Papas

A Parametrized Particle Simulation for FCC-ee

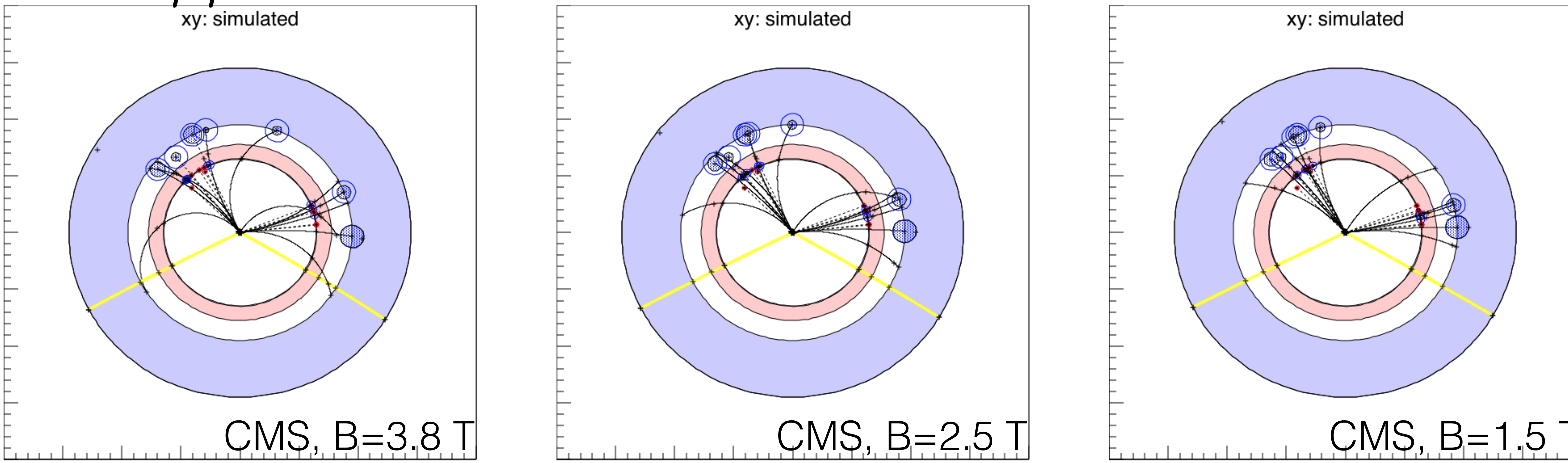
Colin Bernet (IPNL), Alice Robson (CERN)

23rd of November, 2016

2nd Mini-Workshop on FCC-ee detector requirements

Why a parametrized simulation?

$\mathrm{ZH} \rightarrow \mu\mu bb$

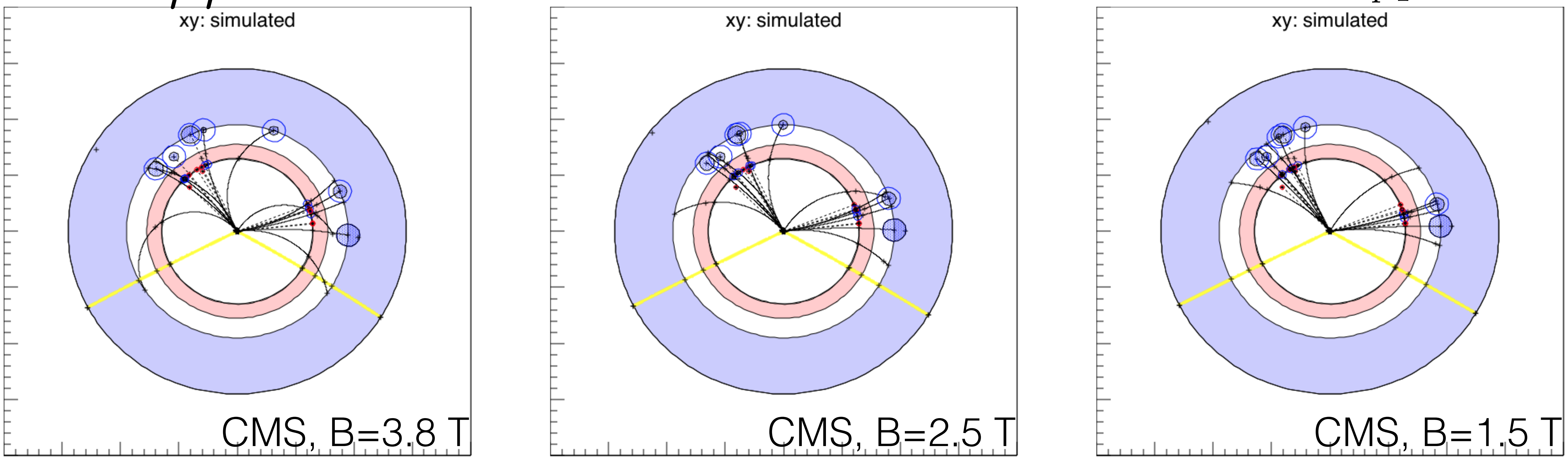


- What is the required detector performance for a given physics programme?
  - set up a number of different analyses
  - scan detector parameters
  - redo simulation and analyses

Need a realistic reconstruction

$\mathrm{ZH} \rightarrow \mu\mu bb$

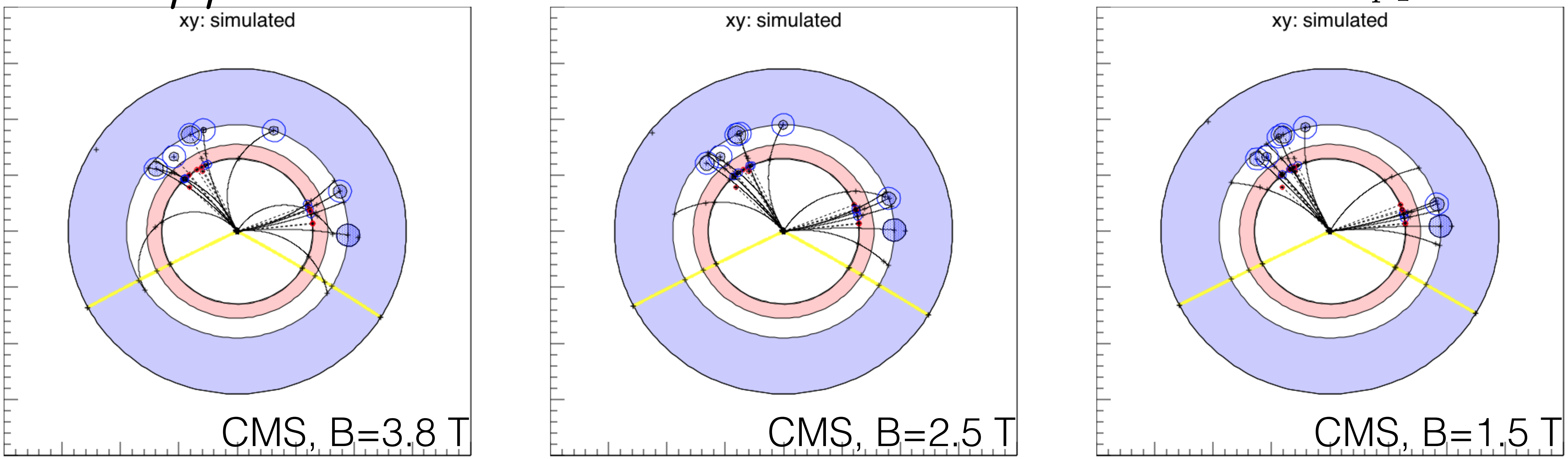
$p_{\mathrm{T}} > 1\,\mathrm{GeV}$



- Smearing the jet energy with a naive resolution $\propto X\%/\sqrt{E}$ does not capture the effect of the B field
- Need to reconstruct:
    - particles from the tracks and clusters (particle flow)
    - jets of particles

