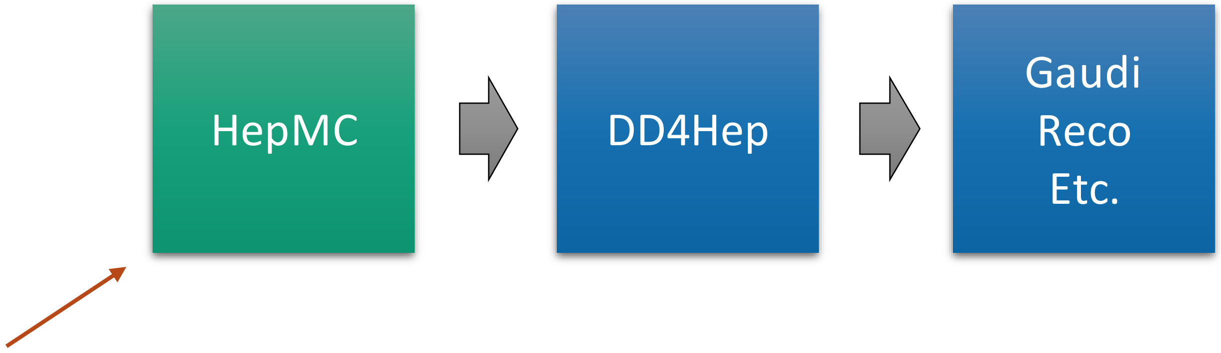
Afterburner implementation

Dmitry Romanov

Curret athena full simulation input

Strictly one format (HepMC) requirment:

- Higher testibility due to
- No input formats combinatorics
- Easier maintenance and further pipeline problems identification



