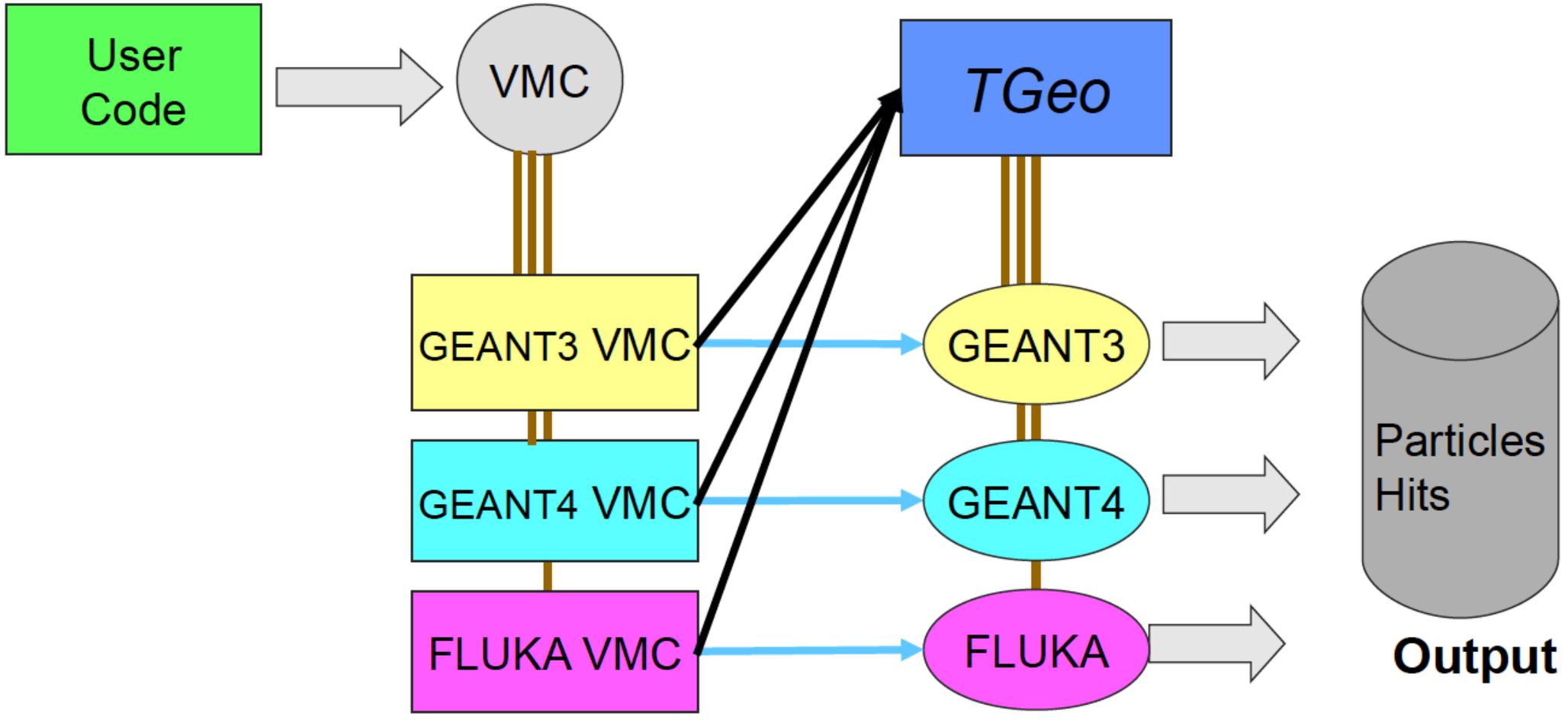
Ideas for Fast Simulation in the VMC Approach

