ALICE Report on the Use of VMC

Andreas Morsch
CERN

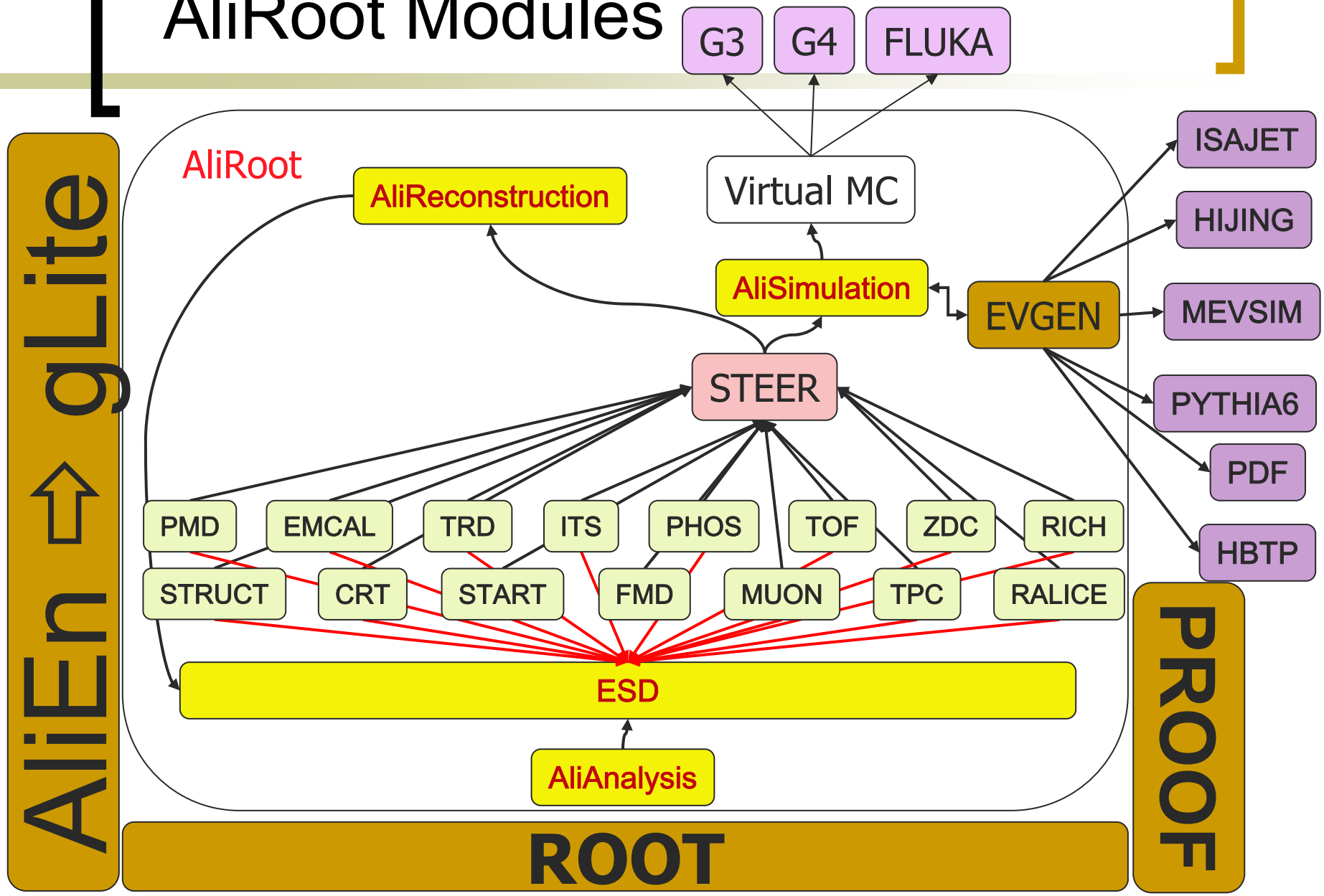
Virtual MC Workshop
CERN 29/11/2004

Introduction

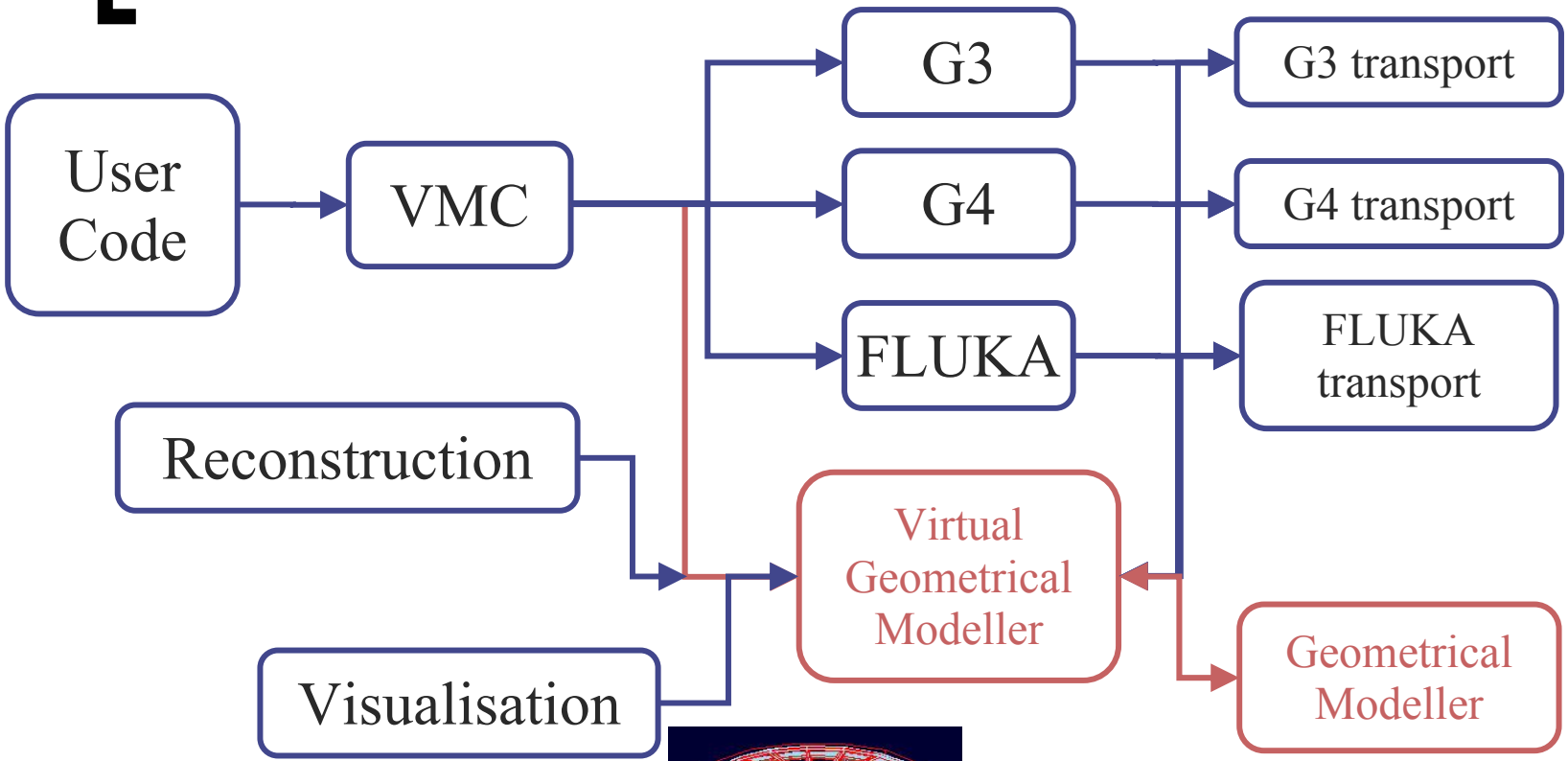
- The concept of the Virtual Monte Carlo (VMC) has been gradually developed by the ALICE software project
  - First implemented as a specific wrapper to GEANT3, *TGeant3*, to allow smooth migration from FORTRAN based simulation to a C++ framework avoiding the parallel development of two framework.
  - The requirement to reuse the user simulation code when ALICE moves from GEANT3 to FLUKA or GEANT4 based simulation lead to an abstraction of *TGeant3*.