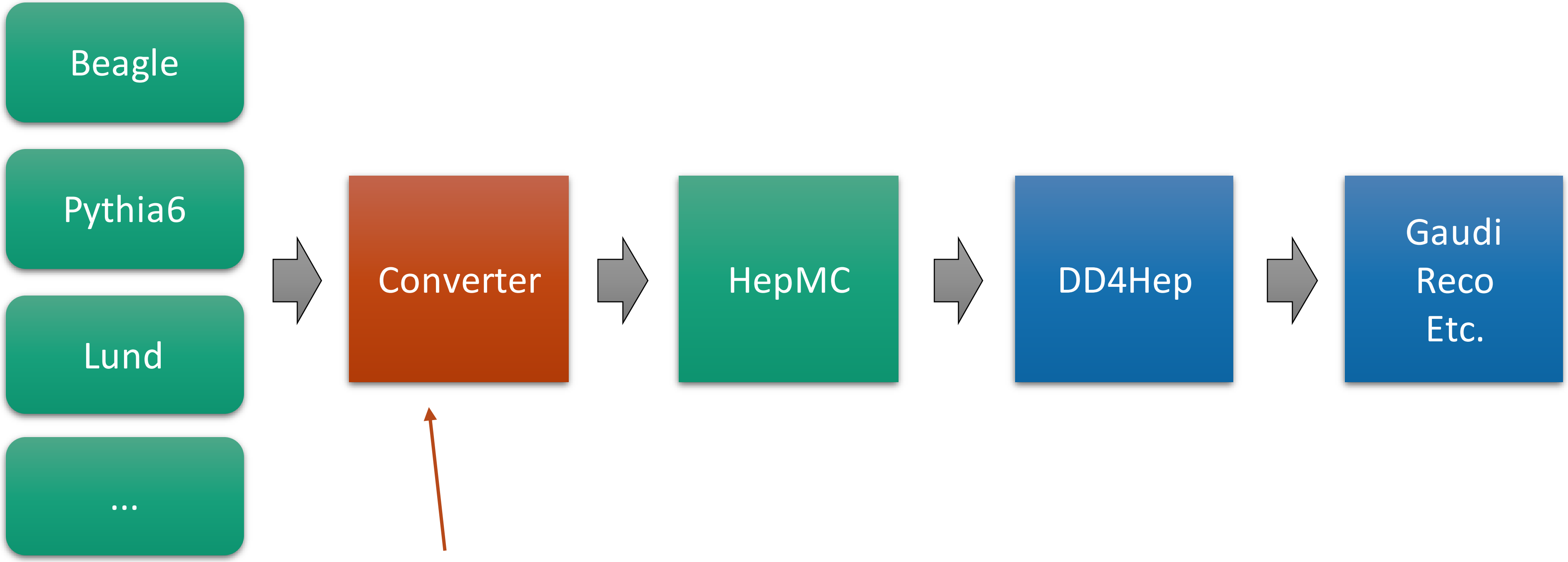
HepMC

DD4Hep

Gaudi
Reco
Etc.

Only one format here

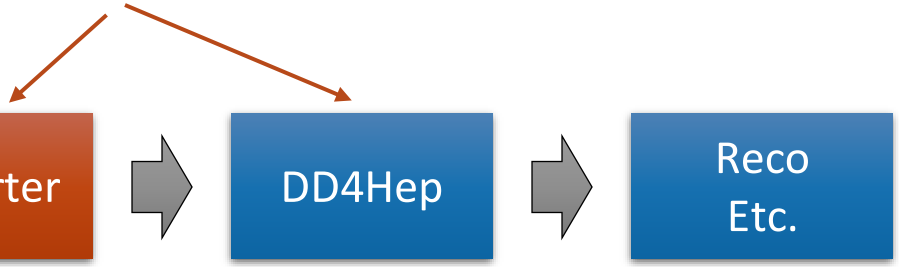
What about other MC formats?



Two converters currently exists:

- mcconv
- EicSmear

Places for afterburner

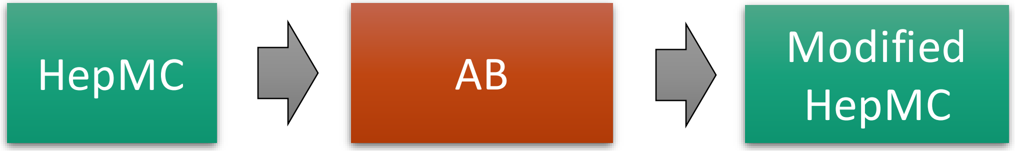


Converter

DD4Hep

Reco
Etc.

Afterburner as a standalone package



HepMC

AB

Modified
HepMC

Procs:

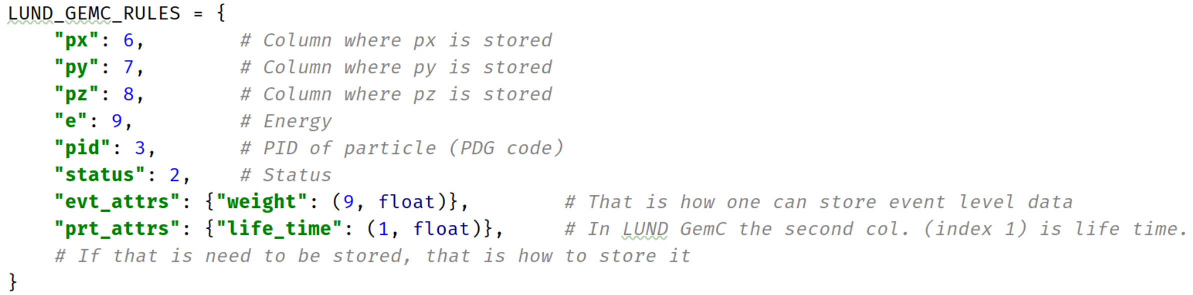
- Easy to test
- Software agnostic
- Simu. determenistic
- No main pipeline jeopardy
- Faster to implement

Cons:

- More file transforms
- Slower
- One more package

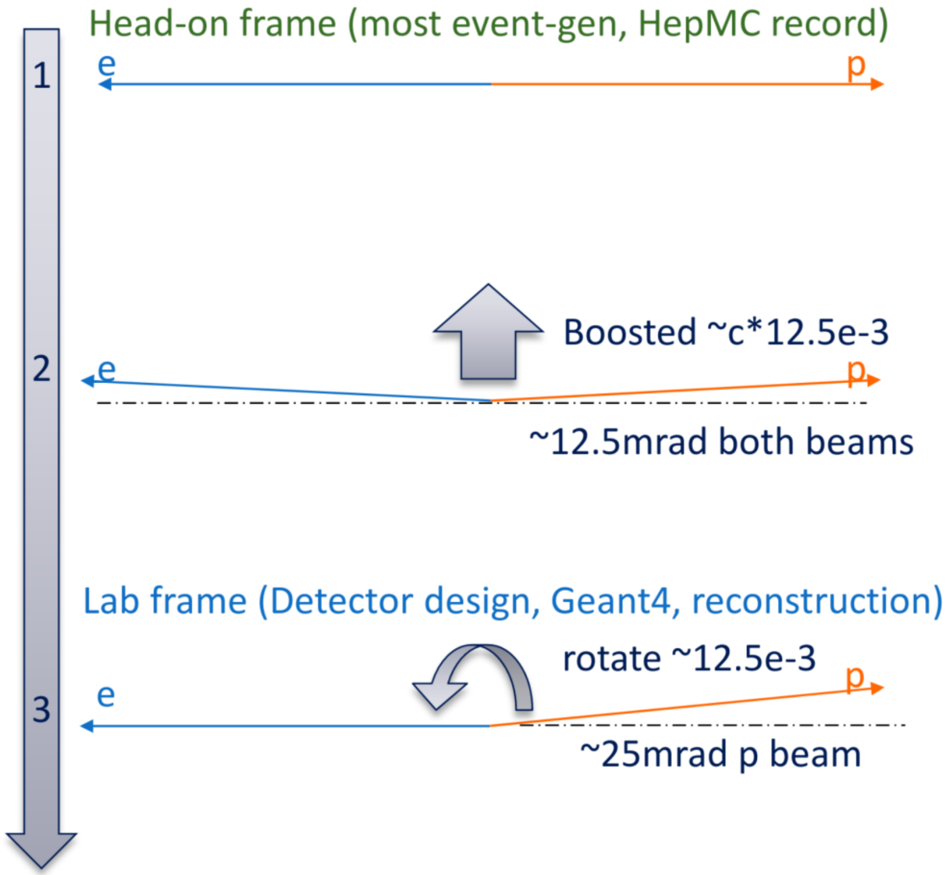
mcconv

- Written in python to
- Ensure high flexibility of configuration
- Easy access and interface for users:
  pip install mcconv
  mcconv myfile.txt
- Has boost, rotate, vtx move at hepmc level



```
LUND_GEMC_RULES = {
    "px": 6,        # Column where px is stored
    "py": 7,        # Column where py is stored
    "pz": 8,        # Column where pz is stored
    "e": 9,         # Energy
    "pid": 3,       # PID of particle (PDG code)
    "status": 2,    # Status
    "evt_attrs": {"weight": (9, float)},        # That is how one can store event level data
    "prt_attrs": {"life_time": (1, float)},     # In LUND GemC the second col. (index 1) is life time.
    # If that is need to be stored, that is how to store it
}
```

Fun4All



Head-on frame (most event-gen, HepMC record)

Boosted ~c*12.5e-3

~12.5mrad both beams

Lab frame (Detector design, Geant4, reconstruction)

rotate ~12.5e-3

~25mrad p beam

Pythia8

Values to be set:

ΔPx_A ΔPy_A ΔPz_A

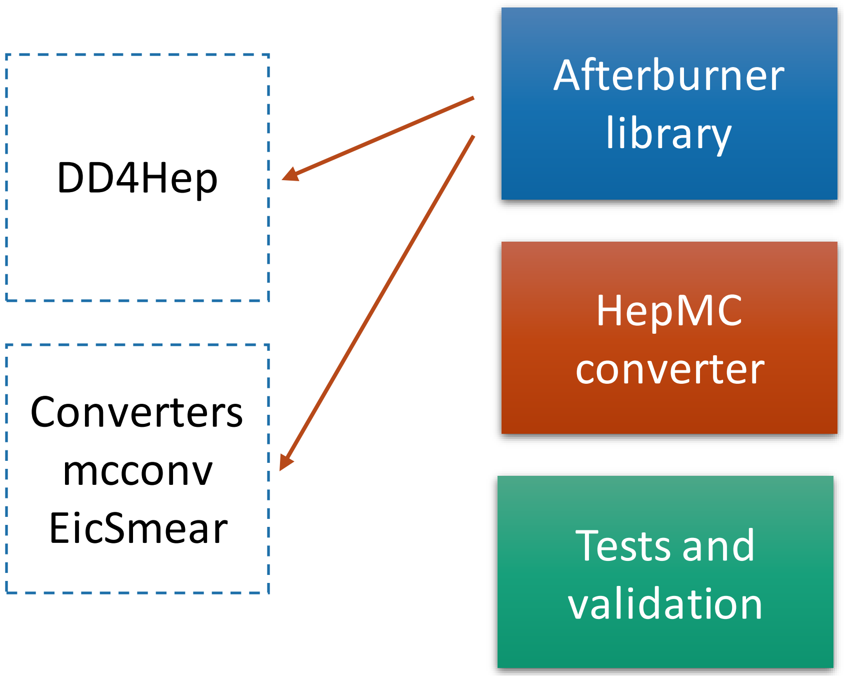
ΔPx_B ΔPy_B ΔPz_B

Vertex:

X Y Z T

https://raw.githubusercontent.com/eic/documents/master/reports/general/Note-Simulations-BeamEffects.pdf

Desired afterburner implementation



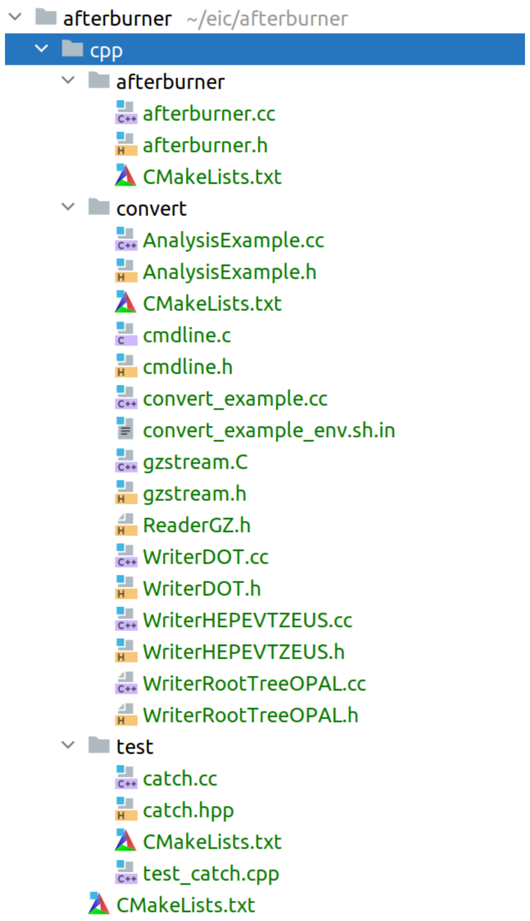
DD4Hep

Converters
mcconv
EicSmear

Afterburner library

HepMC converter

Tests and validation



```
afterburner  ~/eic/afterburner
  cpp
    afterburner
      afterburner.cc
      afterburner.h
      CMakeLists.txt
    convert
      AnalysisExample.cc
      AnalysisExample.h
      CMakeLists.txt
      cmdline.c
      cmdline.h
      convert_example.cc
      convert_example_env.sh.in
      gzstream.C
      gzstream.h
      ReaderGZ.h
      WriterDOT.cc
      WriterDOT.h
      WriterHEPEVTZEUS.cc
      WriterHEPEVTZEUS.h
      WriterRootTreeOPAL.cc
      WriterRootTreeOPAL.h
    test
      catch.cc
      catch.hpp
      CMakeLists.txt
      test_catch.cpp
    CMakeLists.txt
```

What is needed?

- Transformation procedure defined

- Definition of validation and tests