Common simulation framework

Andrea Dell'Acqua – CERN EP/SFT

+

Rene Brun - CERN EP/SFT

Witold Pokorski - CERN EP/LBC

Maya Stavrianakou - CERN EP/CMC

What is it?

- An attempt to answer a request by RTAG 10
  - Common project to address general simulation infrastructure and services
  - Minimize duplication of work, waste of effort and divergence
  - Provide a model for collaboration between experiments and simulation projects
  - Use different simulation engines transparently in the context of the experiment software base
- In fact a very efficient way of exchanging information and profiting of each other's results and experience

The simulation “engines”

- Geant3
  - For years the only package used for extensive and detailed detector simulation
  - Now used as “reference”, but it does not really fit in the plans of any LHC collaboration
- Geant4
  - Developed for detector simulation at the LHC
  - Gaining momentum as the user community grows in size
- Fluka
  - For years THE reference for background calculations and radiation studies
  - Growing interest to use it for overall detector simulation studies

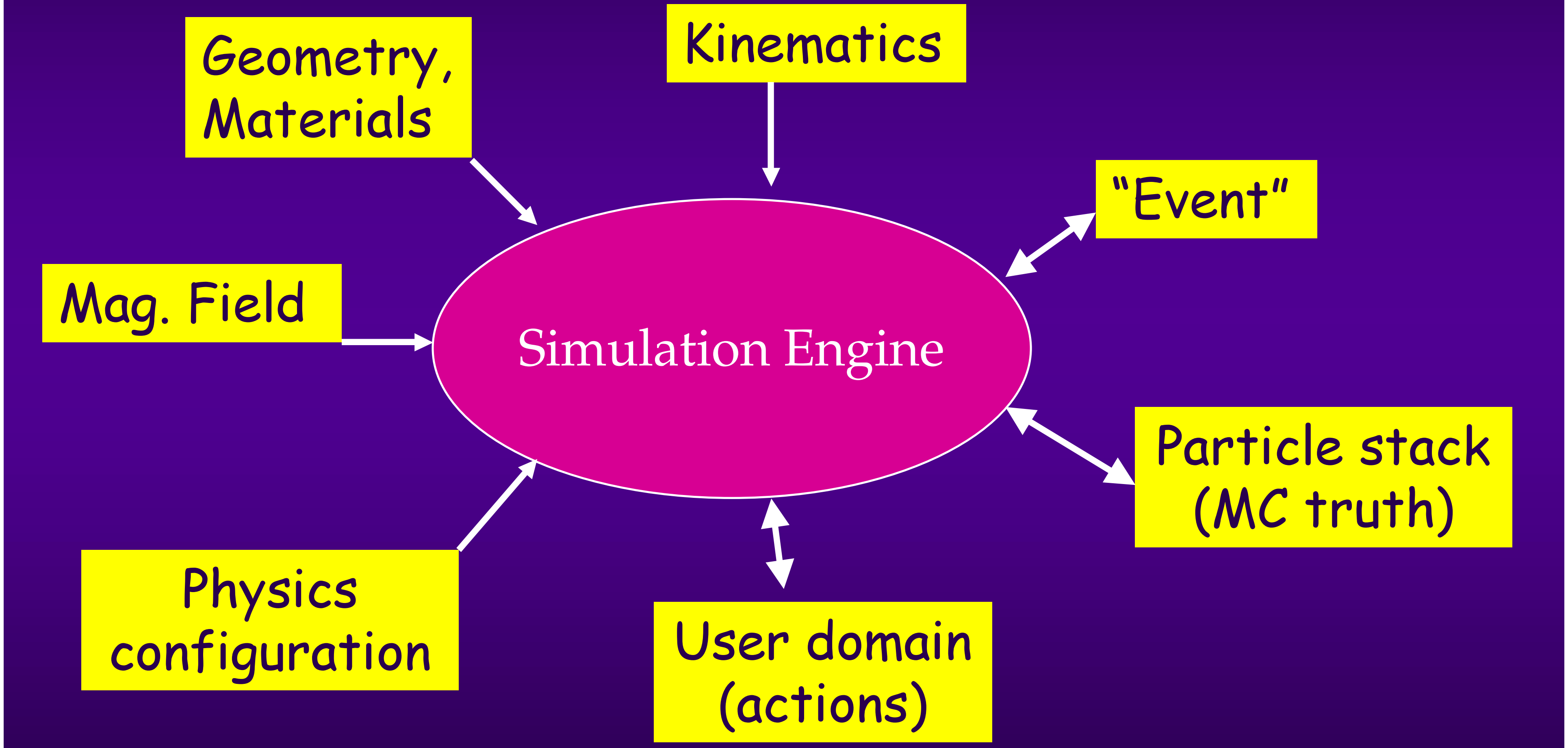
The experiments

- ATLAS, CMS and LHCb are completing their move to Geant4
  - Physics validation studies, optimization, performance
    - Feed back to Geant4 itself