  - Separation of VMC and ALICE specific code allowed to move the interface to ROOT and to make it available to non-ALICE users.

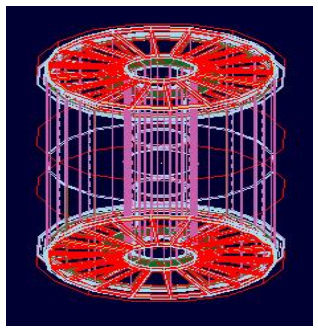
- The ALICE simulation framework is bases on VMC, generator interfaces and ROOT
- ALICE still contributes to VMC. Recently mainly through the implementation of *TFluka*.

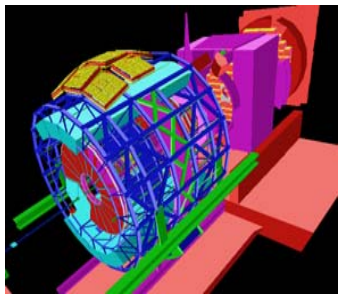
AliRoot Modules



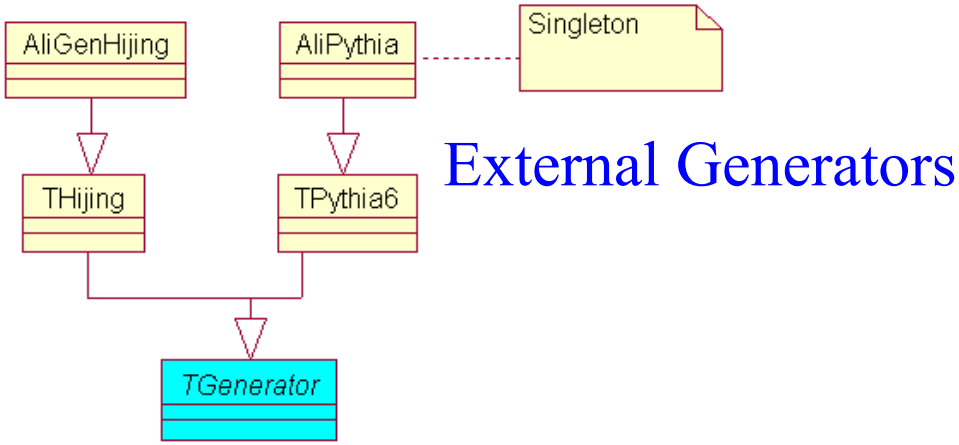
VMC + *TGeo*


User Code
VMC
G3
G4
FLUKA
G3 transport
G4 transport
FLUKA transport
Reconstruction
Visualisation
Virtual Geometrical Modeller
Geometrical Modeller

