Next GEANT steps in Detector Simulation

René Brun

CERN/PH/SFT

Motivation

- Following the PH/SFT review in 2009, it was suggested to improve the synergy between the 2 main projects in the group: **ROOT** and **GEANT4**.
- Both projects are used very successfully by the LHC collaborations. Experiments are concerned by their evolution. Priority must be given now to the LHC.
- Many discussions in the past few months on the best way to improve this synergy.

Possible scenarii

- 1: set up a forum with participants from the LHC experiments to discuss point by point different suggestions.
- 2: set up a Task Force within SFT to gradually implement solutions enabling the synergy.
- 3: Start from existing GEANT4 and use features of ROOT like interpreter, I/O, graphics, GUI, using a common build system.
- 4:New project consistent with the considerable investment in G4 physics in particular, but also including several innovative techniques.

Starting Assumptions (I)

- The LHC experiments use extensively G4 as main simulation engine. They have invested in validation procedures. Any new project must be coherent with their framework.
- One of the reasons why the experiments develop their own fast MC solution is the fact that a full simulation is too slow for several physics analysis. These fast MCs are not in the G4 framework (different control, different geometries, etc), but becoming coherent with the experiments frameworks.
- Giving the amount of good work with the G4 physics, it is unthinkable to not capitalize on this work.

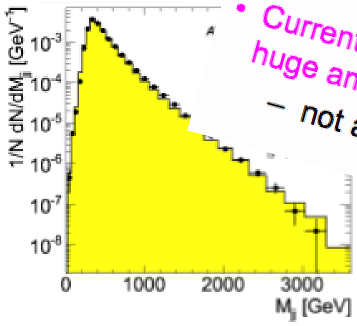
G4 and LHC data

Comparison Dat-MC at 7 Tev through
2 examples

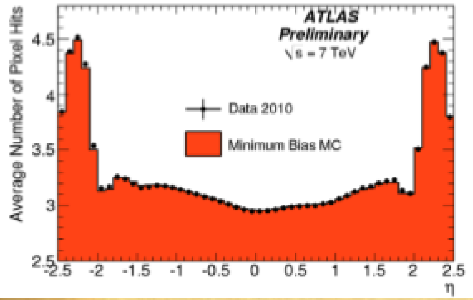
Jets reconstruction

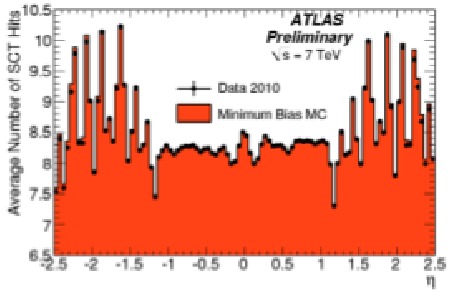
Dijet invariant mass distribution for jets with leading jet p_T(jet)>160 GeV and second jet >30 GeV, lumi~1pb^{-1}, |y|<2.8

Track variables
all detectors are included as well as the beam spot
Excellent agreement



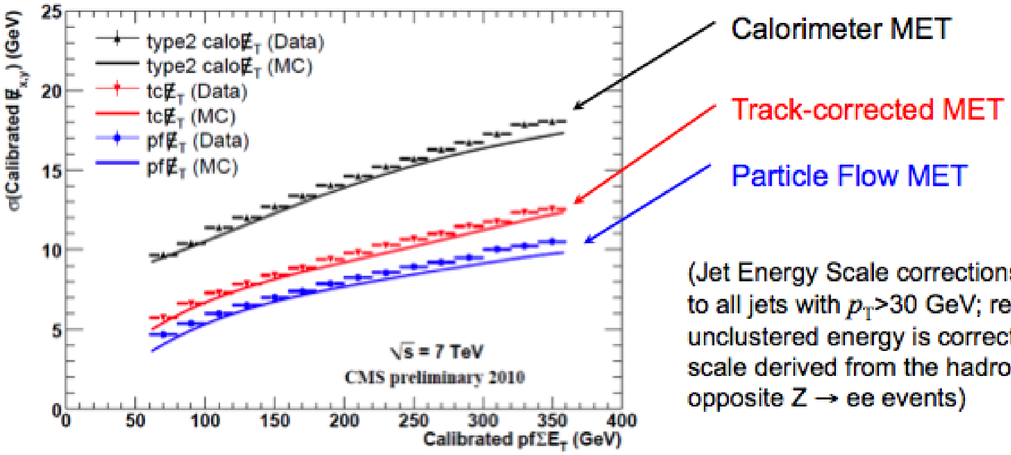
- Current (excellent) level of Data/MC agreement is a product of a huge amount of work over many years by many people
  - not an accident!