  - Planning for big production exercises soon (CMS) or later this year
  - Interest in Fluka as alternative engine (don't want two different simulation streams)
- ALICE's target is Fluka
  - Not a big interest in G4
  - Develop a detector simulation framework where different simulation engines can used from the same application

The Virtual Monte Carlo

- Provides an abstract interface to detector simulation package
- Concrete implementations dealing with the existing packages (Geant3, Geant4, Fluka)
- Actual simulation engine selected at run time
- Abstract interfaces currently available in (and distributed with) ROOT
- Geant3 currently favored over Geant4 in the concrete implementation, interface to Fluka currently under development in ALICE

A detector simulation perspective



Geometry, Materials

Kinematics

"Event"

Mag. Field

Simulation Engine

Particle stack (MC truth)

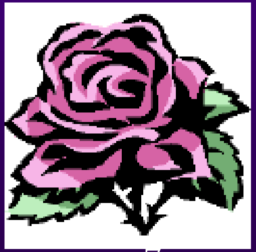
Physics configuration

User domain (actions)

The common simulation framework sub-project

- Aim: to provide a "service" for each yellow box (application domain)
- To be integrated within SEAL
- To be built on top of existing SW, as much as possible
- The glue for connecting the "services" is provided by the framework, as well as basic services (e.g. persistency)

Revised Overall Simulation Environment



What are we doing?