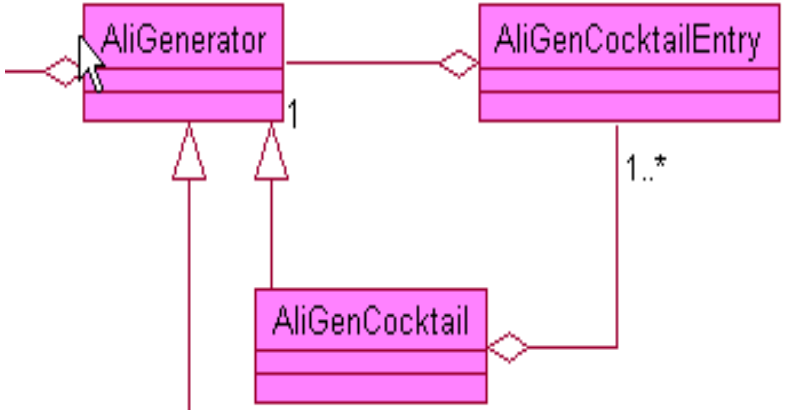




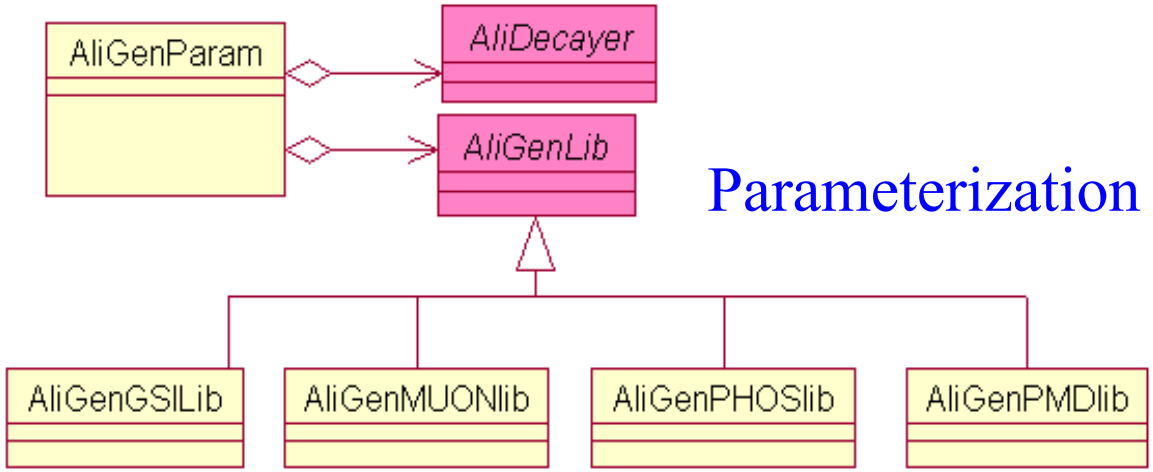
Event Generator Interfaces