Need a varied physics programme

$\mathrm{ZH} \to \mu\mu bb$ $p_\mathrm{T} > 1\,\mathrm{GeV}$



- The physics programme must be sensitive to the parameters being studied
  - e.g. jet energy resolution does not matter much in this channel
  - B field affects resolution on muon momentum and thus on recoil mass

A possible roadmap

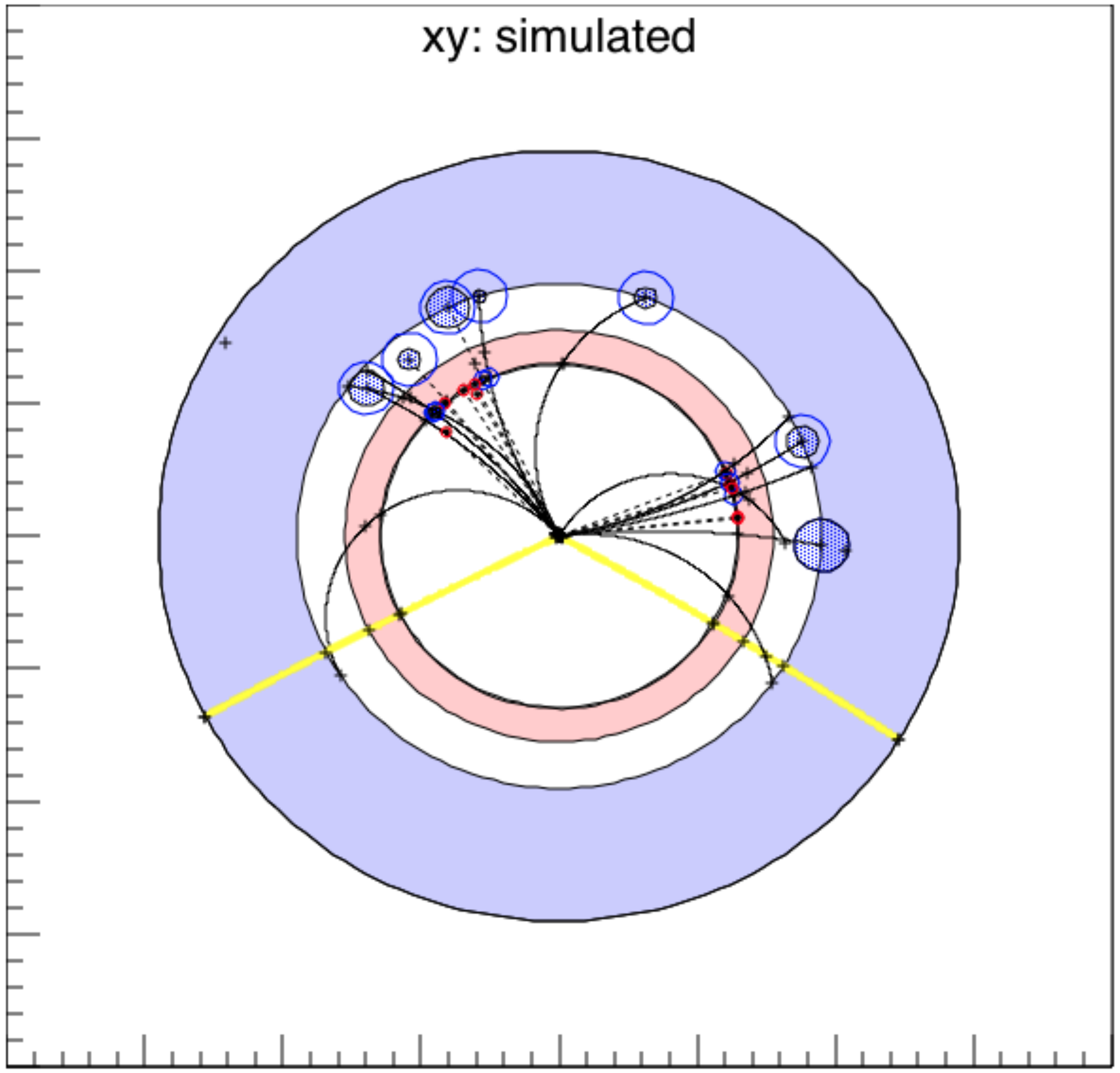
Let's do like our FCC-hh colleagues

- What is the required performance?
  - Detector parameter scans in analysis with parametrized simulation
    - Papas (C.B., A. Robson)
    - Delphes (particle flow mode, M. Selvaggi)
    - both integrated in FCC software,
      can use both to bracket the results (easy!)
- How to reach this performance?
  - Design specific parts of the detector in full sim (A. Zaborowska et al.)
    - e.g. ECAL + preshower to see how well ECAL showers can be separated.
    - Just need a piece of calorimeter
    - No need to simulate the whole event (particle gun, fast/full framework)

Papas: Outline

Parametrized particle simulation

- Simulate particle flow inputs:
  - Inner tracker
  - Calorimeters
- Particle flow reconstruction
- Jets & jet flavour tagging
- Leptons
- Example analyses



