

HepMC Interface & nanoAODs

Alice Offline Week
November 2014

HepMC Interface

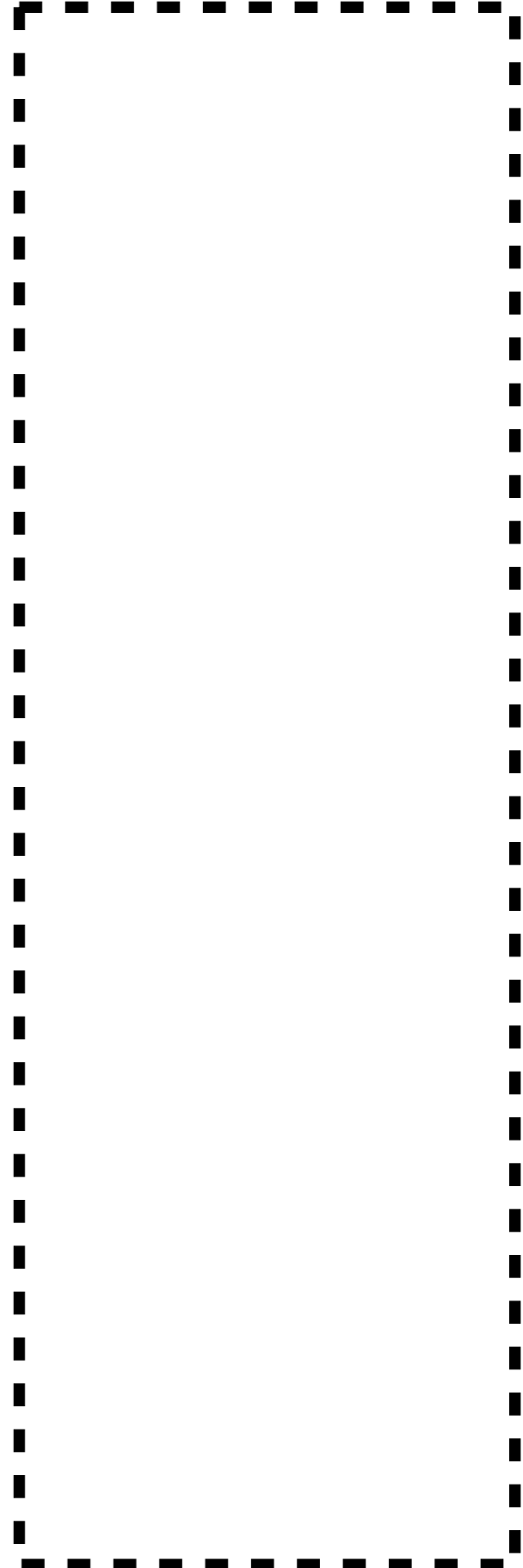
- **Why?**

- A lot (all?) **modern MC** generator produce an **output** in HepMC format
 - EPOS (CRMC), JEWEL, Pythia, Herwig++, Sherpa
- Implementing an interface for those generators automatically supports all of them!

- **Development**

- Most of the development done by Brian Thorsbro (Summer Student 2014)
 - Slides: <https://indico.cern.ch/event/291328/contribution/1>
 - Report: <http://cds.cern.ch/record/1751399?ln=en>
- A lot of help from Alina, in preparing the grid package and debugging grid issues