External Generators

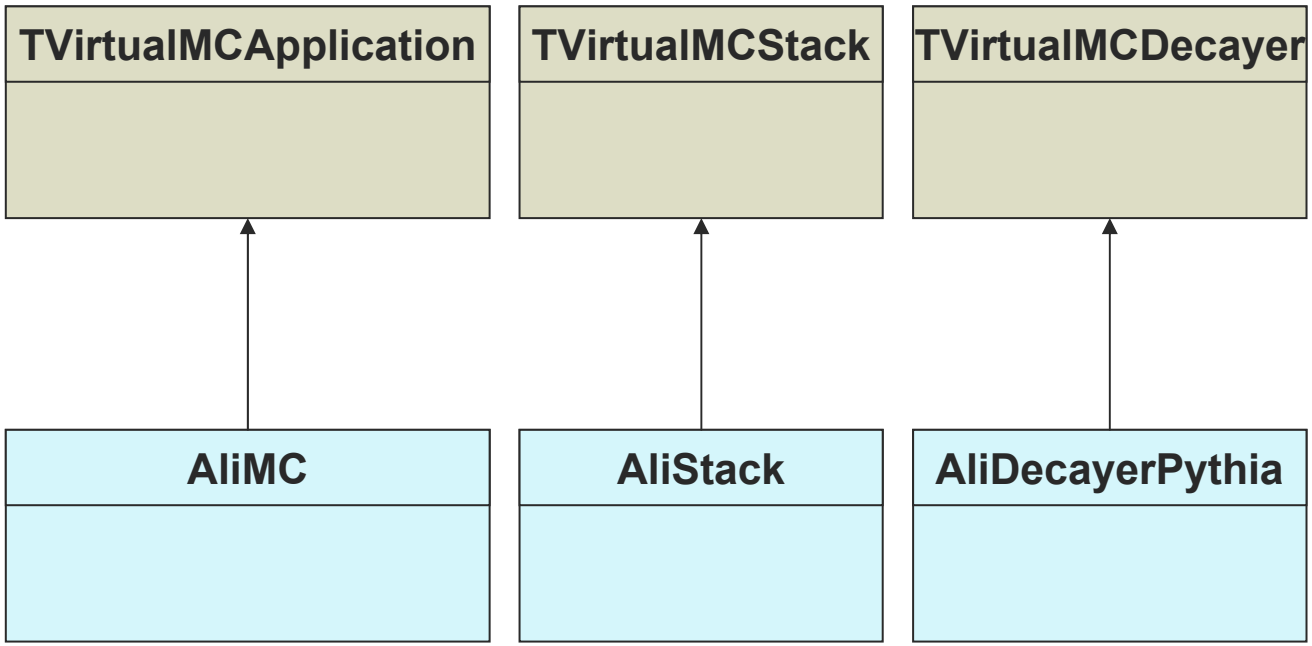


Event cocktails

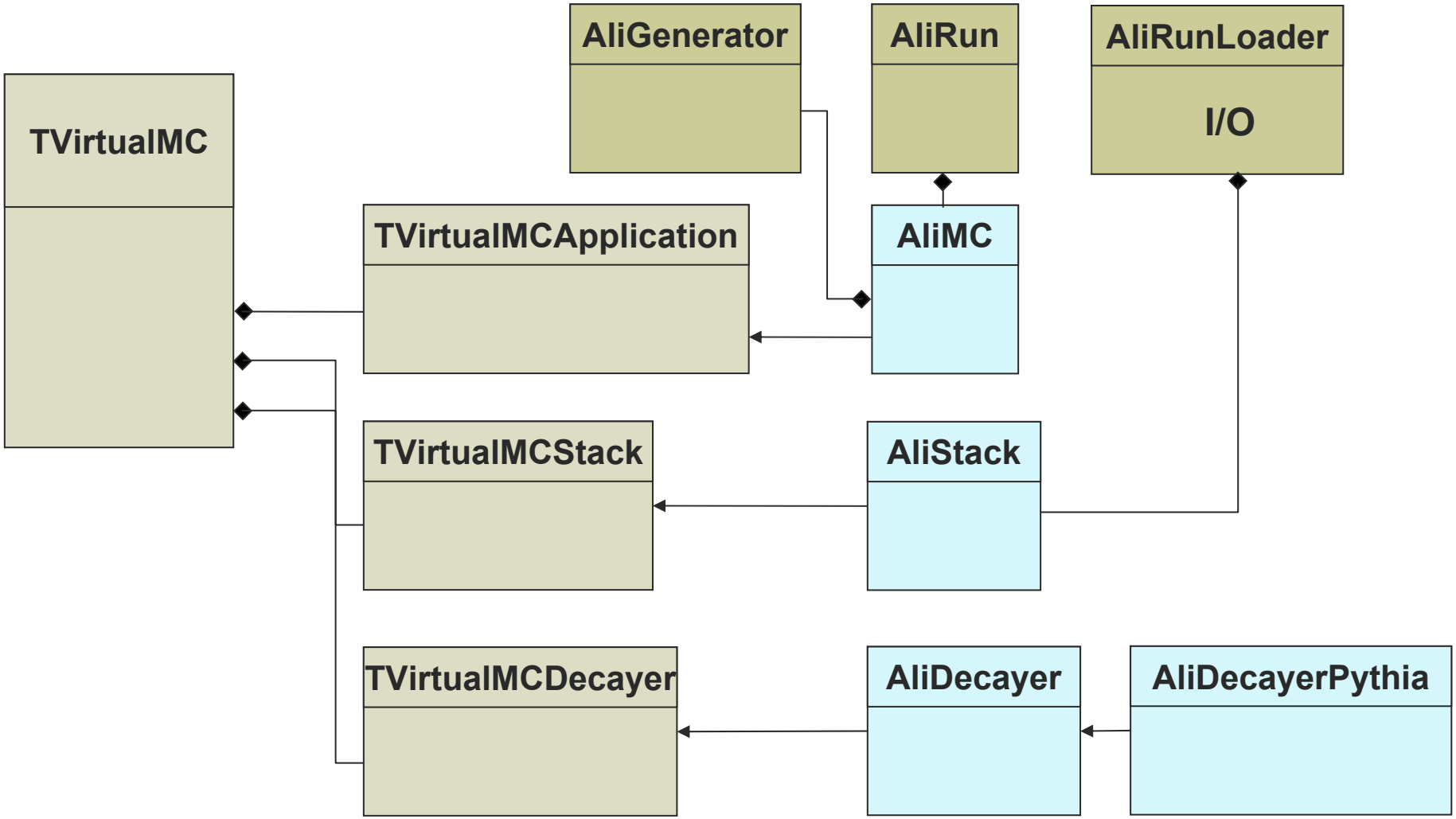