Andreas Morsch
CERN

ALICE Offline Week, 15/3/2018

Virtual Monte Carlo VMC

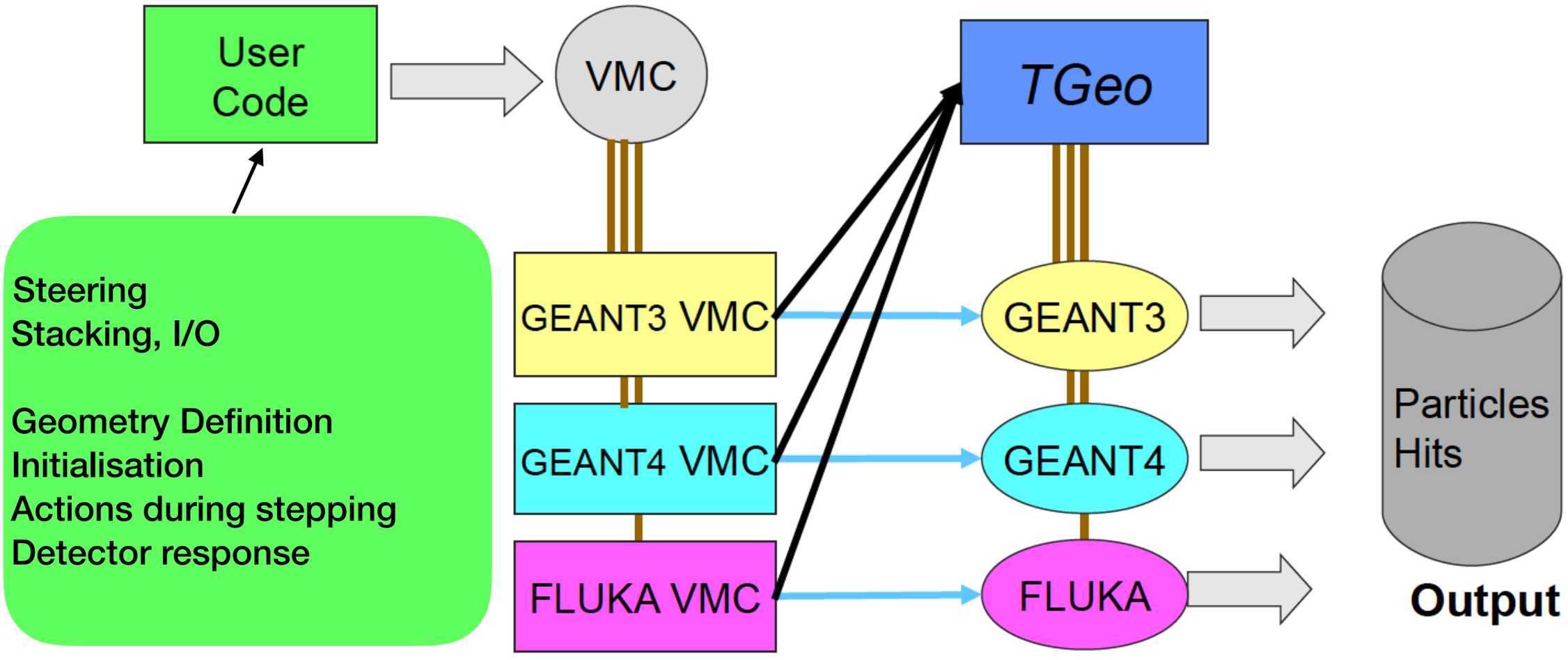
- **VMC allows user to run different transport Monte Carlos without changing**
  - **geometry definite**
  - **detector response definition**
  - **I/O format**
  - **…**
- **Base class for Transport MC *TVirtualMC***