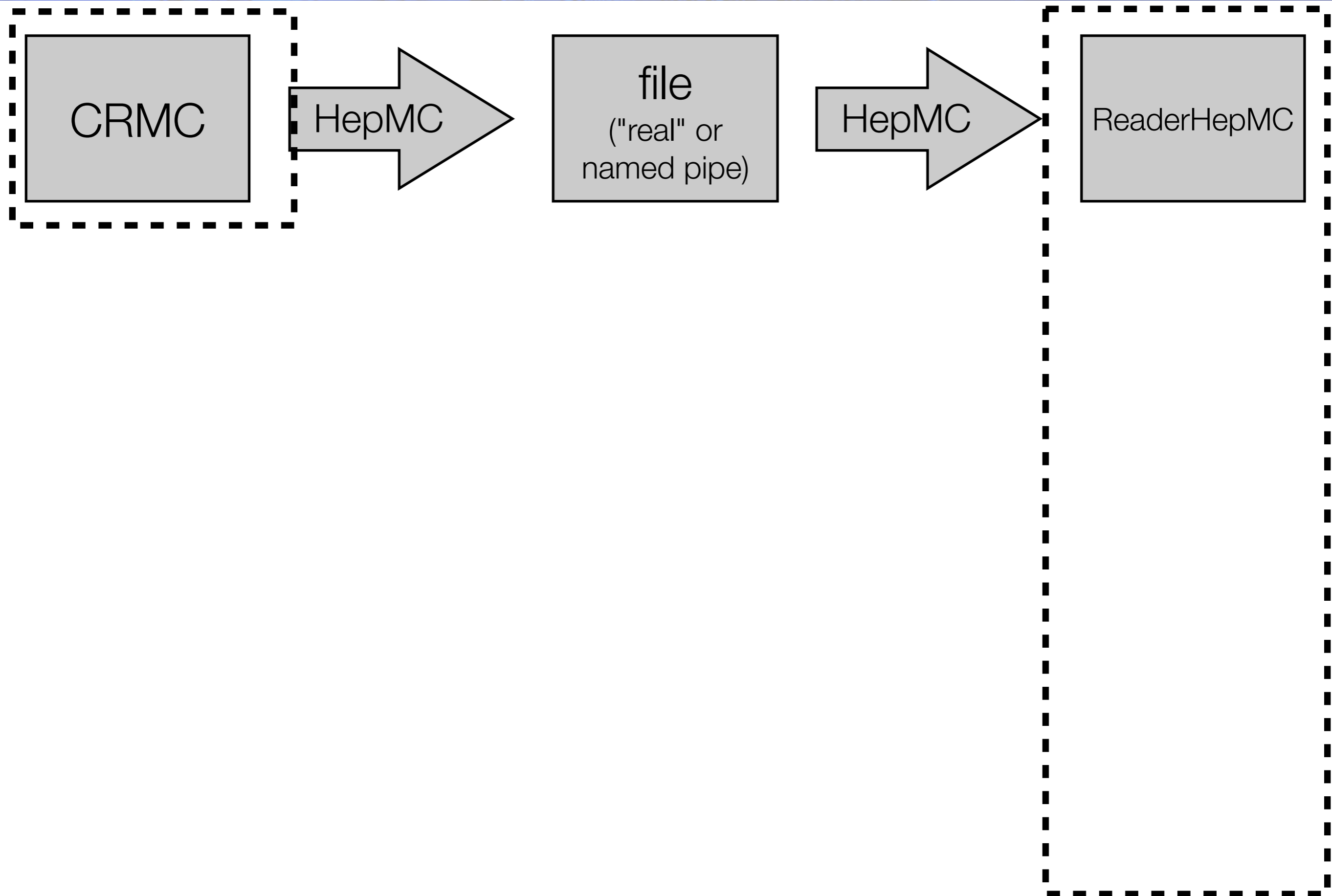
Work Flow



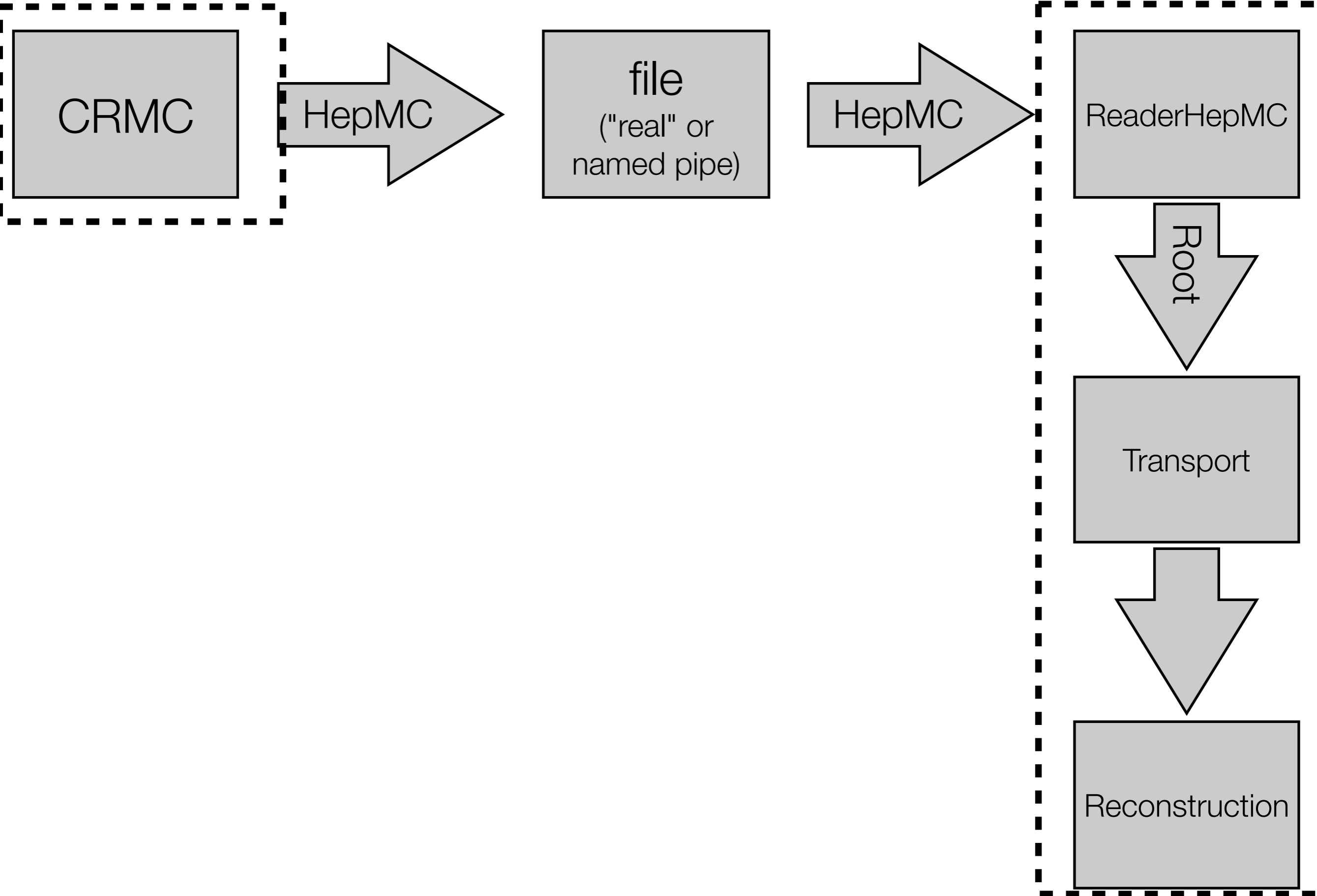
CRMC

HepMC

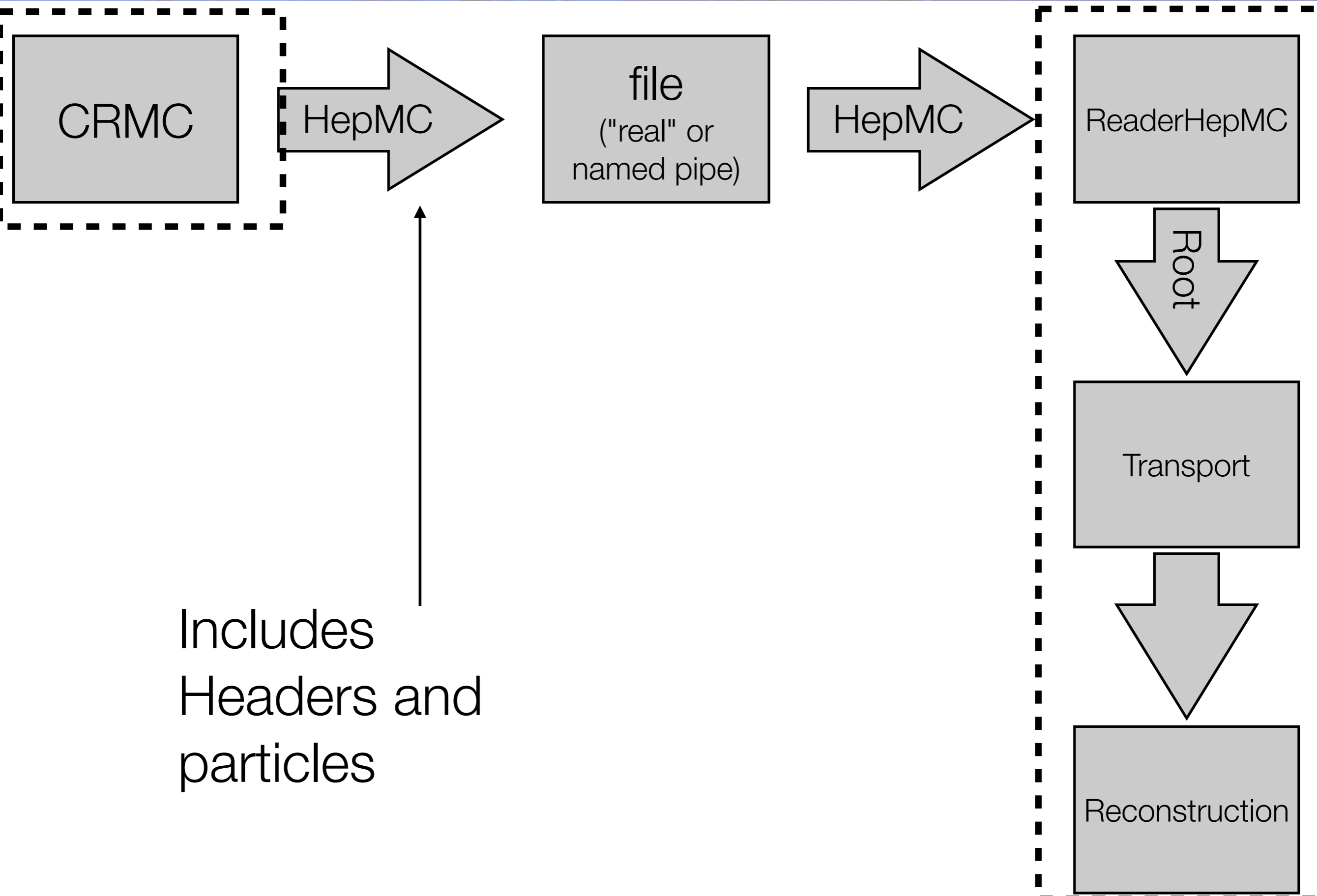
A flow diagram showing a grey rectangular box labeled "CRMC" on the left. A grey arrow labeled "HepMC" points from the right side of the "CRMC" box to the right. The "CRMC" box and the arrow are enclosed within a dashed black rectangular border.



