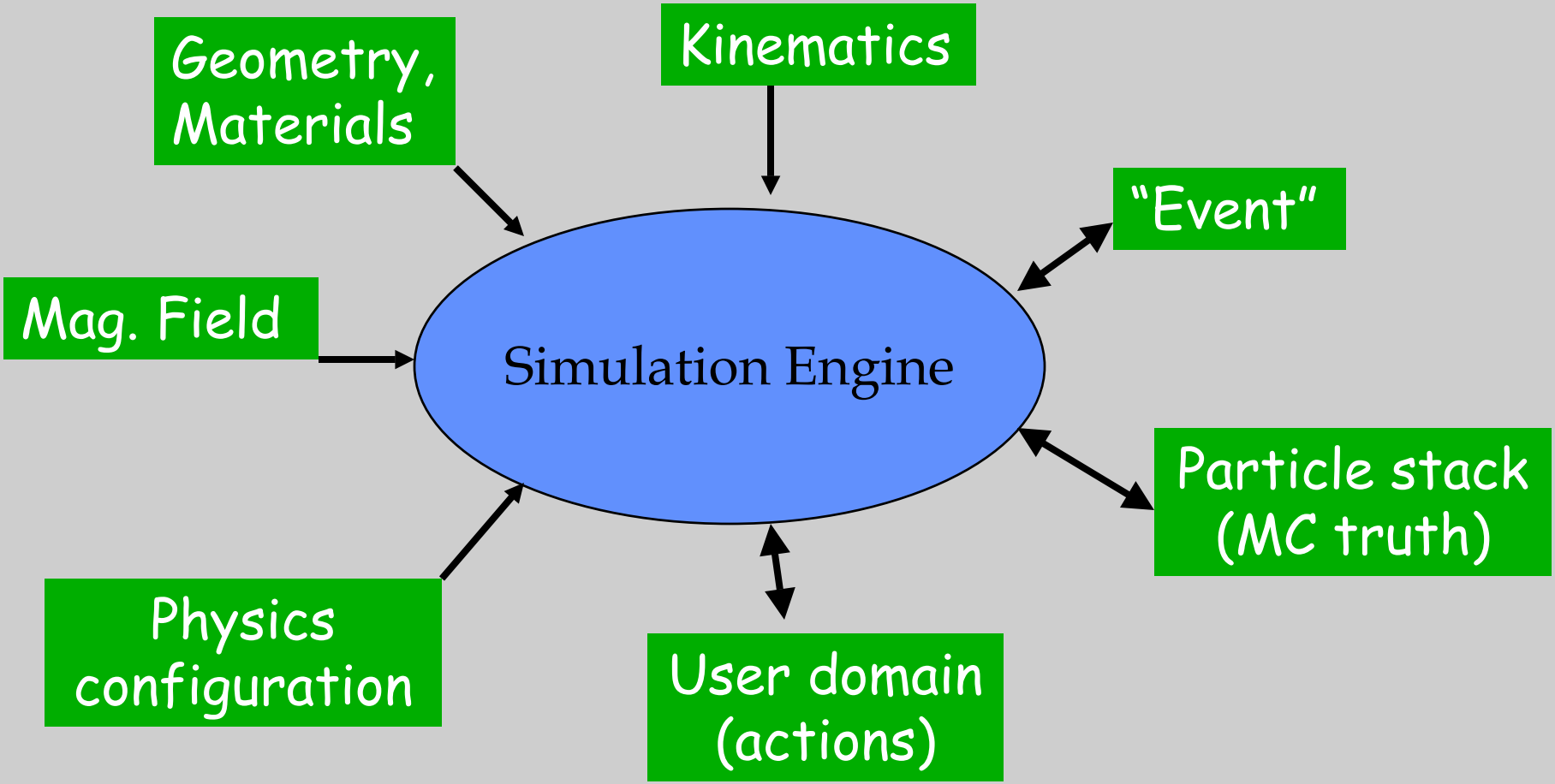
Common simulation framework

Andrea Dell'Acqua – CERN EP/SFT

What is it?

- An answer to 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
- Point of contact between the LHC collaborations and the simulation projects (Geant4, Fluka)
- In fact e very efficient way of exchanging information and profiting of each other's experience

An "ideal" detector simulation package



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

The simulation "engines"

- Geant3
  - 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
  - Adopted by ATLAS, LHCb and CMS as main simulation suite
- Fluka
  - For years THE reference for background calculations and radiation studies
  - Growing interest to use it for overall detector simulation studies, if a common simulation framework can be developed
  - ALICE's target

Who are the "clients"?