Parameterization Libraries

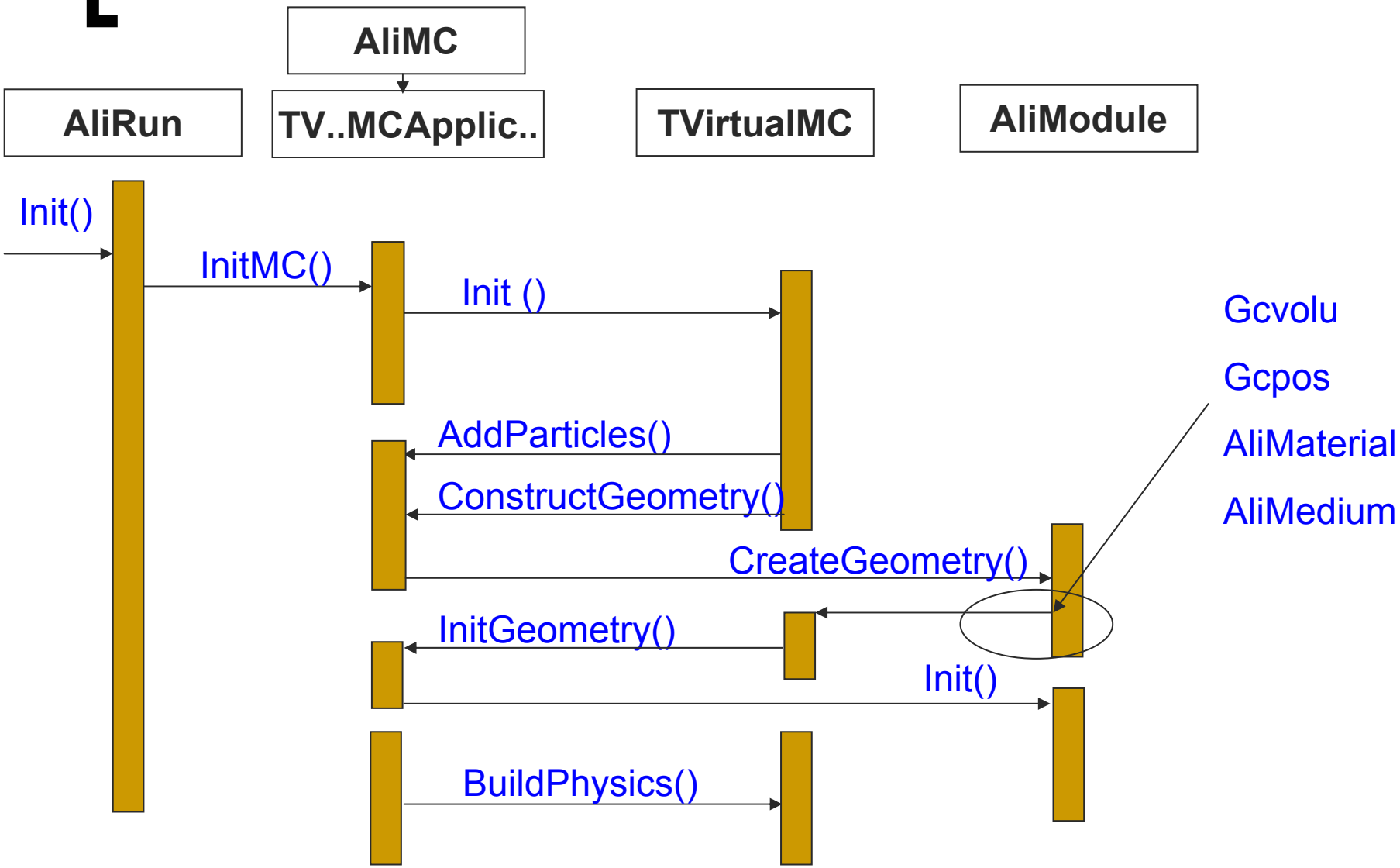
ALICE Specific Implementations of VMC Components