Virtual Monte Carlo VMC

Transport MC transparent to the user application

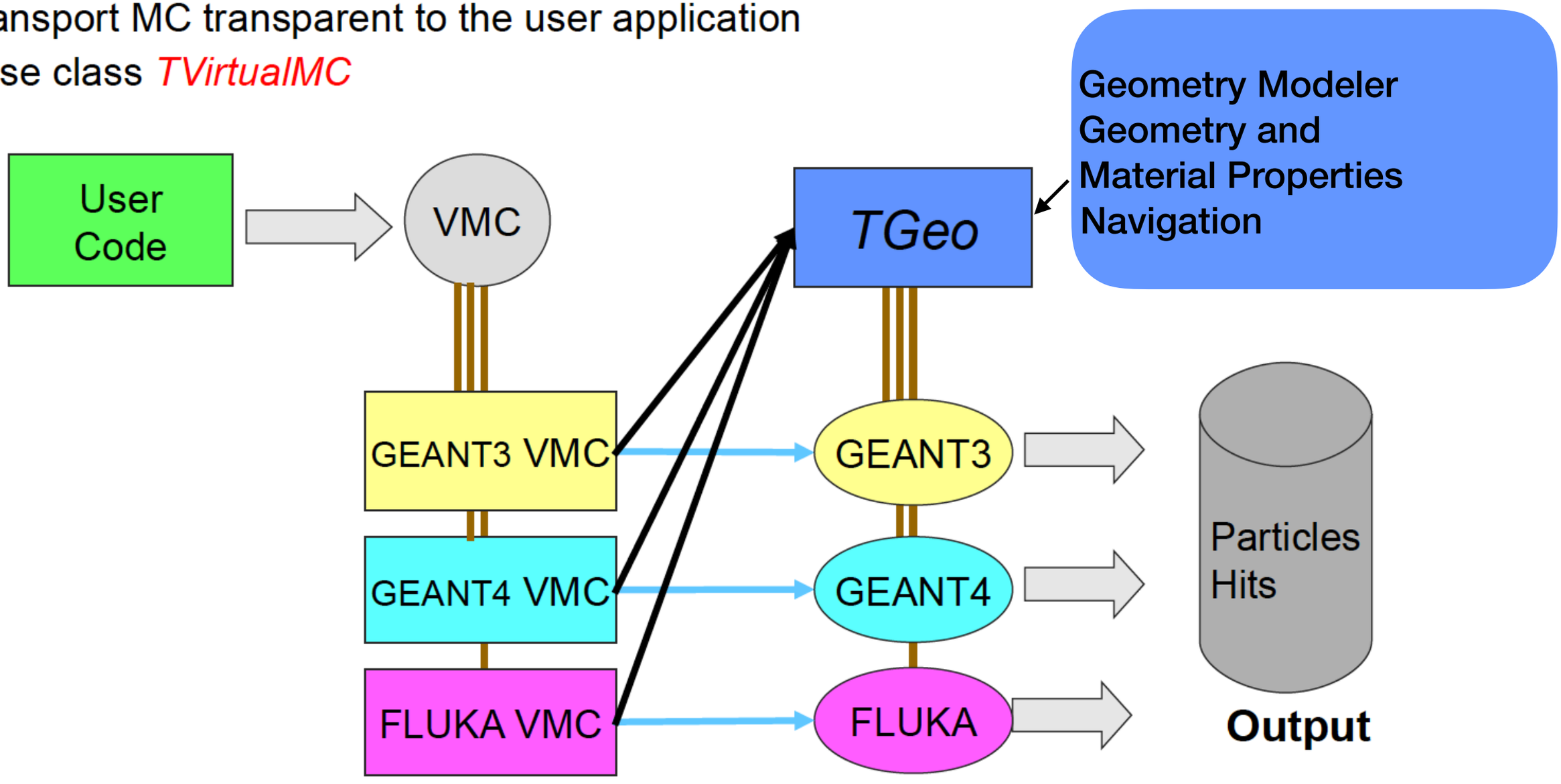
Base class *TVirtualMC*



Virtual Monte Carlo VMC

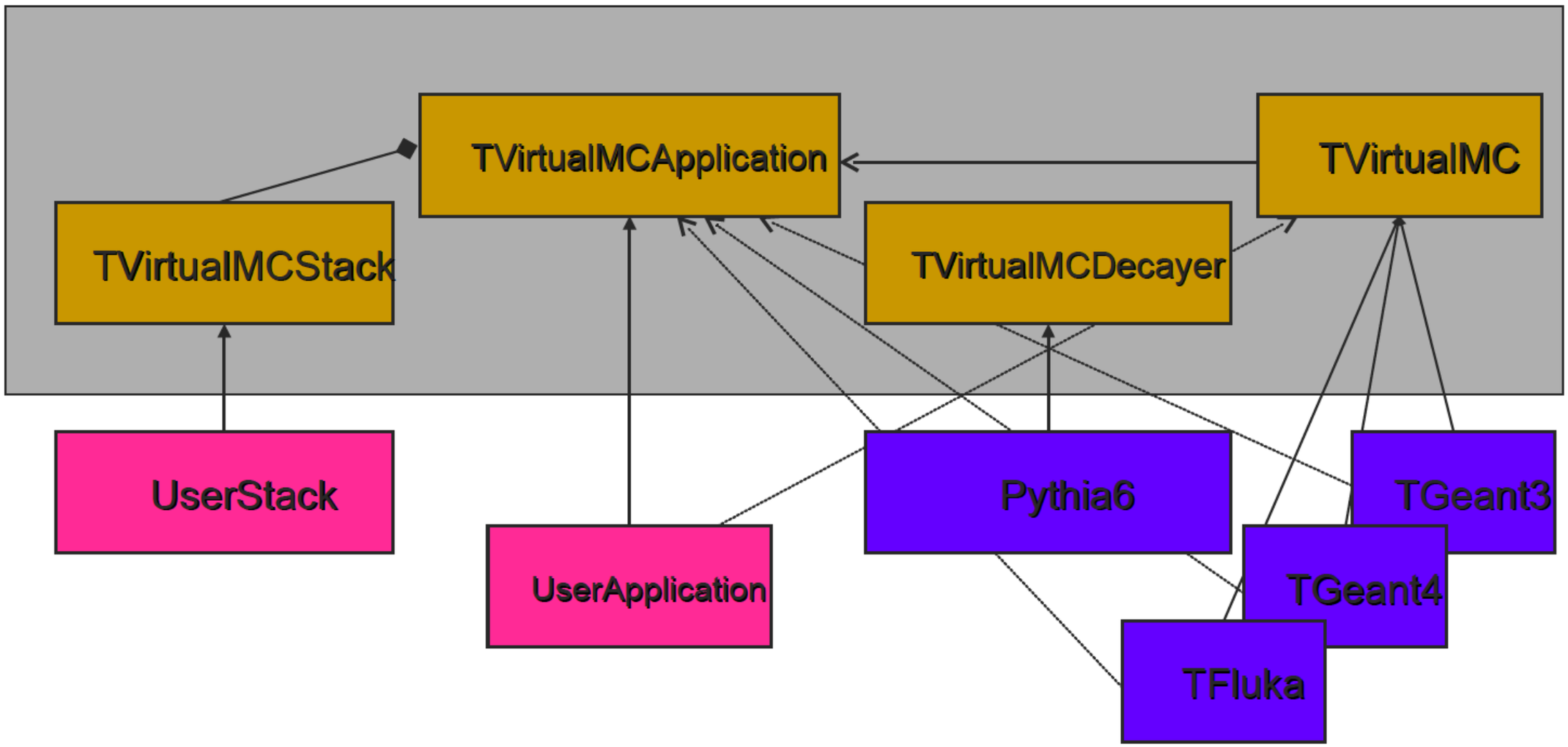
Transport MC transparent to the user application

Base class *TVirtualMC*



VMC Class Design

- User has to implement two mandatory classes
  - TVirtualMCApplication (AliMC)
  - TVirtualMCStack (AliStack)
- Optionally an external decayer
  - TVirtualMCDecayer (AliDecayerPythia6(8))