Inner tracker

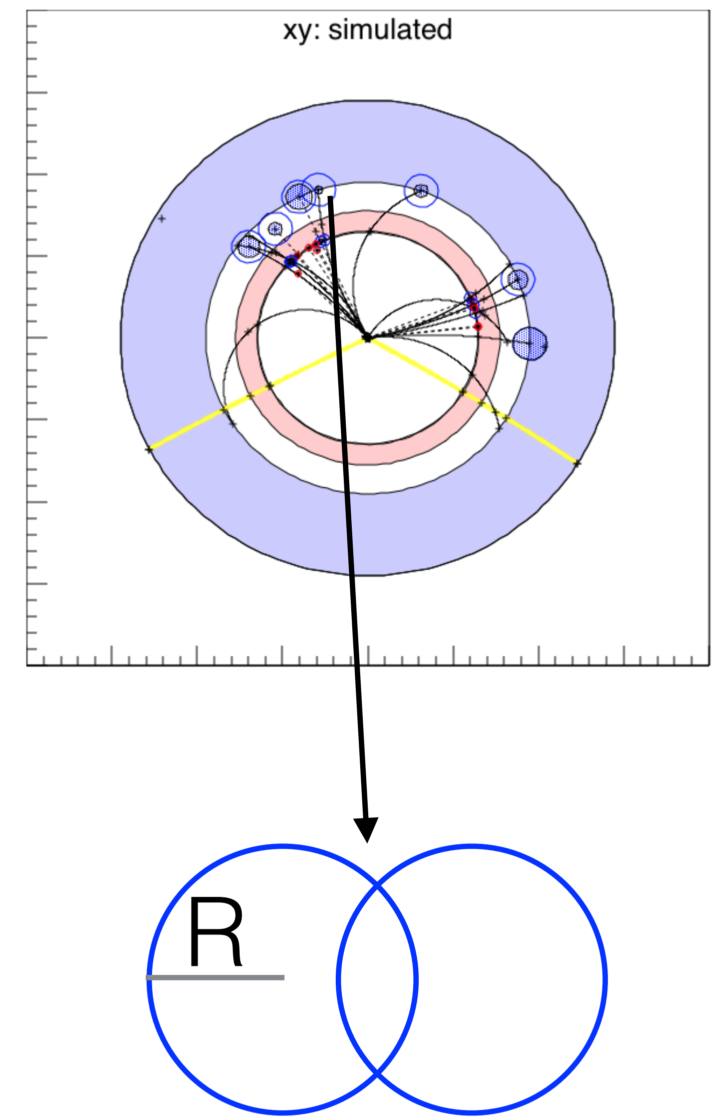
```
class Tracker(DetectorElement):
    #TODO acceptance and resolution
    #depend on the particle type

    def __init__(self):
        volume = VolumeCylinder('tracker', 1.29, 1.99)
        mat = material.void
        super(Tracker, self).__init__('tracker', volume,  mat)

    def acceptance(self, track):
        pt = track.pt
        eta = abs(track.p3.Eta())
        if eta < 1.35 and pt>0.5:
            return random.uniform(0,1)<0.95
        elif eta < 2.5 and pt>0.5:
            return random.uniform(0,1)<0.9
        else:
            return False

    def pt_resolution(self, track):
        # TODO: depends on the field
        pt = track.pt
        return 1.1e-2
```

- Inputs:
    - simple geometry (cylinder)
    - acceptance model
    - resolution model
    - (+ B field)

- These methods are mandatory
- You can write anything you want inside (very flexible)

Calorimeters

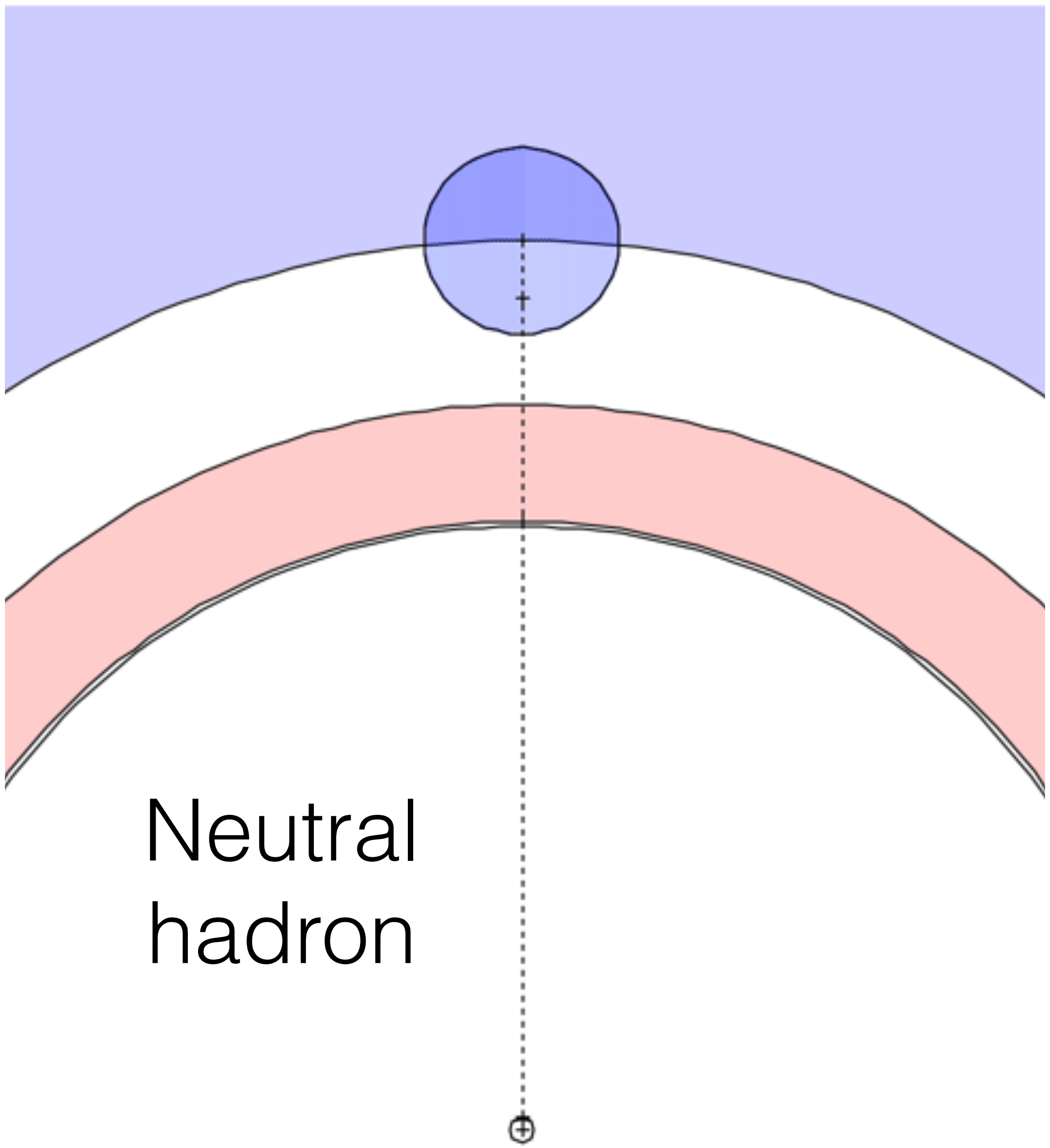
- Similar code
- Inputs:
  - simple geometry (2 cylinders)
  - material
    - hadron showers in ECAL
  - energy resolution and response
  - acceptance
    - thresholds
  - cluster size R
    - a way to model calorimeter granularity