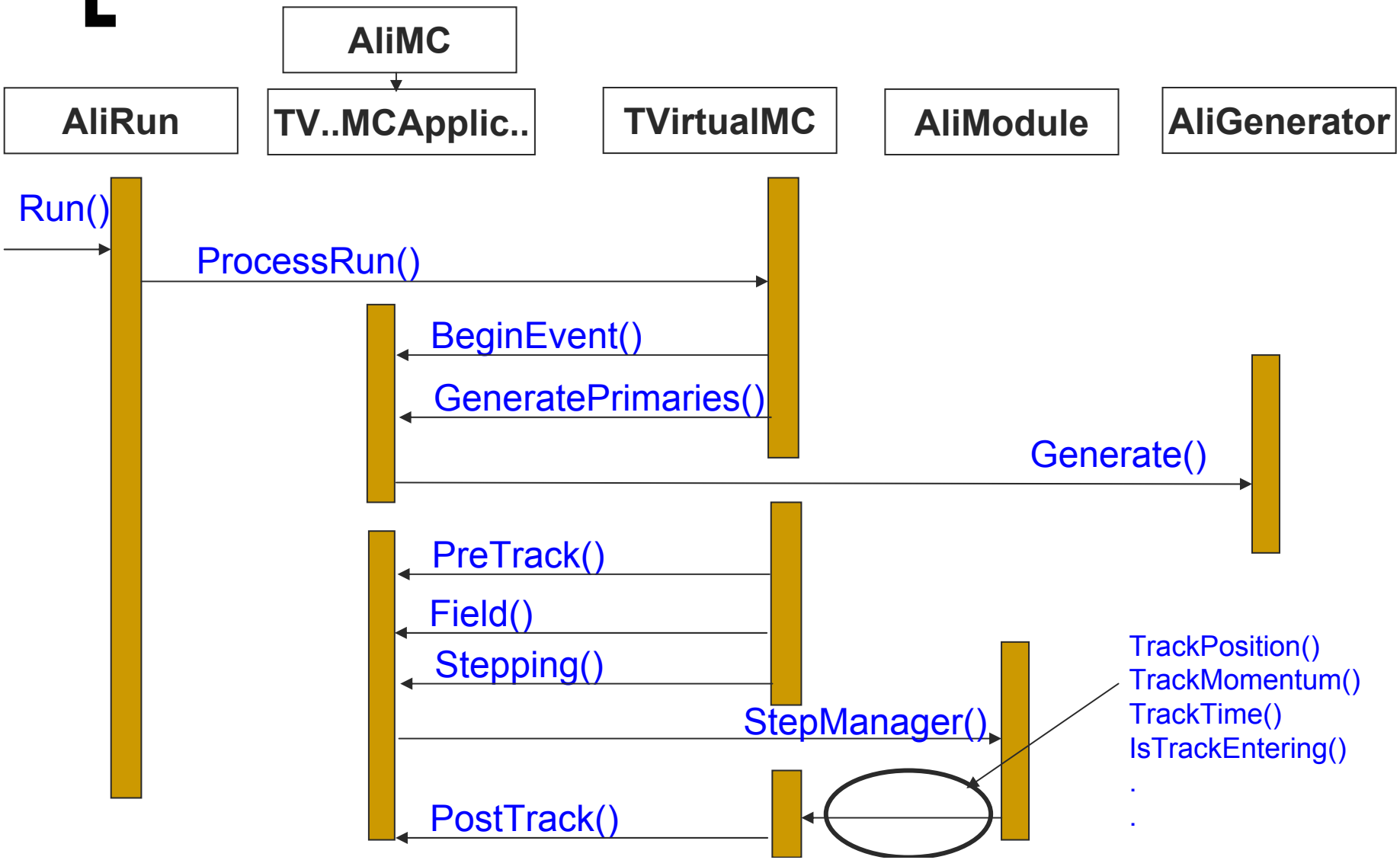
VMC in AliRoot



Initialisation



Simulation Run



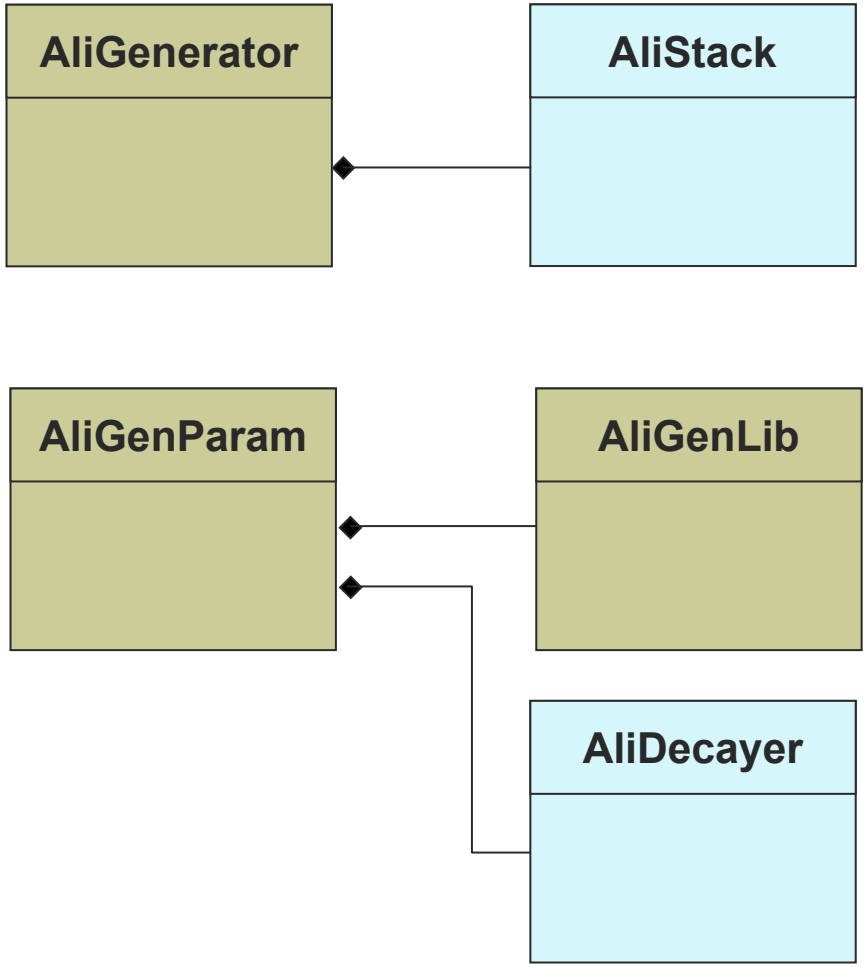
AliStack

- Implemented as *TClonesArray* of *TParticles*
- Mother-daughter relation via indices
- *PurifyKine()*
  - Keep only particles along direct genealogical line between flagged secondary (e.g. secondary producing hit) and primary particle.
- *ReorderKine()*
  - Reorder daughter particles of the same mother into continuous sequence. Needed for TFluka.