- The LHC collaborations
  - All experiments are very much involved in the commissioning/fine tuning of their simulation programs
  - Big productions are starting right now (CMS) or will start quite soon (ATLAS)
  - ALICE are deploying their own strategy (VMC)
  - No big interest in a common framework if that has deep repercussions on ongoing activities
    - Longer term programme
- The physics validation project (+beam test simulations)
  - Expressed interest in running all simulation engines for comparisons/cross checks
  - Common description of e.g. geometry is essential
  - They want it now!
    - Need for short term strategy

The Geometry conundrum

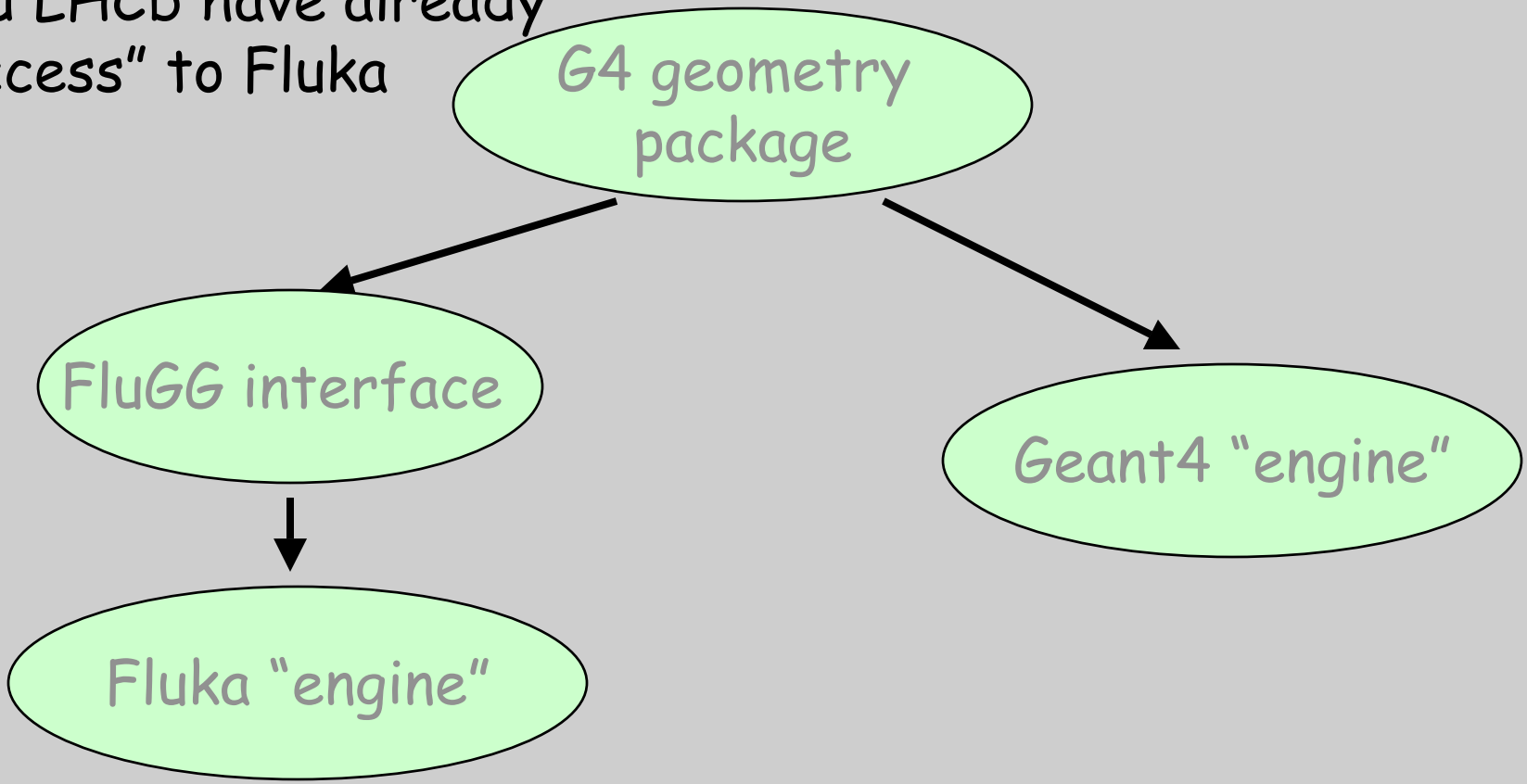
- Geant and Fluka come with their own (incompatible) geometry packages
  - The coupling between geometry and tracking is normally quite strong: geometry packages are easily interchangeable
- All "Engines" must speak the same geometry language
- ATLAS, CMS and LHCb committed to G4 and its geometry
  - But all have developed/are developing an independent geometry model which hides the geometry implementation
- ALICE moving to the ROOT geometry which, used in the context of the Virtual MC, can provide an unifying factor
- FluGG interface allows using the G4 geometry package from within a Fluka application