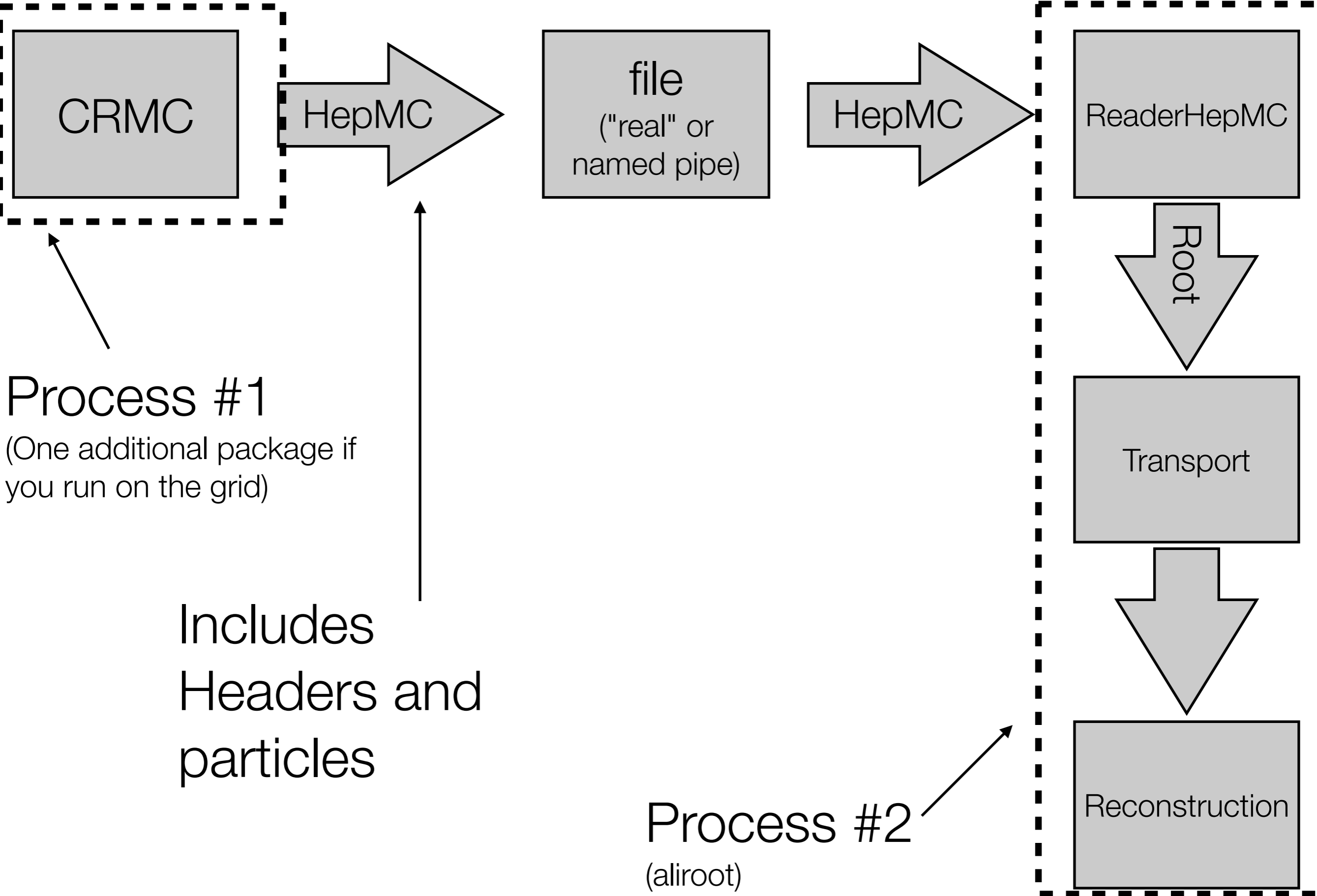
Work Flow

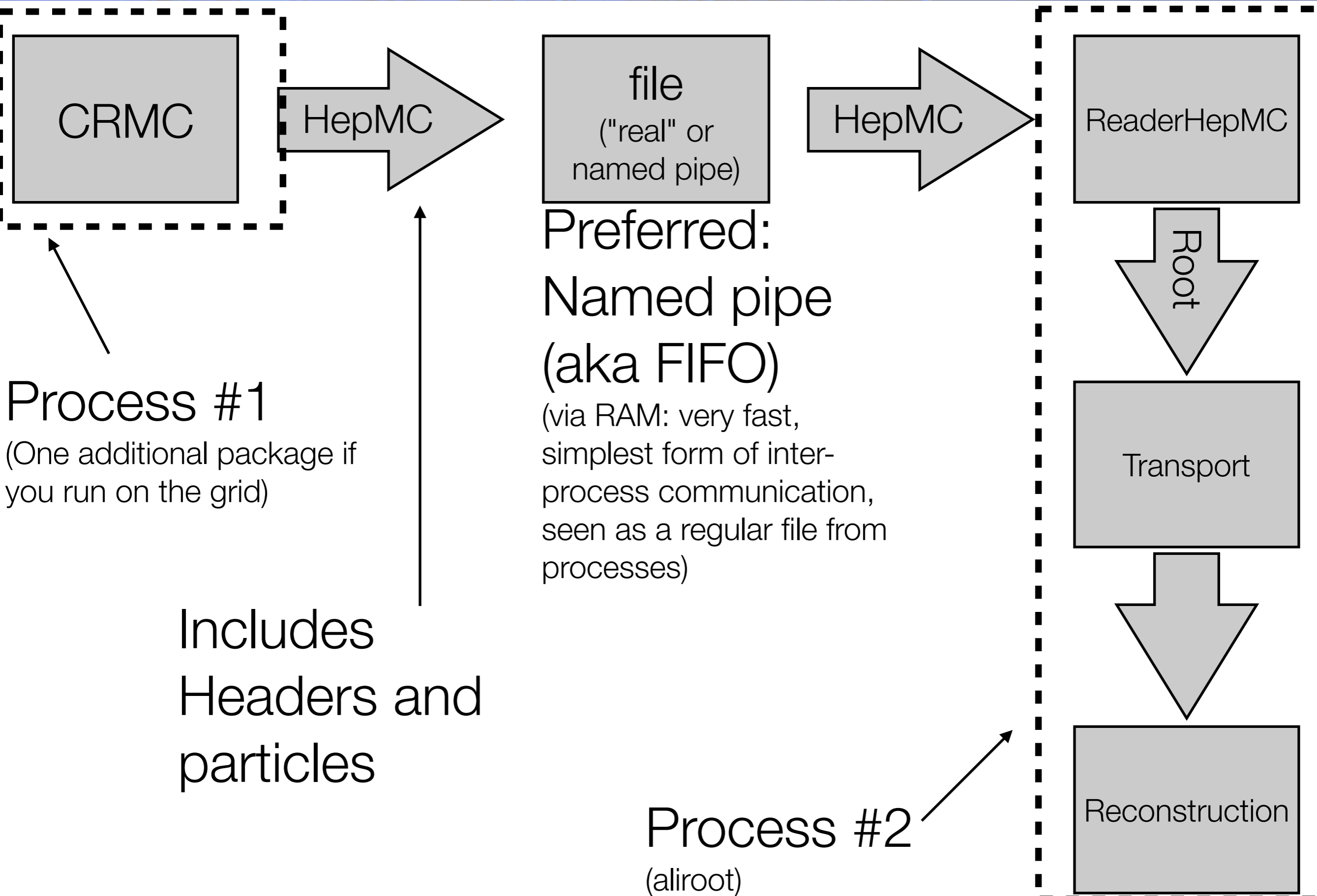


Work Flow



Includes
Headers and
particles





JDL

```
Packages = {  
  "VO_ALICE@CRMC::v1.4",  
  "VO_ALICE@AliRoot::vAN-20141111",  
  "VO_ALICE@APISCONFIG::V1.1x",  
  "VO_ALICE@fastjet::v3.0.6_1.012",  
  "VO_ALICE@jemalloc::v3.0.0"  
};
```