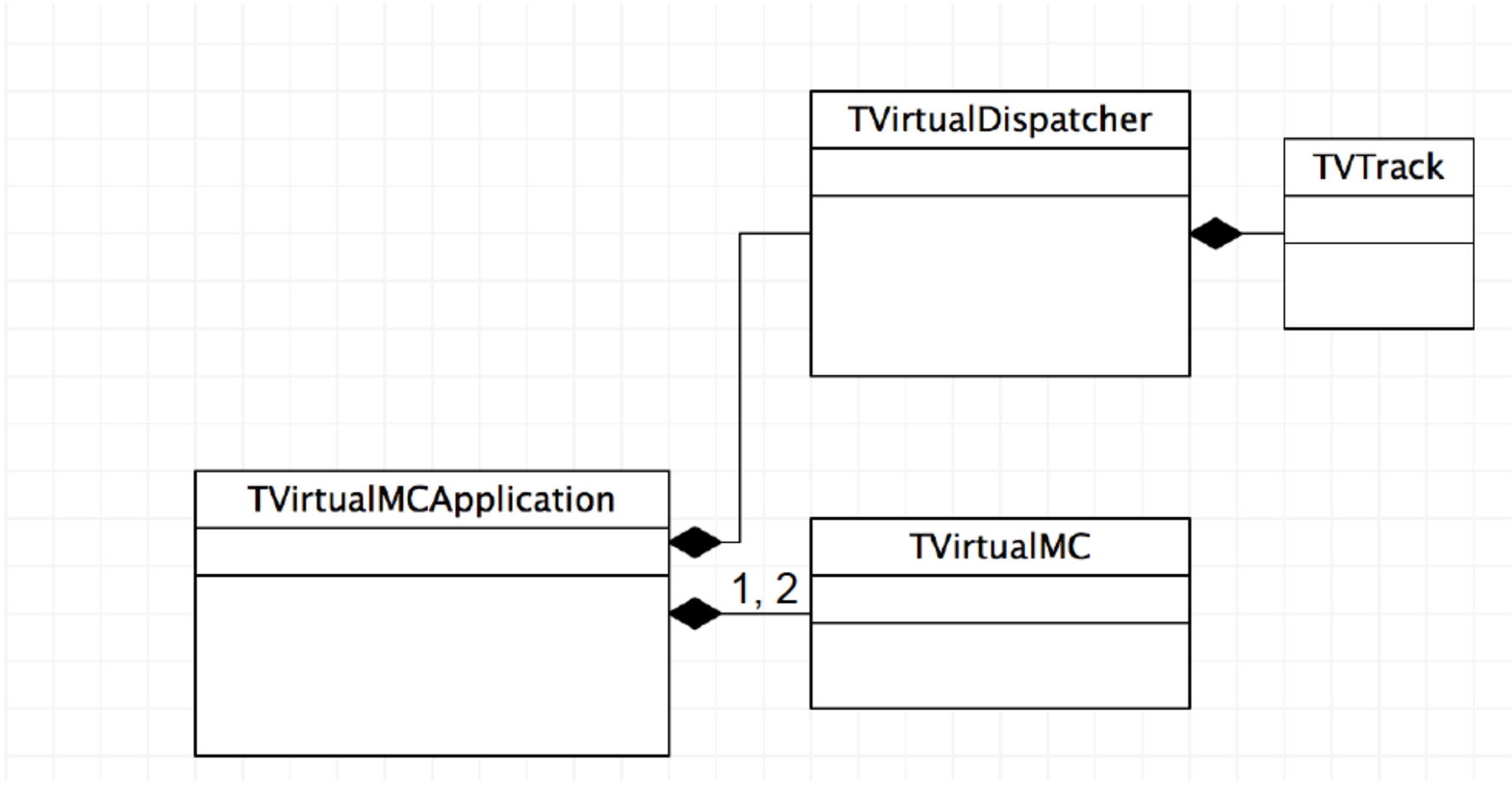
TVirtualMC Realisations

- TGeant3
  - still our main transport engine
- TGeant4
  - being validated / used for specific productions
- TFluka
  - radiation calculations
  - specific simulations
- so far no fast simulation

