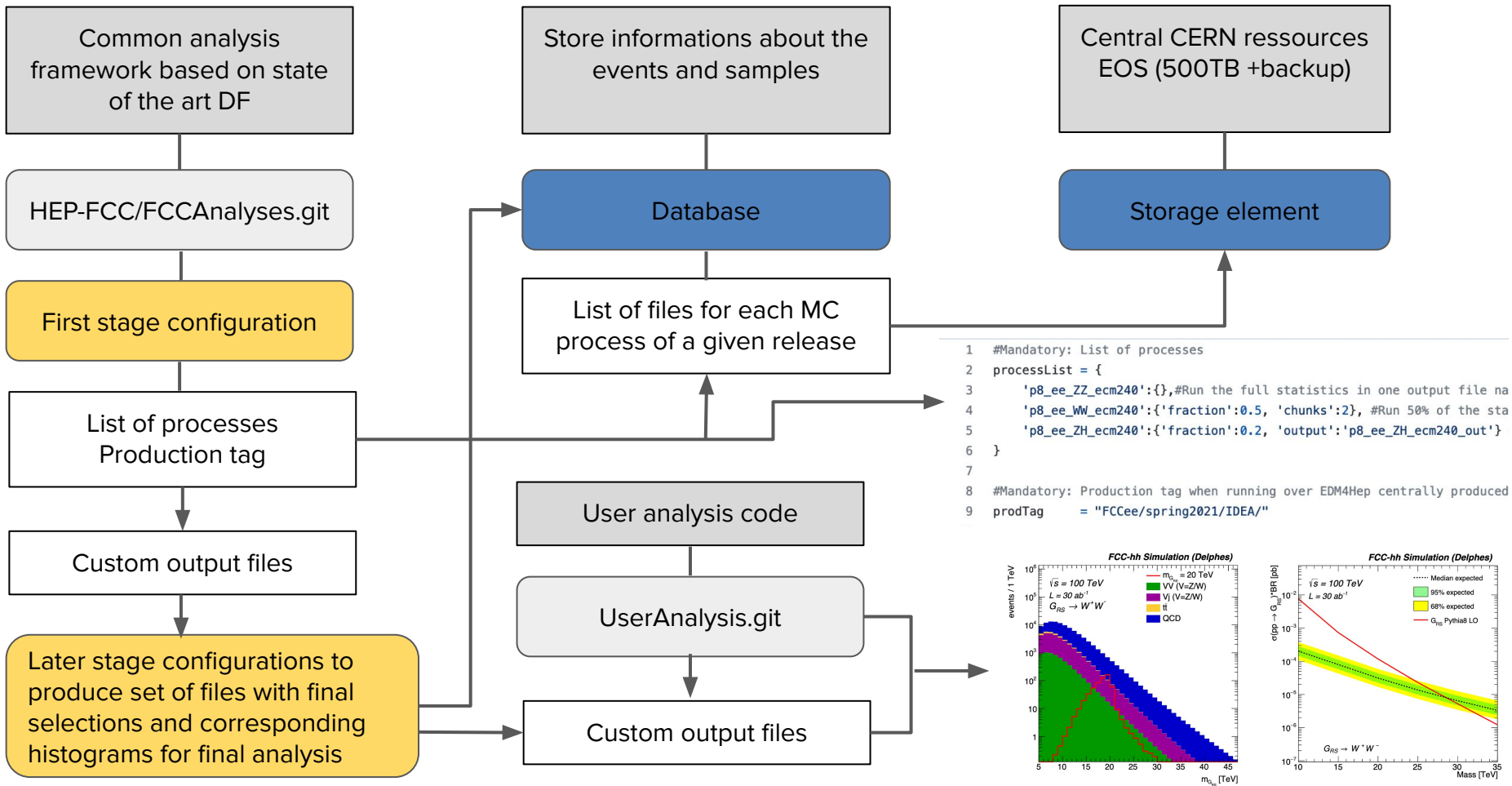
- Create a custom tool to support the analysis of large amount of data
- Flexible configuration, quick connection with events produced

The challenges

- Start from scratch with my own expertise in data analyses
- Make it efficient, low maintenance and easy to run from the user perspective