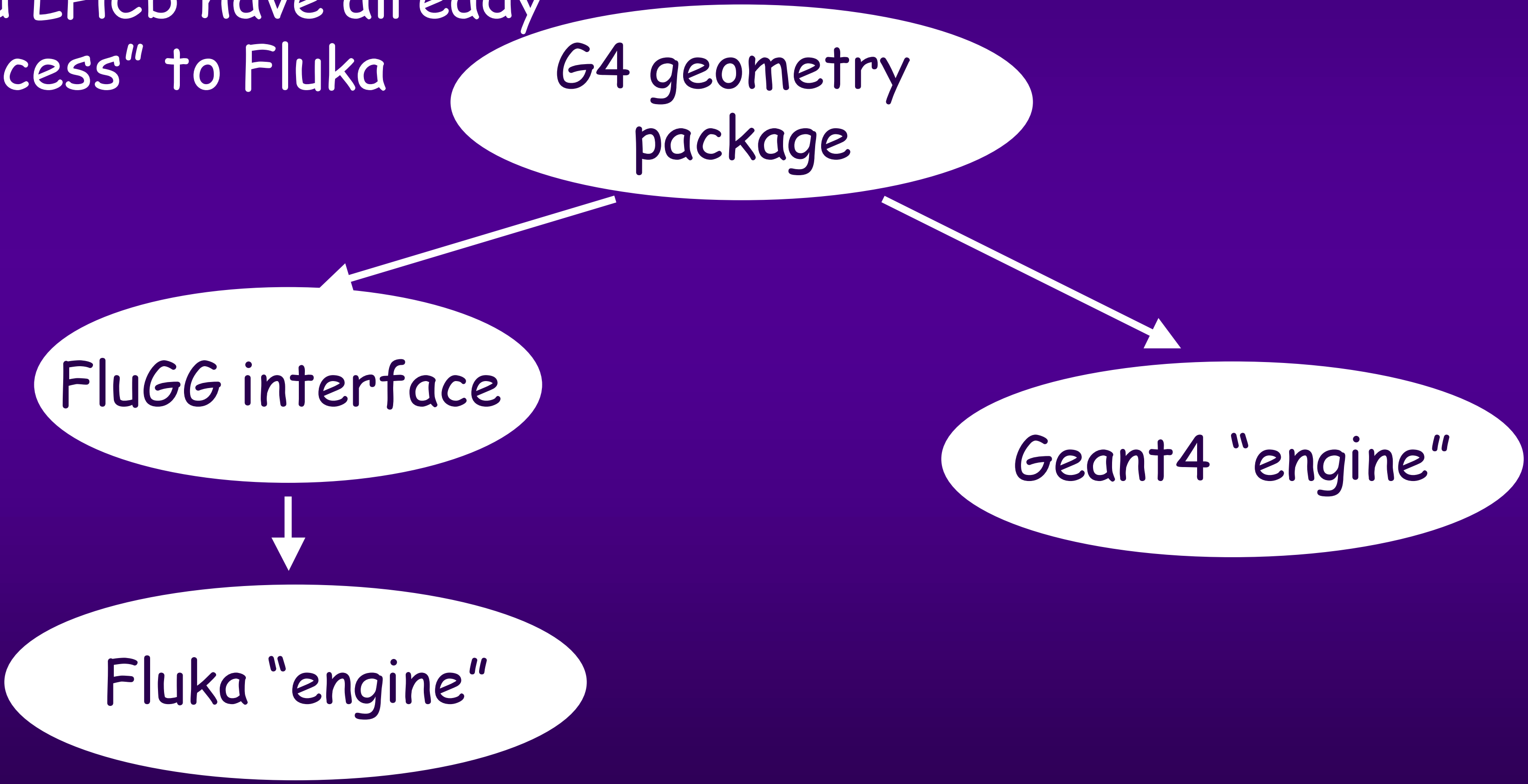
- Examining requirements, looking for common discussion ground
- "Designing" high level services, starting from existing applications
- Learning about SCRAM
- Inquiring about SEAL and its functionality
  - We are actually happy to finally have something to play with
- Toying with the VMC
- Setting up some prototype…