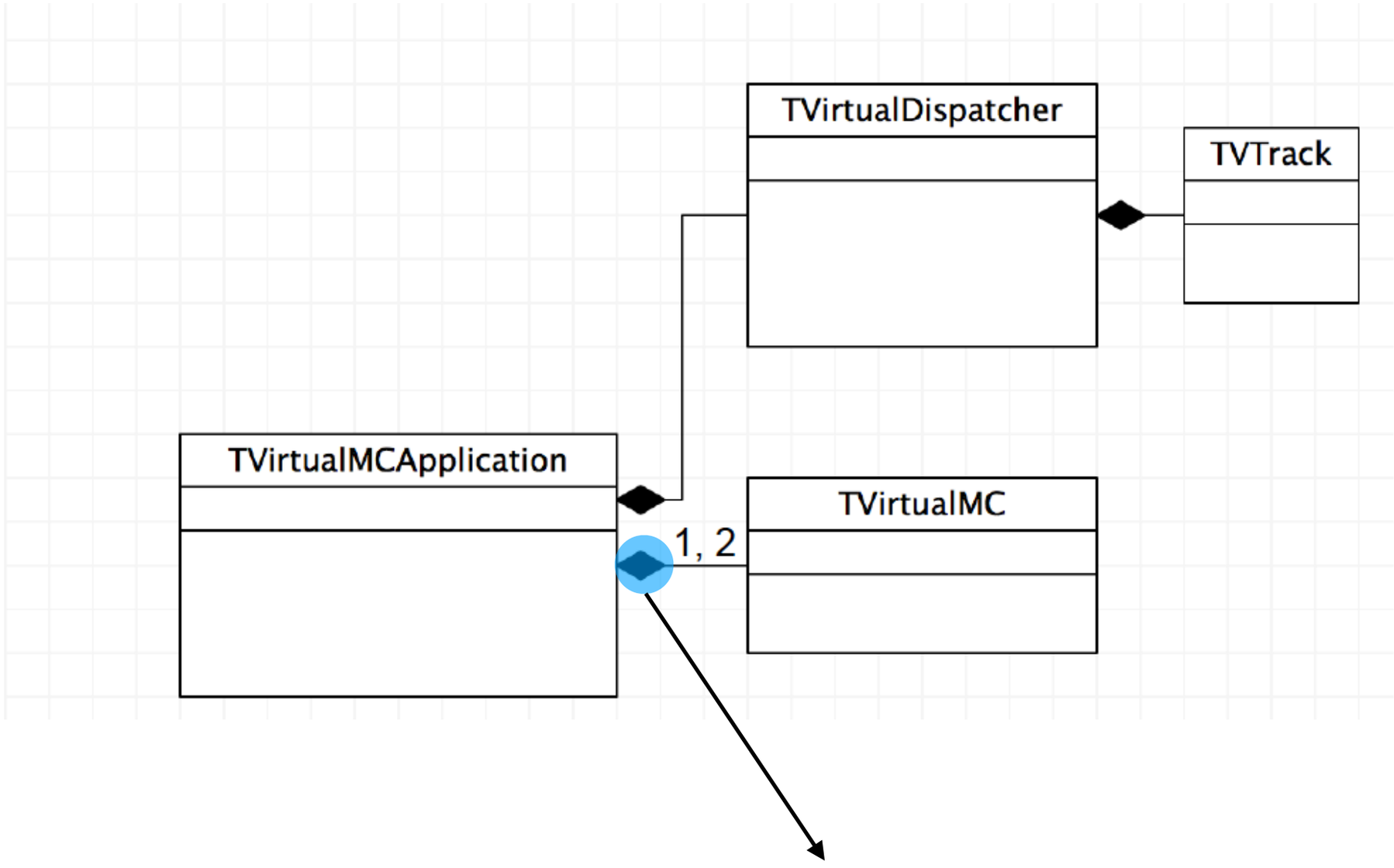
Possible VMC Extension for Fast Simulation

- Integrated Analysis Facility: Combine fast and slow simulation
- Ideally Fast/Slow simulation engine transparent to user code
  - same StepManager, Hit and MC truth management
  - same or simplified geometry
- Need more then one TVirtualMC instance
- Need dispatcher component to decide which instance is active depending on
  - particle type
  - region
  - volume
- Need global track class
  - to pass information between different transport engines