Merged clusters
- energy is summed
- **cannot be resolved in particle flow**

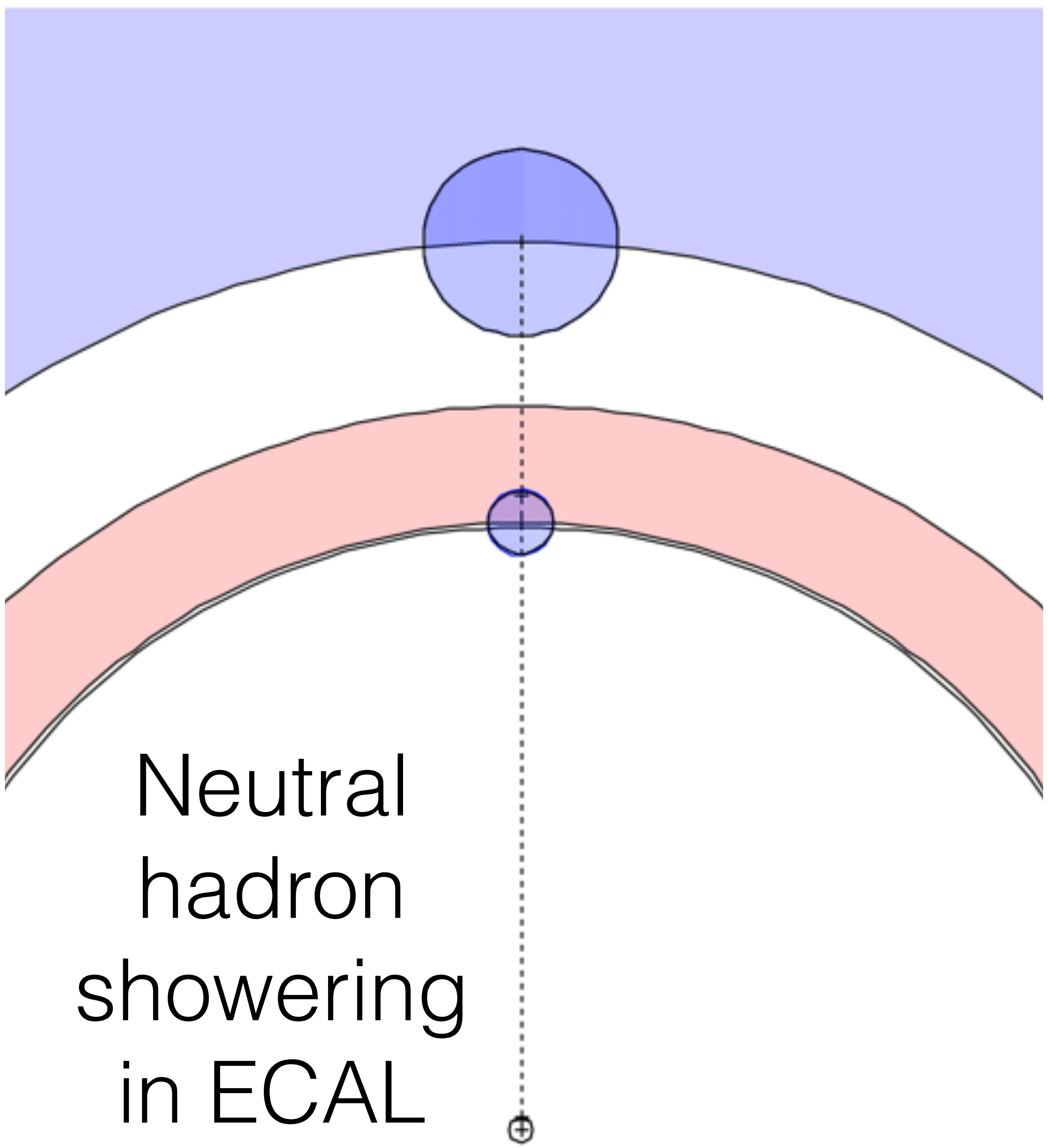
Particle flow: neutral hadron

Papas features a real particle flow algorithm ~ in CMS

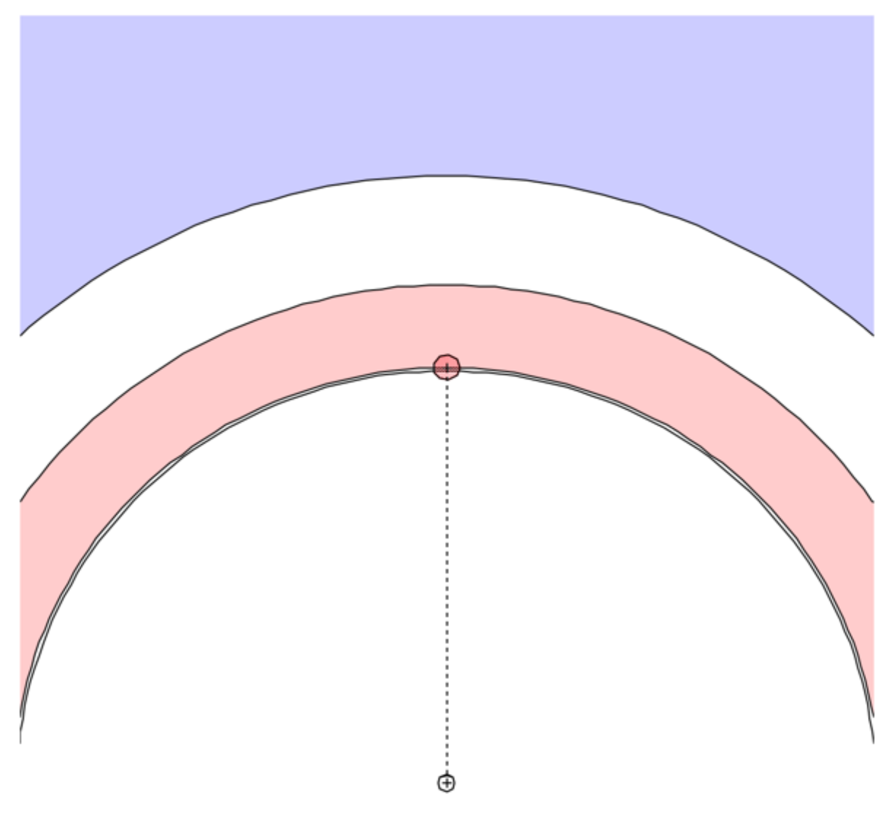


- HCAL cluster
  → Neutral hadron
  - energy and direction from the cluster
  - assume m_K0L
  - assume primary vertex origin