1 additional
package

Config.C

```
AliGenerator* GenReader() {  
  AliGenReaderHepMC *reader =  
    new AliGenReaderHepMC();  
  reader->SetFileName("crmceventfifo");  
  AliGenExtFile *gener = new  
  AliGenExtFile(-1);  
  gener->SetReader(reader);  
  return gener;  
}
```

HepMC reader
as the
generator

simrun.sh

```
#make fifo and run crmc in the bg  
mkfifo crmceventfifo  
crmc -t-c crmc.param -f crmceventfifo -o hepmc \  
-p3500 -P-3500 -n200 -m0 &
```

External
generator
(number of events must be
synced with sim.C)

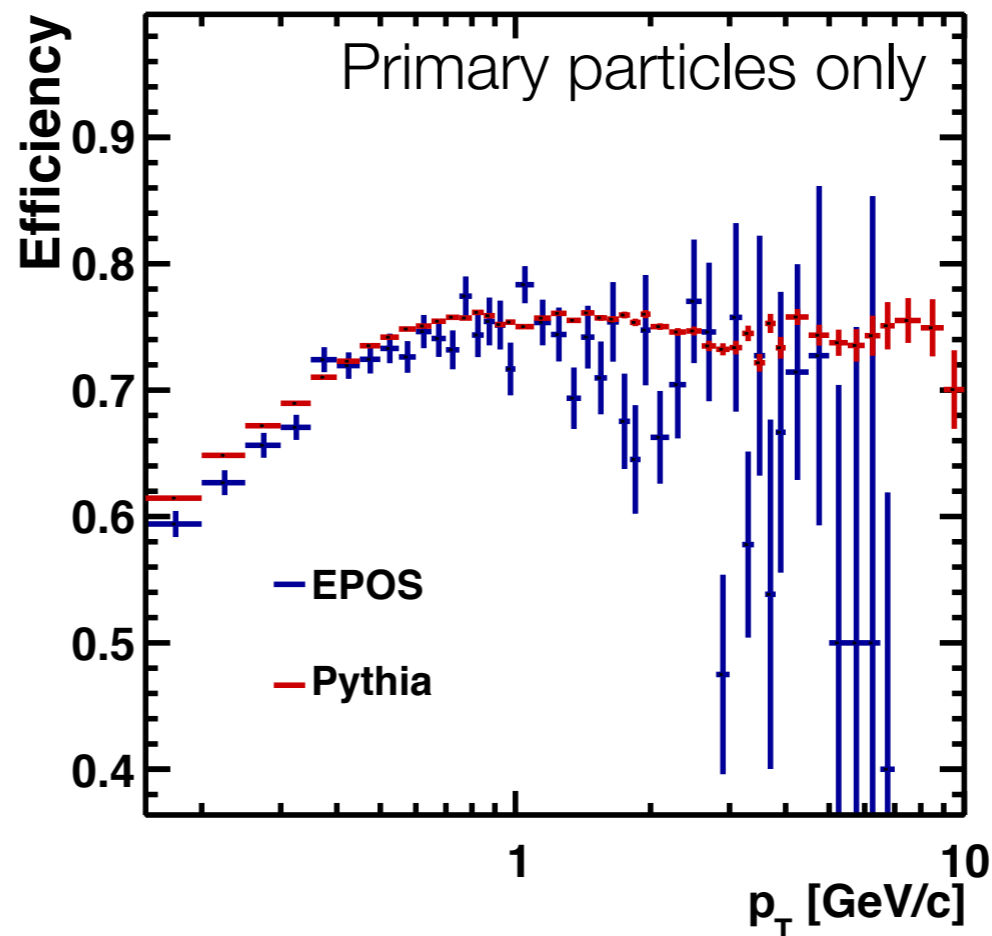
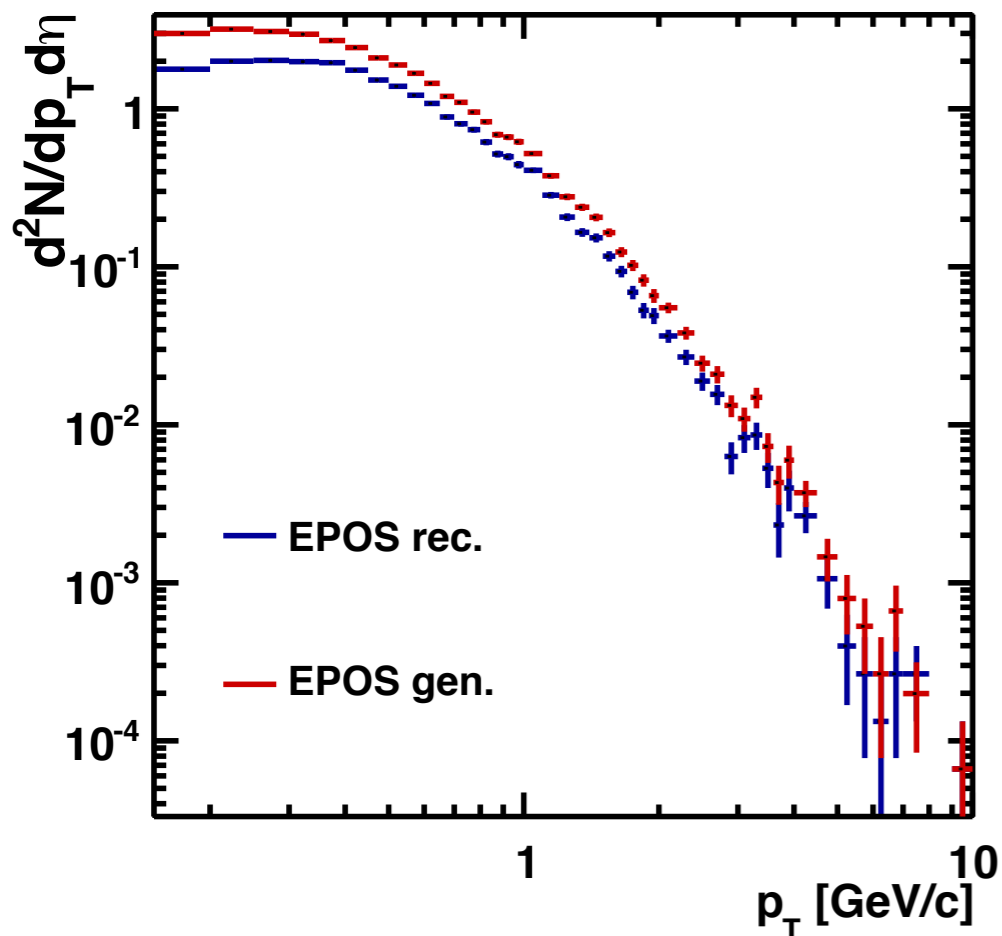
Small test production