The hottest – the Geometry Box

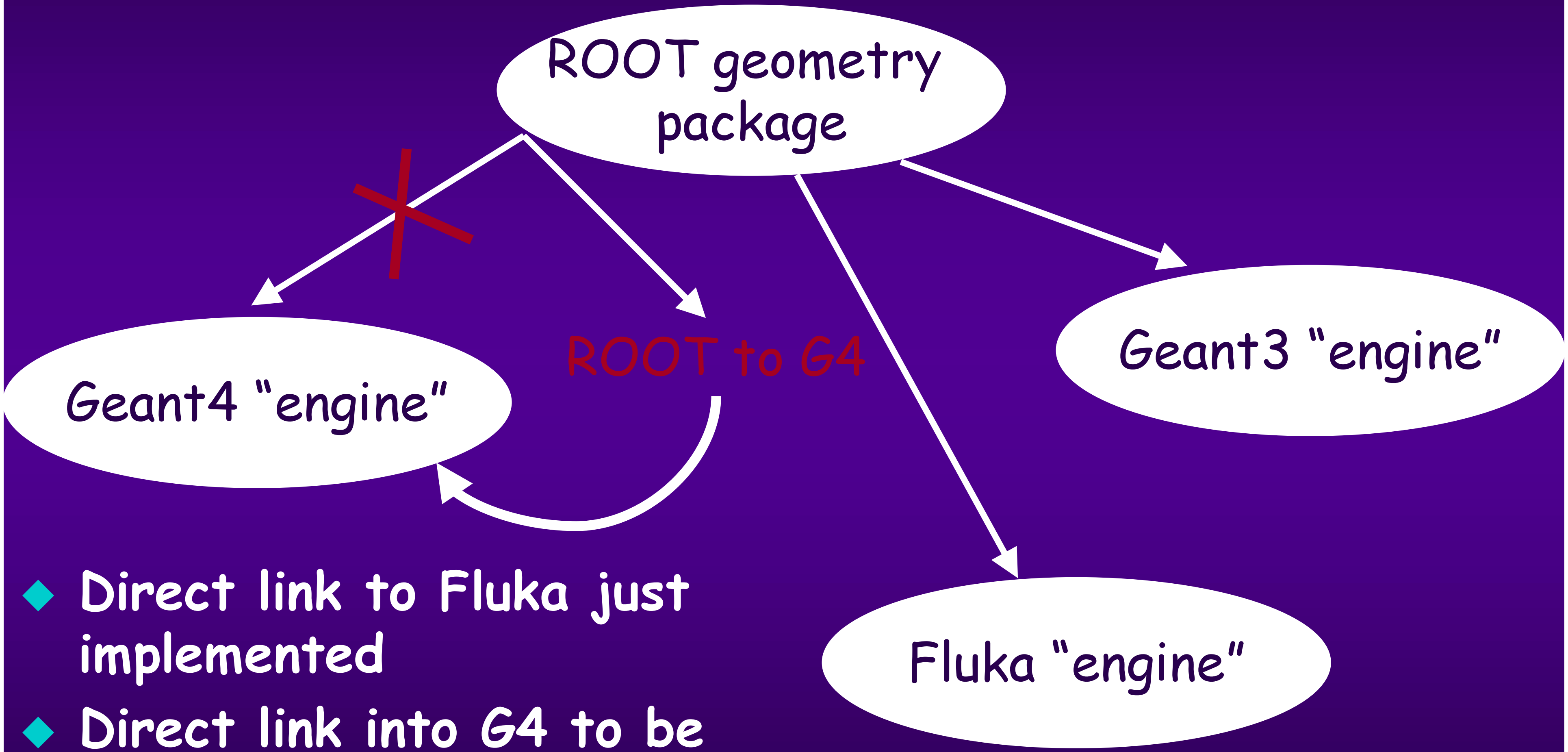
- Each experiment has developed/is developing its own solution
  - ATLAS, CMS and LHCb hide the Geant4 geometry construction behind a "Detector Description" layer+ automatic builders
  - ALICE use the ROOT geometry package
- There does not seem to be a different solution than supporting all of them
- Can we converge on 1 geometry package which satisfies all needs?
  - All "engines" must speak the same geometry language

Geometry from the G4 viepoint

In principle ATLAS, CMS and LHCb have already "access" to Fluka



Geometry from the VMC viewpoint



- Direct link to Fluka just implemented
- Direct link into G4 to be realized

The most urgent – MC truth

- Generic solution needed for
  - Storing the actual event being simulated
  - Storing secondaries of interest (decay products)
  - establishing association between hits and tracks
- Different experiments, different solutions
  - From simple “ad hoc” solutions to complete re-implementation of the particle stack
- Different engines require different strategies

Future steps

- Evaluation of the VMC
  - Does it satisfy all requirements?
  - Does it fit into the proposed SW infrastructure?
  - Help implementing the “missing” link?
- Prototype implementation
  - Close collaboration with the Physics Validation project to provide them with a working infrastructure for comparisons/tests with different engines