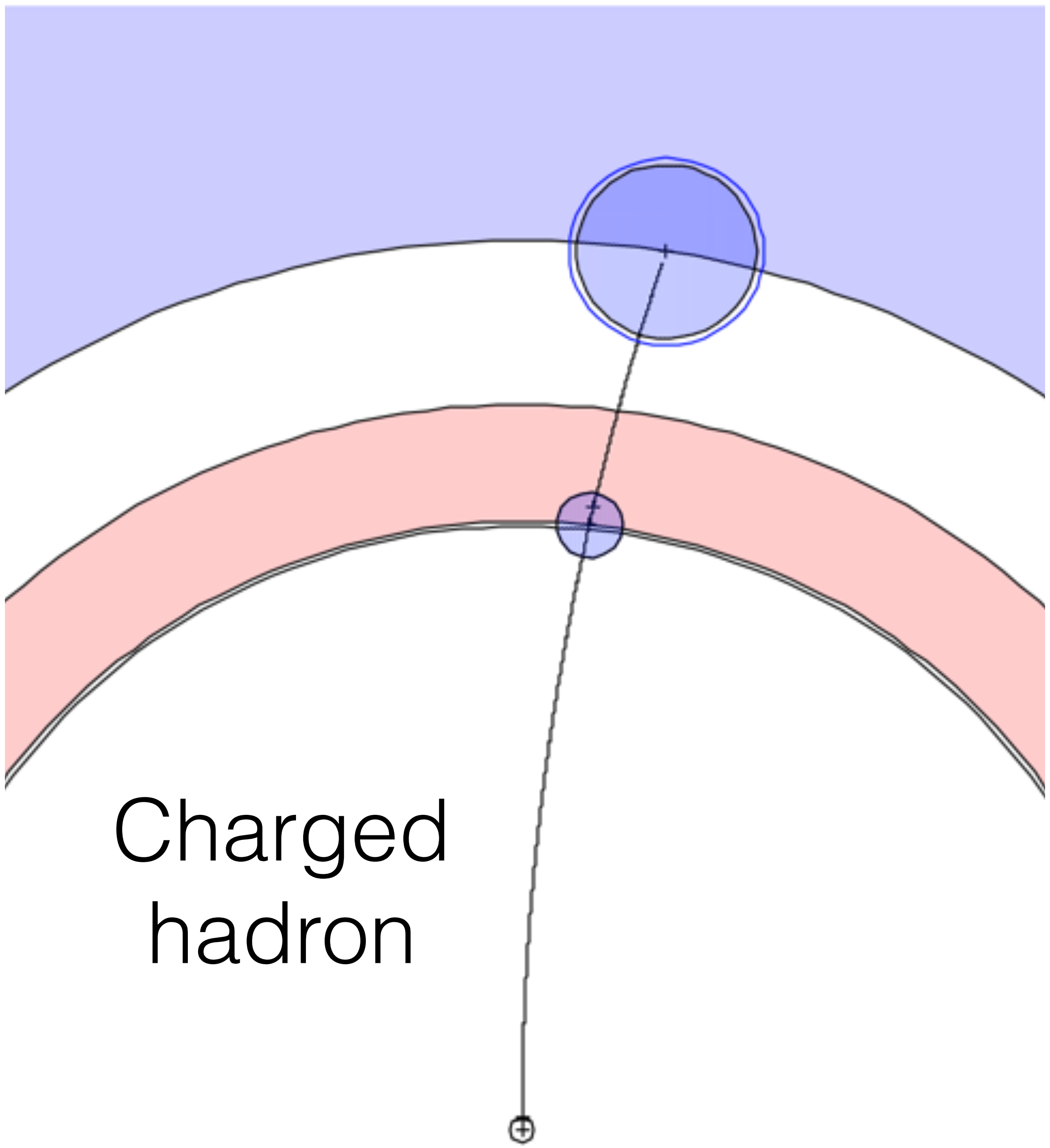
Particle flow: neutral hadron



- HCAL cluster
  → Neutral hadron
- ECAL cluster
  → Photon

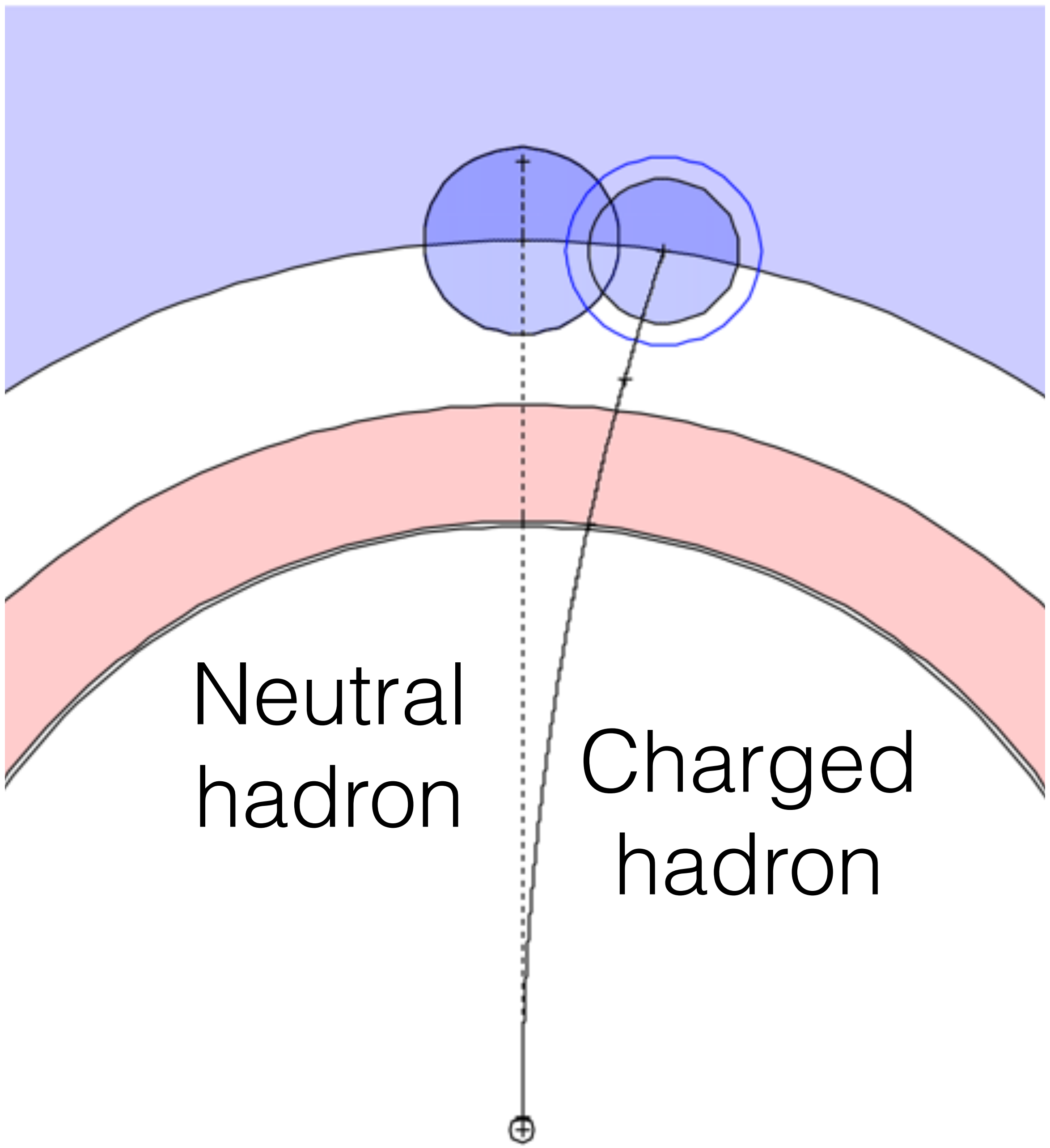


Particle flow: charged hadron



- Track
  → Charged hadron
  - momentum from the track
  - assume m_pi
  - clusters not used
- electrons and muons discussed later