Possible Extension for Fast Simulation



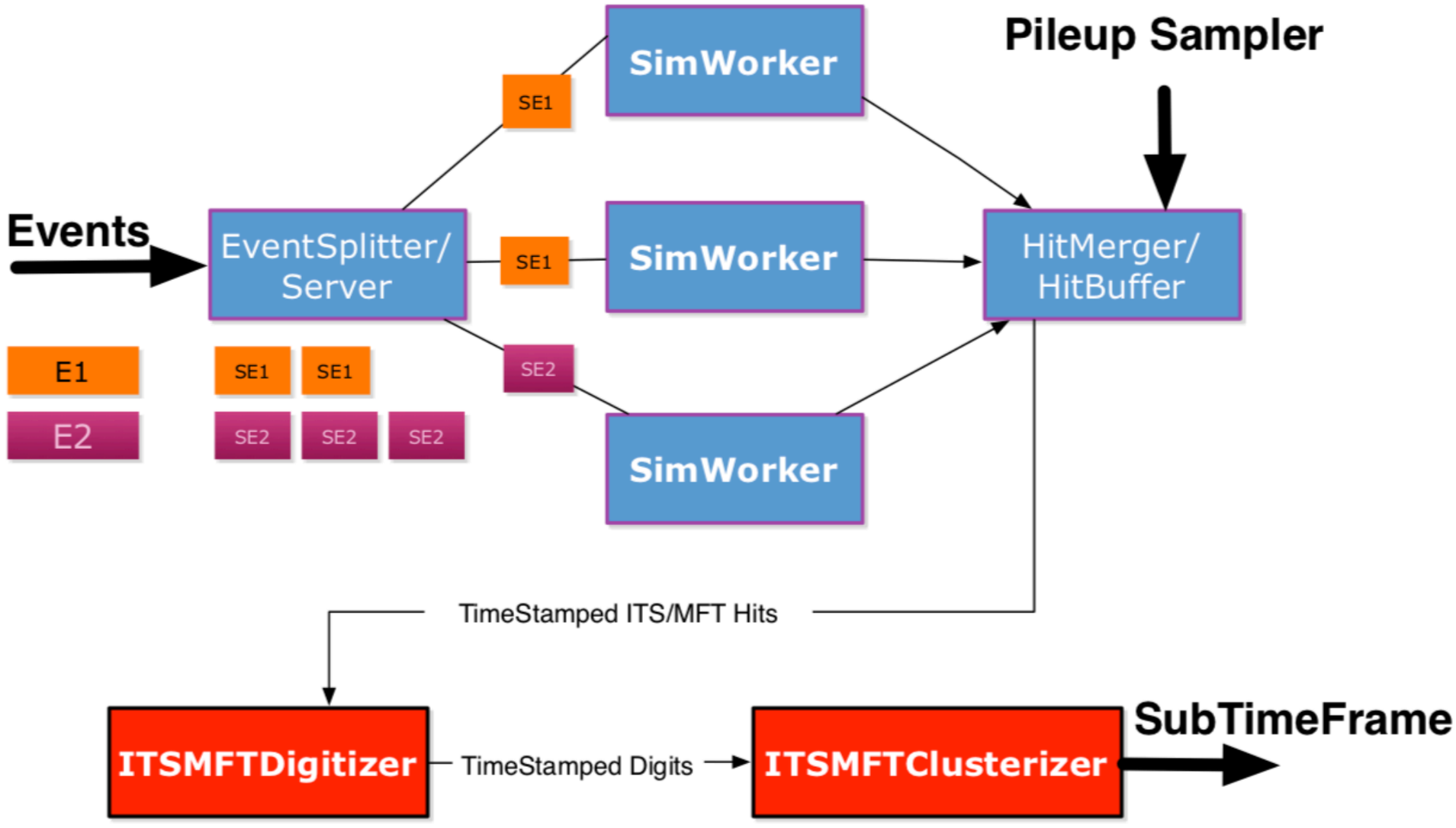
Possible Extension for Fast Simulation



to be defined: pointers, processes, nodes …

PROTOTYPE

Fully asynchronous simulation and digitization



- New simulation scheme might directly forward hits to digitization/clusterization to have fully asynchronous production of readout signal, yielding SubTimeFrames of clusters
- Can avoid writing hits / digits to disc
- **Prototype ready for ITSMFT!**

Next Steps

- VMC Extension
  - Implementation of new classes with minimal functionality
  - **Define Calling Sequences**
    - stop and restart transport with different transport codes
    - dispatcher step to signal transition to user code
      - track status: changing transport
    - preserve track identity
- Rethink double role of stack
    - particle history
    - task queue
- …

Next Steps

- Discuss and define ALICE use cases
  - can existing central barrel code be ported to VMC ?
  - other use cases?
    - Machine Learning & FastSimulation
- Full implementation