AliMCDecayerPythia

- Implemented using Pythia6 routine py1ent
  - Possibility to force certain predefined decay-modes
    - dimuon
    - smilileptonic
    - D->Kπ ....
- Used by
  - *TGeant*3 for decays of particles with finite lifetime but unknown decay modes
  - Generators like *AliGenParam* to decay short-lived resonances

VMC components and Generators



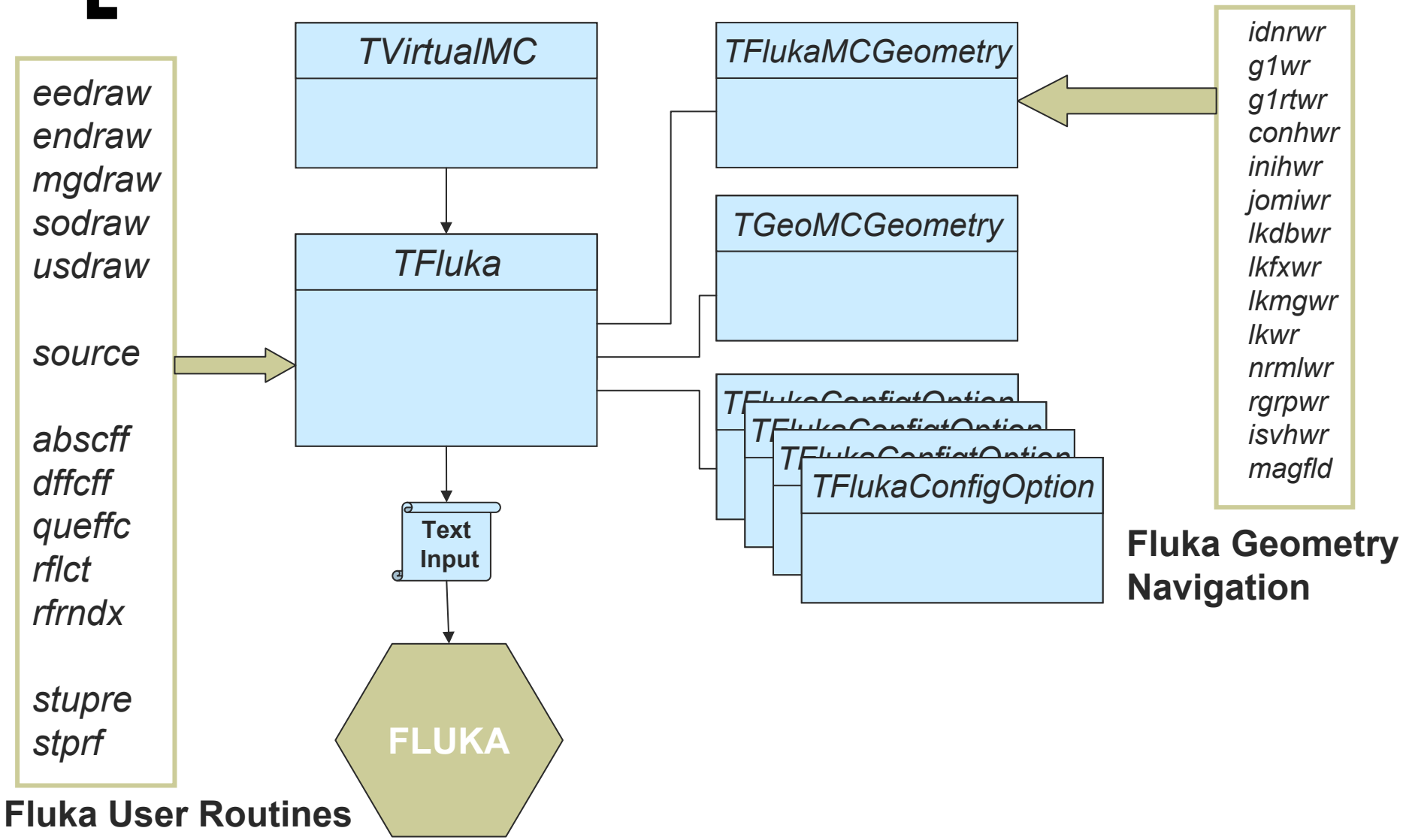
Run Time Configuration

```
{
new TGeant3("C++ Interface to Geant3");
TVirtualMCDecayer *decayer = new AliDecayerPythia();
decayer->SetForceDecay(kAll); decayer->Init();
gMC->SetExternalDecayer(decayer);
gMC->SetProcess("DCAY",1);
gMC->SetProcess("PAIR",1);
gMC->SetProcess("COMP",1);
gMC->SetProcess("PHOT",1);
.
.
.
```