The results

- Default tool in use for physics results
- Can process tens of billions of events on batch in couple of hours



```
[helsens@lxplus789]:~/FCCsoft/HEP-FCC/FCCAnalyses$ fccanalysis run --help
usage: fccanalysis run [-h] [--files-list FILES_LIST [FILES_LIST ...]] [--output OUTPUT] [--nevents NEVENTS] [--test] [--bench]
[--ncpus NCPUS] [--preprocess] [--validate] [--rerunfailed] [--jobdir JOBDIR]
      [--eloglevel {kUnset,kFatal,kError,kWarning,kInfo,kDebug}] [--batch]
      pathToAnalysisScript
```

optional arguments:

-h, --help show this help message and exit

User options:

pathToAnalysisScript path to analysis script
--files-list FILES_LIST [FILES_LIST ...]
Specify input file to bypass the processList
--output OUTPUT Specify output file name to bypass the processList and or outputList, default output.root
--nevents NEVENTS Specify max number of events to process
--test Run over the test file
--bench Output benchmark results to a JSON file
--ncpus NCPUS Set number of threads
--preprocess Run preprocessing
--validate Validate a given production
--rerunfailed Rerun failed jobs
--jobdir JOBDIR Specify the batch job directory
--eloglevel {kUnset,kFatal,kError,kWarning,kInfo,kDebug}
Specify the RDataFrame ELogLevel

Internal options, NOT FOR USERS:

--batch Submit on batch

```
1 #FCCAnalyses first stage configuration
2
3 #List of processes when you want to use samples from the database
4 processList = {
5     'p8_noBES_ee_Ztautau_ecm91_EvtGen_TauMinus2MuMuMu': {},
6     'p8_noBES_ee_Ztautau_ecm91_EvtGen_TauMinus2PiPiPinu': {}
7 }
8
9 #Production tag when running over the processList
10 prodTag     = "FCce/spring2021/IDEA/"
11
12 #Output directory, default is local running directory
13 outputDir   = "Tau3Mu"
14
15 #Test file trigger when using --test command line option
16 testFile= "/eos/experiment/fcc/ee/generation/DelphesEvents/spring2021/IDEA/p8_noBES_ee_Ztautau_ecm91_EvtGen_TauMinus2PiPiPinu/events_179808277.root"
17
18 #Many other options to run on batch, etc, etc..
19
```

```

19
20 #Mandatory: RDataFrame class where the use defines the operations on the TTree
21 class RDataFrame():
22     #_____
23     #Mandatory: analysers function to define the analysers to process, please make sure you return the last dataframe, in this example it is df2
24     def analysers(df):
25         #Inside this function we use RootDataFrame syntax
26         df1 = (
27             df
28             #Possibility to Alias complex names
29             .Alias("Muon0", "AllMuon#0.index")
30             #Using C++ code define inside analyzers/dataframe, input is collections inside the input file
31             .Define("muons", "FCCAnalyses::ReconstructedParticle::get(Muon0, ReconstructedParticles)")
32             #Can now use the collection we created just before
33             .Define("n_muons", "FCCAnalyses::ReconstructedParticle::get_n(muons) ")
34             #FCCAnalyses namespace loaded by default, can be dropped
35             .Define("n_muons_2", "ReconstructedParticle::get_n(muons) ")
36             #To call more complex code that needs a constructor
37             .Define("mc_tauplus_vec", ROOT.MCParticle.sel_pdgID(-15, 0), ["Particles"])
38             #Can filter on the collection
39             .Filter("mc_tauplus_vec.size()==1")
40             #Get the element (edm4hep::ParticleData)
41             .Define("mc_tauplus", "mc_tauplus_vec[0]")
42             #Get the px momentum
43             .Define("mc_tauplus_px", "mc_tauplus.momentum.x")
44         )
45         #Can also declare some code inline
46         ROOT.gInterpreter.Declare("""
47         float myDummyCode(){return 1.;;}
48         """)
49         df2 = (
50             df1
51             #and now we can use it
52             .Define("myDummyCollection", "myDummyCode()")
53         )
54         return df2

```

```
56
57 #_____
58 #Mandatory: output function, please make sure you return the branchlist as a python list
59 def output():
60     branchList = [
61         "n_muons",
62         "mc_tauplus_px",
63     ]
64     return branchList
65
```

```
fccanalysis run examples/FCCee/higgs/mH-recoil/mumu/analysis_stage1.py
```

```
fccanalysis run examples/FCCee/higgs/mH-recoil/mumu/analysis_stage1.py \  
  --output <myoutput.root> \  
  --files-list <file.root or file1.root file2.root or file*.root>
```