pid	owner	first seen	last seen	subjobs	DONE		ERRORS	
	mfloris (383)				58%	58	42%	42
443124529	mfloris	17.11.2014 08:49	17.11.2014 16:18	100				

20 K EPOS events requested (pp at $\sqrt{s} = 7$ TeV, anc. to LHC10d)
 40% fail (TTL: 200 evts/chunk)

Output in `alien:///alice/cern.ch/user/m/mfloris/TestHepMC/TestPPLHC10d/output`

Quick analysis by Vytautas Vislavicious



Next:

- Port changes to release
- Run an official production
- Test other generators

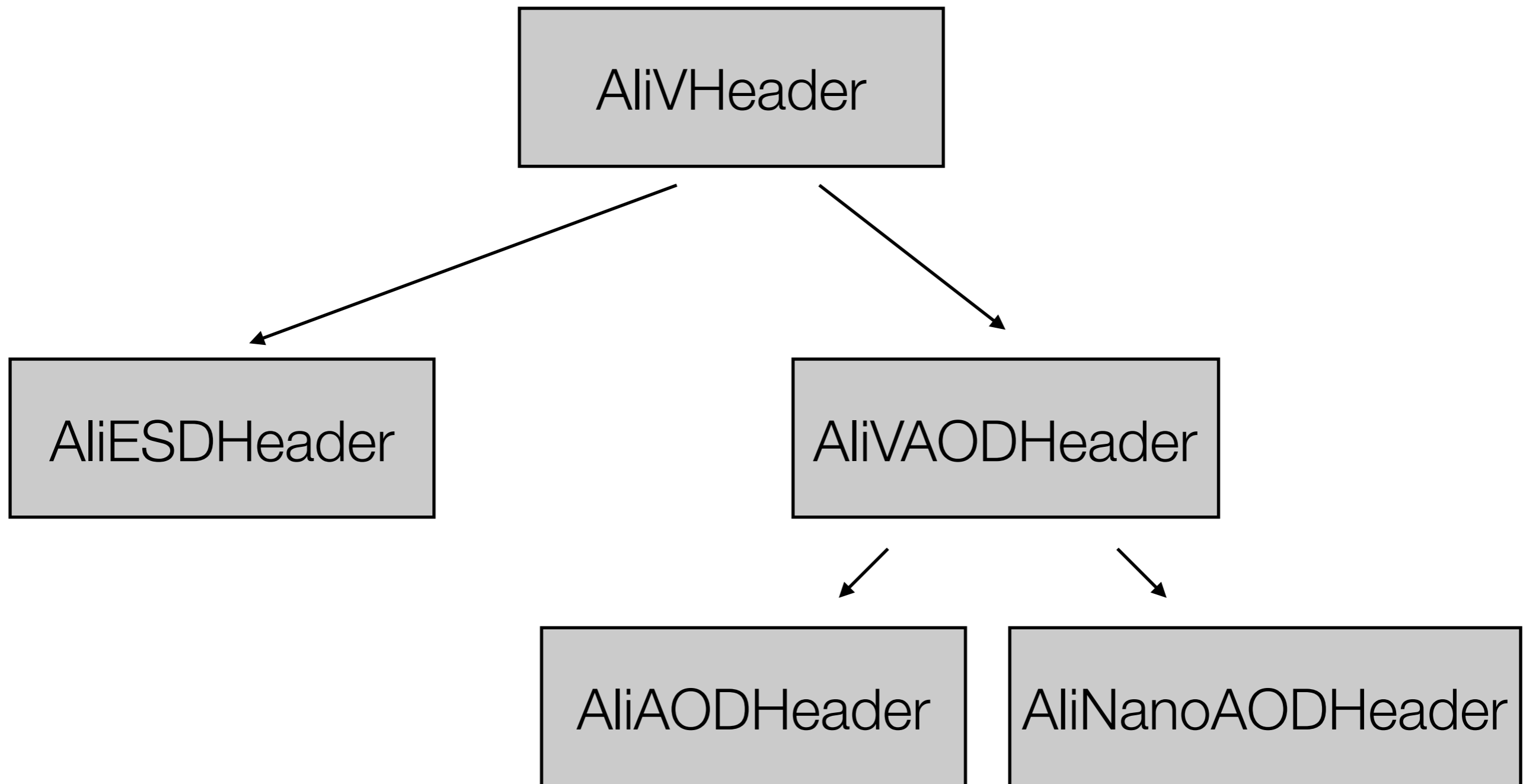
nanoAOD

- Clean up of **Events, Tracks, Vertices** classes
 - **Homogeneous** ESD/AOD interfaces (as much as possible)
 - Push interface back to the **virtual classes**
 - Benefit for nanoAOD vs AOD tasks
 - **Clarify** meaning if ambiguous members/getters (comments)
 - Remove **duplications**
- Required changes to [Virtual,ESD,AOD][Events,Tracks,Vertices]
- Will have to be reconsidered when we switch to a flat AOD structure
- Changes **implemented** in the NanoAODdev branch (A. Festanti, R. Russo, R. Shahoyan, M. Floris), for details: JIRA [PWGPP-9](#)
- Merged on 27/10 in master

- AODs now contain **2 types** of tracks/headers:
 - AliAODTrack, AliAODHeader
 - AliNanoAODTrack, AliNanoAODHeaderBoth derive from the corresponding AliV classes
- **Proposal** (March 2014):
Change AliAOD{Track,Header}* to AliV{Track,Header}* in **AliAODEvent**
 - Required patching some analysis tasks and framework classes
 - If a user can rely on the V track interface, no changes are needed in the task between processing AOD or nanoAOD format
 - If one needs specific methods, the tracks will have to be explicitly recasted.
 - Problem with header found during implementation (next slide)
- **Implementation**
 - Tasks using AOD tracks / headers automatically patched to add a dynamic cast to AliAODtracks
 - Changes merged into master on 27/10/2014

```
AliAODTrack *aodtrack =dynamic_cast<AliAODTrack*>(event->GetTrack(i));  
if(!aodtrack) AliFatal("Not processing a standard AOD");
```

- Added AliVAODHeader, because AOD interface very different wrt ESD/Virtual
- Possibility to make interface more uniform will be considered in the future



- Complete **AliNanoAODHeader**
 - Only most basic methods are implemented
- Develop **PID interface**
 - Basic requirement: maintain **compatibility with PID framework**
 - Idea:
 - **AliDetectorPID** a transient object used to cache calibrated PID information in AliAODtracks
 - Make it non-transient in AliNanoAODTracks
 - Only store the PID information which is actually requested in this object
- **Clean up** (coverity etc)
- Use it!
- **Development** has been **slow**: any **help** to speed it up would be appreciated
 - No show-stopper, just need time fore development/tests