Particle flow: charged + neutral



- Track
  → Charged hadron
  - momentum from the track
  - assume m_pi
- Merged cluster: unresolved neutral
- Significant energy excess in HCAL
  → Neutral hadron
  - calorimeter resolution for hadrons important if granularity is bad

e and μ

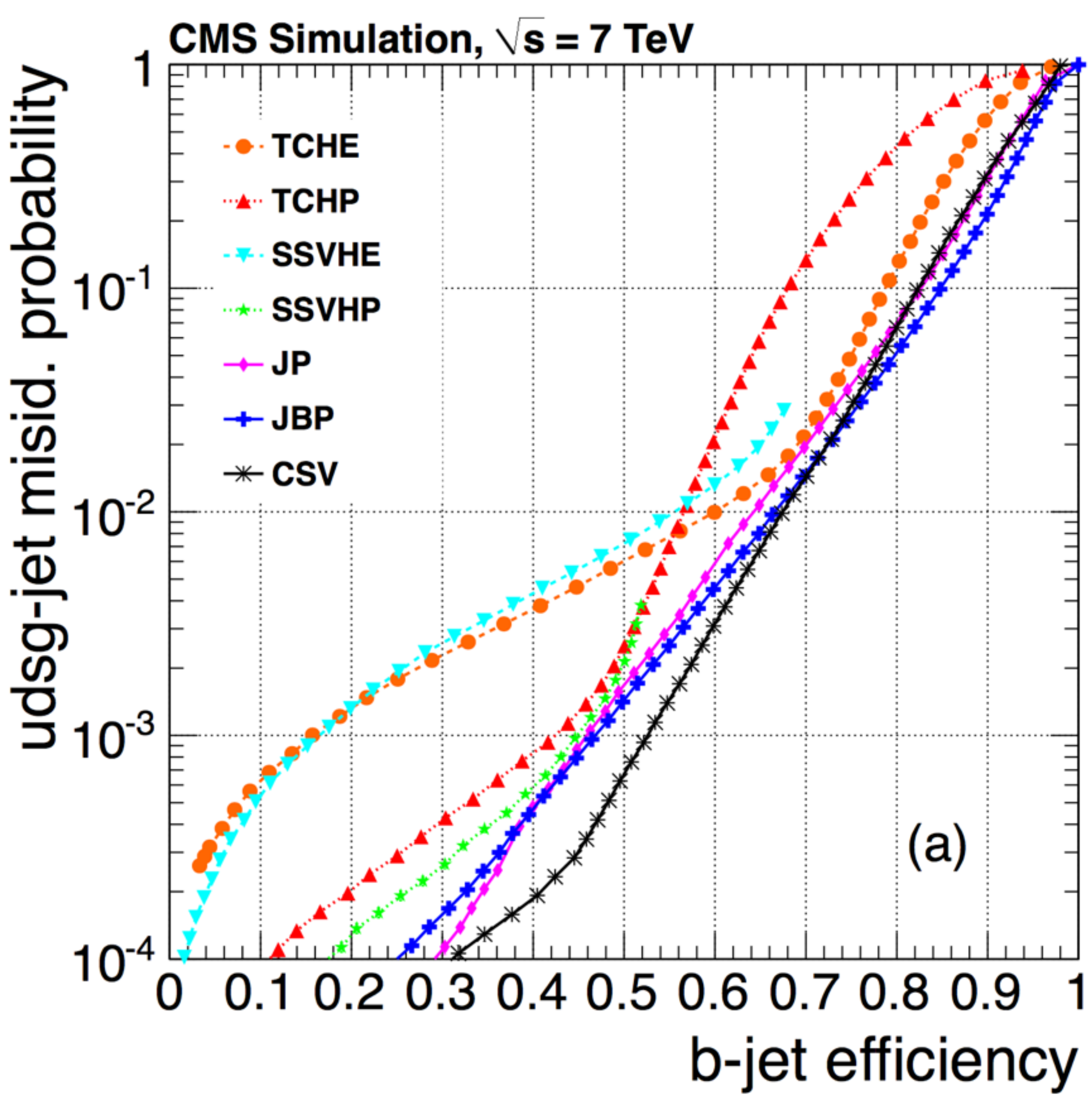
- In papas:
  - Momentum smearing
    - resolution parametrisation (e.g. vs pT, E, θ, …)
  - Acceptance
    - efficiency parametrisation

- Something more ambitious could be done
  - model material in tracker, track fit,
    particle-flow electron reconstruction

- Better go full sim?

Jets and tagging

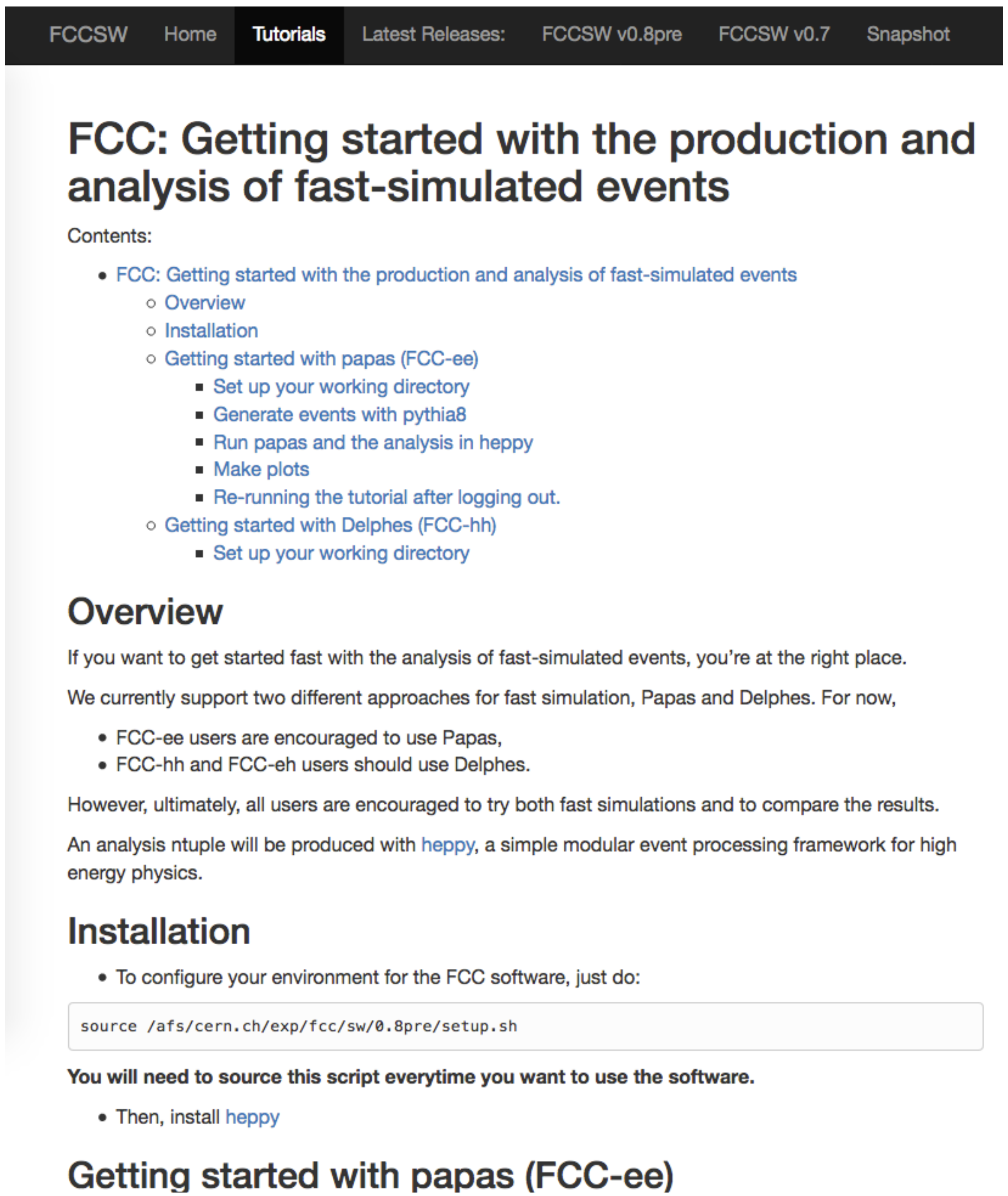
- papas runs fastjet
    - kT-ee exclusive, …
- Simple parametrization of the tagging performance
    - provide a few points on your ROC curve
    - works for b, c, τ, tagging