The Virtual Monte Carlo

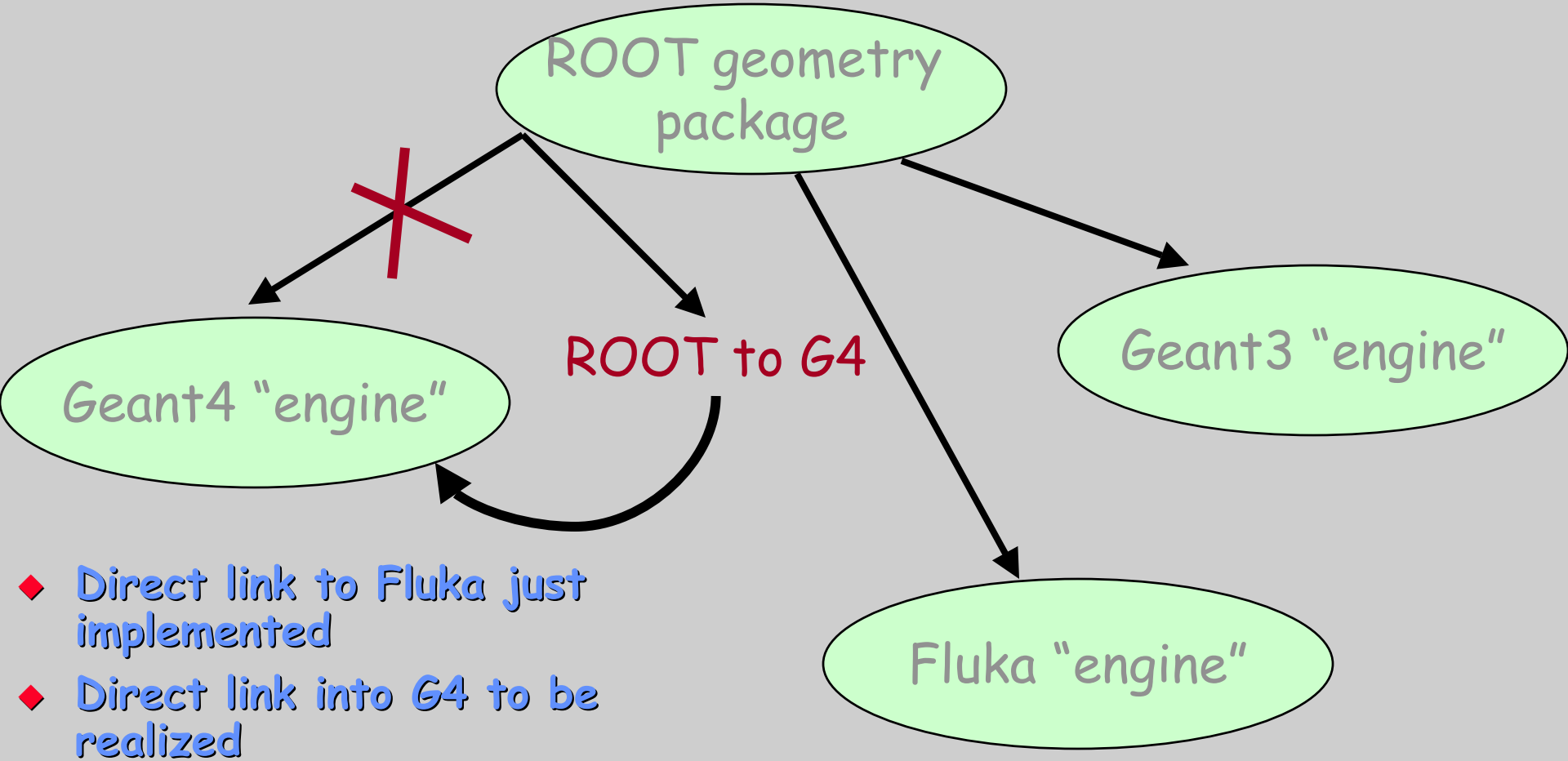
- Provides an abstract interface to detector simulation package
- Concrete implementations dealing with the existing packages (Geant3, Geant4, Fluka)
- Actual simulation engine selected at run time
- Abstract interfaces currently available in (and distributed with) ROOT
- Geant3 currently favored over Geant4 in the concrete implementation, interface to Fluka currently under development in ALICE
- Use the ROOT geometry for detector description. Interface to G3 and Fluka ready, interface to G4 requires in-memory translation

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

In the short term…

- Support the physics validation activities and test beam simulations
- Set up a simulation infrastructure based on Geant4 and Fluka/Flugg
  - Existing G4 benchmarks/simulation packages can be run with Fluka with minimum fuss
  - Try and standardize the simulation "output" by mimicking what the VMC does
- Evaluate the VMC (wrt the functionality offered by FluGG)

In the longer term…

- Evaluate the VMC as soon as it provides fully operational interface to Fluka, in order to understand if it fulfils other collaborations requirements
- Develop/Help developing the "missing link" between the ROOT geometry and G4
- Find a way to converge from distinct geometry description mechanisms in the experiments into common geometry models useable by the simulation engines
- Study the possibility of translating G4 geometries into ROOT geometries (ROOT to G4 already provided by the VMC)

In the meantime…

- A persistent exchange format for geometry is needed
  - For debugging purposes
  - For visualization
  - Also requested by the (non-LHC) G4 community
- An exchange format based on XML (GDML) has been developed primarily for G4 but could be generalized to cover other (e.g. ROOT) geometry systems
- It was felt that GDML could provide the bases for a persistency exchange format and should be supported
- Proposal on detector description work to be presented to the SC2 (TW+AD)