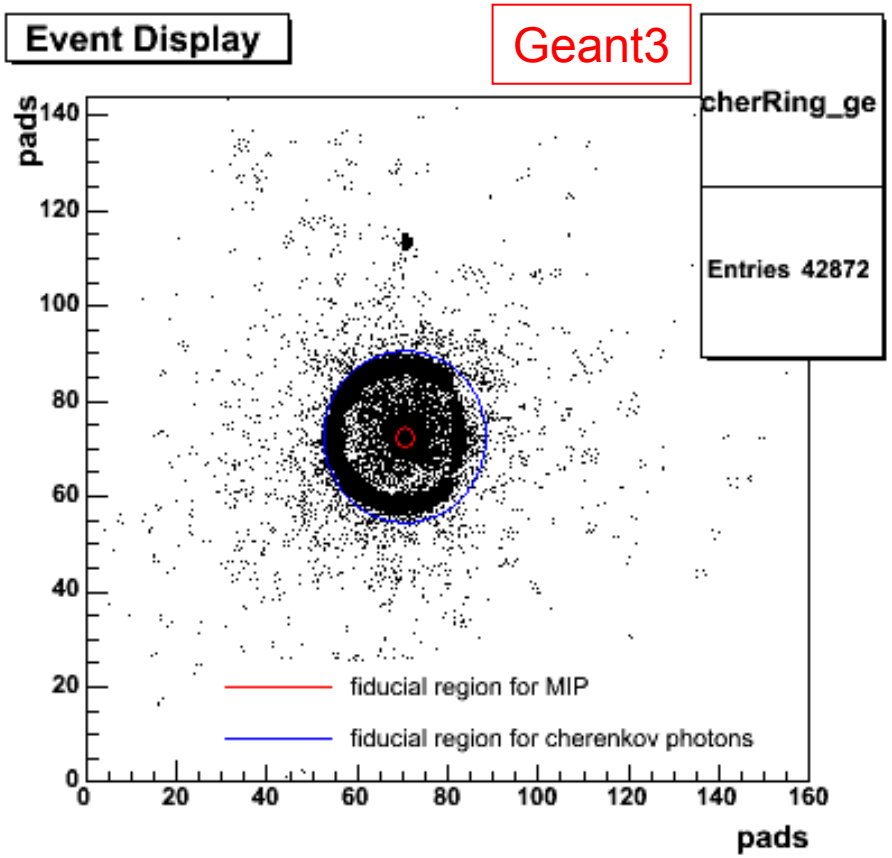
Virtual Monte Carlo: Status and Plans

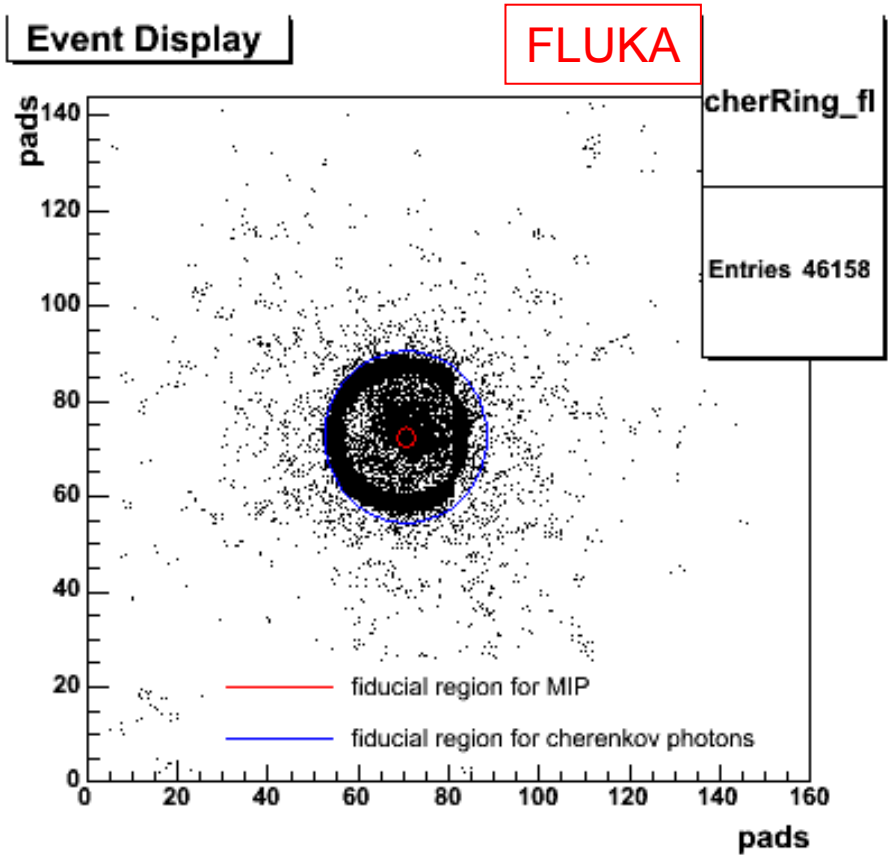
- TGeant3
  - Used in production
- TGeant4
  - Used for Geant4 physics validation
- TFluka
  - Implemented
  - Testing ongoing (in collaboration with Cuban institute (CEADEN))
  - Next PDC with TFluka

TFluka Implementation



HMPID: 5 GeV Pions





More ...

... in Andrei Gheata’s presentation