Identification of b-quark jets
with the CMS experiment
https://arxiv.org/abs/1211.4462

Getting started

http://fccsw.web.cern.ch/



FCCSW Home Tutorials Latest Releases: FCCSW v0.8pre FCCSW v0.7 Snapshot

FCC: Getting started with the production and analysis of fast-simulated events

Contents:

Overview

If you want to get started fast with the analysis of fast-simulated events, you're at the right place.

We currently support two different approaches for fast simulation, Papas and Delphes. For now,

- FCC-ee users are encouraged to use Papas,
- FCC-hh and FCC-eh users should use Delphes.

However, ultimately, all users are encouraged to try both fast simulations and to compare the results.

An analysis ntuple will be produced with heppy, a simple modular event processing framework for high energy physics.

Installation

- To configure your environment for the FCC software, just do:

```
source /afs/cern.ch/exp/fcc/sw/0.8pre/setup.sh
```

You will need to source this script everytime you want to use the software.

- Then, install heppy

Getting started with papas (FCC-ee)

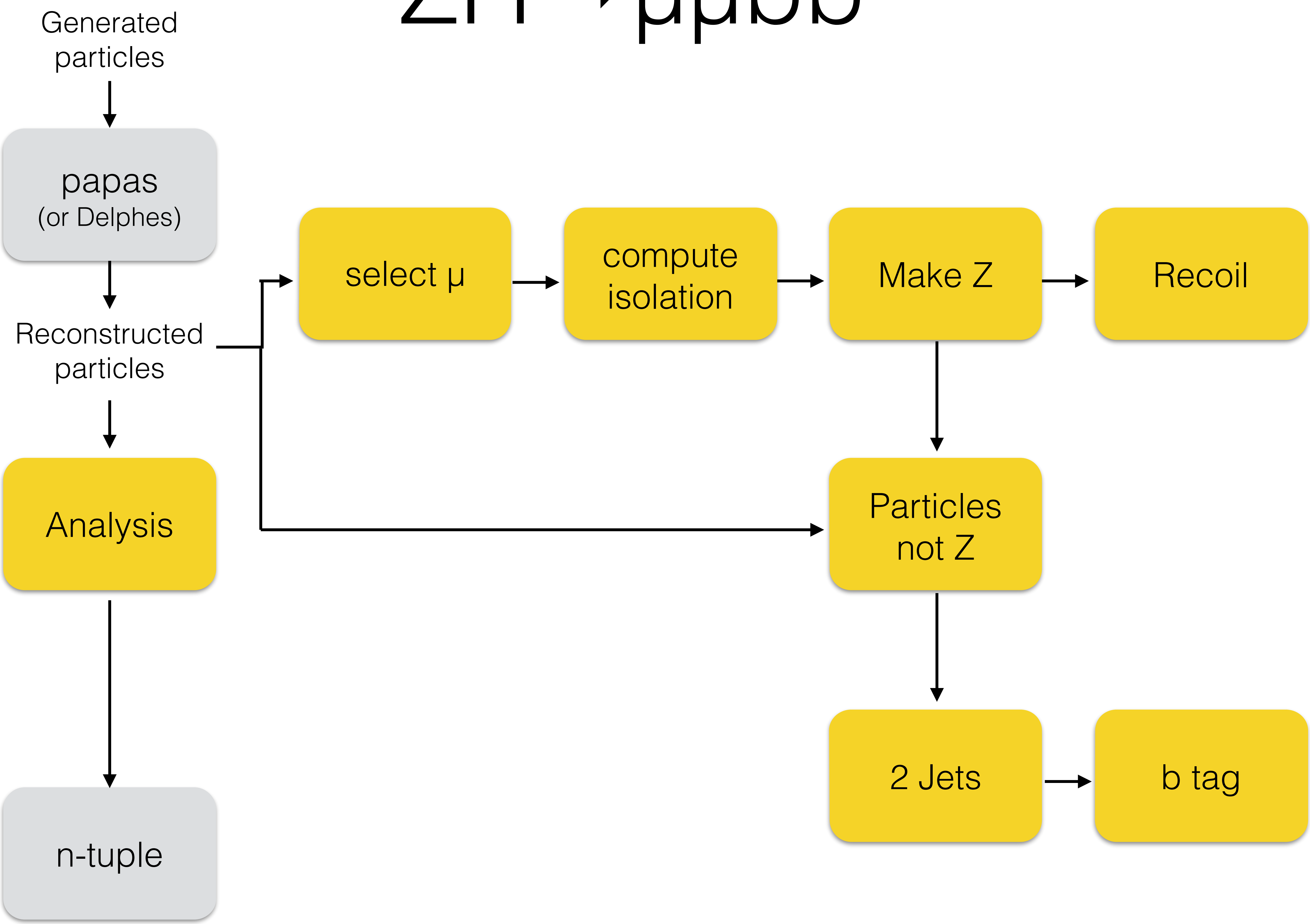
- FCC software documentation much improved recently
  - J. Lingemann and all
- Getting started page shows how to:
  - generate FCC EDM events with pythia
  - run papas
  - do analysis and get plots
  - how to run Delphes
    (FCC-hh, but applicable to FCC-ee as well)

Heppy

High Energy Physics in PYthon

- Event processing framework
- modular ~ CMSSW, Gaudi, Athena, Marlin
- written in python
- can read from root trees, CMS, FCC, LCIO event data models
- widely used in CMS
- many tools
    - batch processing, physics tools, …

ZH→μμbb



Generated particles
papas
(or Delphes)
Reconstructed particles
Analysis
n-tuple
select μ
compute isolation
Make Z
Recoil
Particles not Z
2 Jets
b tag

Get started: install

- Follow instructions from the « getting started » page on lxplus or SLC6 (1 min)

```
source /afs/cern.ch/exp/fcc/sw/0.8pre/setup.sh
git clone git@github.com:HEP-FCC/heppy.git
cd heppy/
source ./init.sh

mkdir Workdir
cd Workdir/
wget https://raw.githubusercontent.com/HEP-FCC/fcc-physics/master/pythia8/
ee_ZH_Zmumu_Hbb.txt
cp $HEPPY/test/analysis_ee_ZH_cfg.py .
```

Get started: run pythia

- generate 200k events with pythia8
  - 40 jobs, 5k events/job, 5 mins

```
pythia_batch.py  -o Pythia_sample/ ee_ZH_Zmumu_Hbb.txt 100
```

script is still private
please don't try it now :-)

The usual pythia8 card file

Get started: papas and analysis

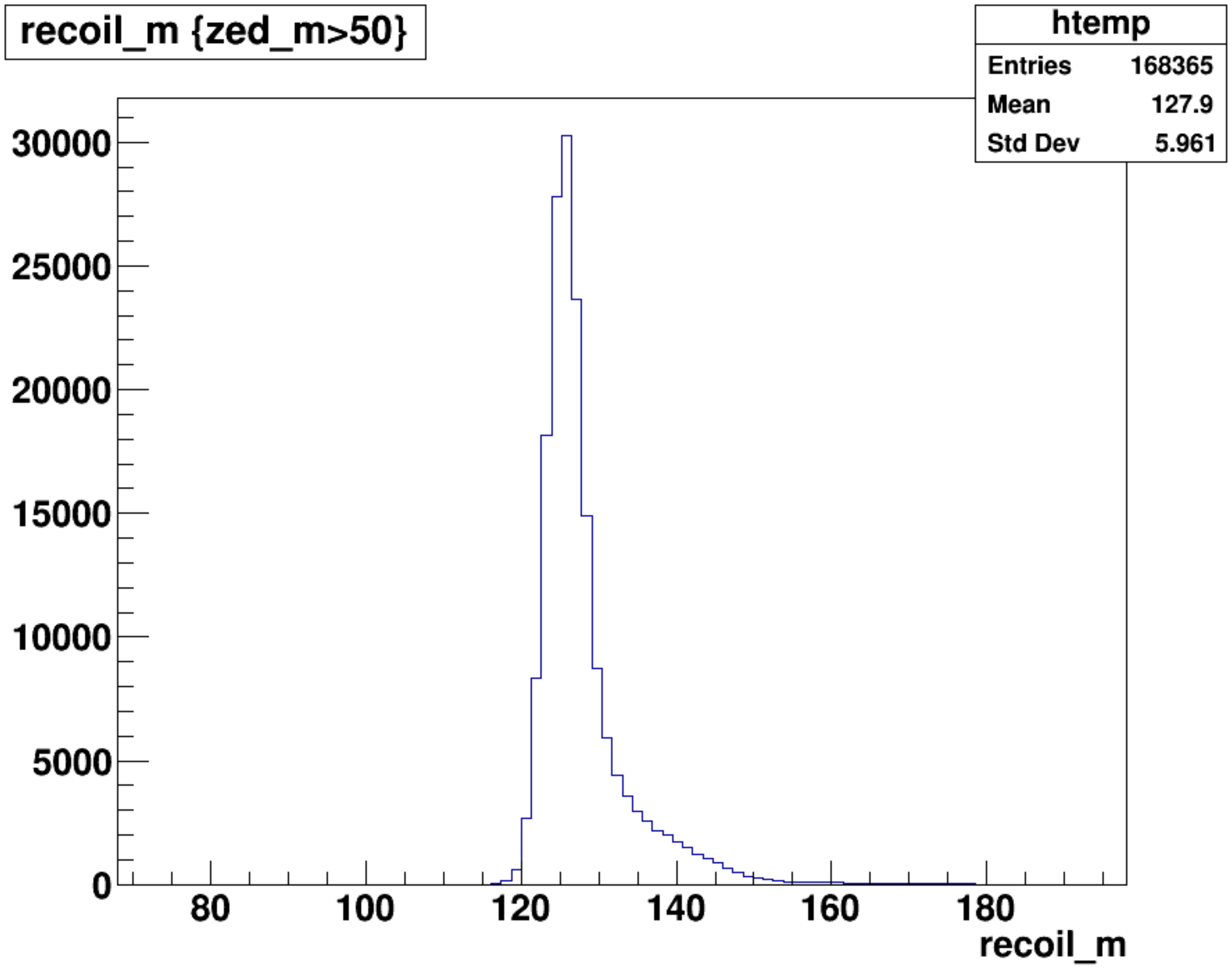
- Run papas and the analysis sequence on the CERN LSF cluster with heppy
  - 40 jobs, 5k events/job, ~ 15 mins

```
heppy_batch.py -o Ntuples/ analysis_ee_ZH_cfg.py \
                                   -b ‘bsub -q 8nh < batchScript.sh’
```

- Harvest the results (1 min)

```
cd Ntuples/
heppy_check.py *
heppy_hadd.py .
```

Make the plot




```
events->Draw(« recoil_m », « zed_m > 50 »)
```

Users in FCC-ee

- ttbar analysis and b tagging studies
  - Nicolò Foppiani, Patrizia Azzi
- Sterile neutrinos
  - Sissel Bay Nilsen, Mogens Dam
- W mass
  - Marina Beguin, Elizabeth Locci
- ZH → 4 jets
  - Katharina Behr, Krisztian Peters, + students
- ZH → νν bb; WW → H
  - Gaël Touquet, C.B.
- Contact me if you want to join
  - together with Michele Selvaggi if you want to use Delphes

Current activities

- C++ version of papas (A. Robson)
  - 10 times faster (now about 10 events/s in python)
  - Integration in fast/full simulation framework
- Cards for other detectors: ILD, FCC-ee(!)
  - also for Delphes (with M. Selvaggi)
- More example analyses
- Produce FCC EDM events with heppy/papas (B. Hegner)
  - share our samples
- More refined simulation?
  - preshower, material effects in inner tracker, track fit — or full sim
  - high-level reconstruction (e.g. τ identification)

Conclusion

- Papas: generic and detailed model of particle flow
  - detector parameter scan from CMS to ILC-like detectors
- Get your samples in < 1 day
  - heppy batch tools (LSF, eos)
- Compare to Delphes in particle flow mode,
  and use the full sim.
  - both fully integrated, several users already in FCC-hh,
    same analysis code can be used

Getting started

http://fccsw.web.cern.ch/

FCCSW | Home | Tutorials | Latest Releases: | FCCSW v0.8pre | FCCSW v0.7 | Snapshot

FCC: Getting started with the production and analysis of fast-simulated events

Contents:

Overview

If you want to get started fast with the analysis of fast-simulated events, you're at the right place.

We currently support two different approaches for fast simulation, Papas and Delphes. For now,

- FCC-ee users are encouraged to use Papas,
- FCC-hh and FCC-eh users should use Delphes.

However, ultimately, all users are encouraged to try both fast simulations and to compare the results.

An analysis ntuple will be produced with heppy, a simple modular event processing framework for high energy physics.

Installation

- To configure your environment for the FCC software, just do:

```
source /afs/cern.ch/exp/fcc/sw/0.8pre/setup.sh
```

You will need to source this script everytime you want to use the software.

- Then, install heppy

Getting started with papas (FCC-ee)

- FCC software e-group:
  - fcc-experiments-sw-dev
- You'll be very welcome at our meetings:
  - every other Wed, 11:00
- We would be happy to report in the detector WG, e.g.
  - fast/full simulation framework (Anna)
  - designing a full sim detector model in the FCCSW
  - pileup infrastructure (Valentin)