ution Studies

test of noise simulation, showering, and
elements have to be correct



(Jet Energy Scale corrections are applied to all jets with p_T>30 GeV; remaining unclustered energy is corrected using a scale derived from the hadronic recoil opposite Z → ee events)

Starting Assumptions (II)

- Considering the latest plans for the LHC schedule, it is unlikely that experiments are going to move to a new solution in the short/medium term.
- Also the experiments want to see a clear maintenance plan and usual developments at this stage for G4.
- Before moving to anything new, it must be demonstrated soon enough that this « anything new » is worth investigating. This implies a solid participation of the experiments right from the start to this « anything new »

Starting Assumptions (III)

- Experiments continue to validate their real data versus the full G4 simulation.
- Atlas and CMS are investing in FastMC in addition to G4.
- GeantE is used in reconstruction.
- The VirtualMC is successful as a way to compare simulation tools or to move from old to new ones.

Starting Assumptions (IV)

- From ROOT we want to use
  - I/O
  - Interpreters
  - Graphics
  - EVE
  - Math
  - infrastructure

FastMC

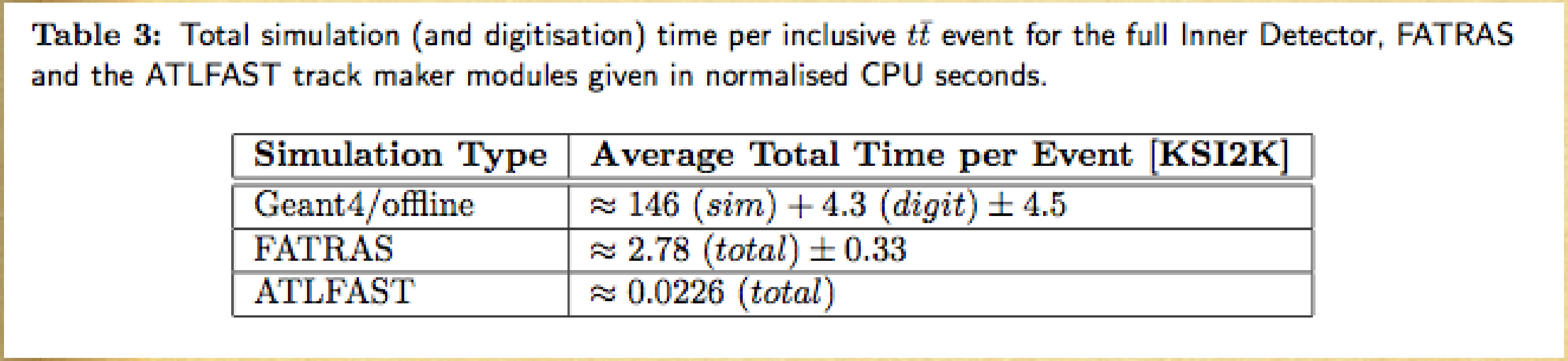


Table 3: Total simulation (and digitisation) time per inclusive $t\bar{t}$ event for the full Inner Detector, FATRAS and the ATLFAST track maker modules given in normalised CPU seconds.

Simulation Type	Average Total Time per Event [KSI2K]
Geant4/offline	$\approx 146\ (sim) + 4.3\ (digit) \pm 4.5$
FATRAS	$\approx 2.78\ (total) \pm 0.33$
ATLFAST	$\approx 0.0226\ (total)$

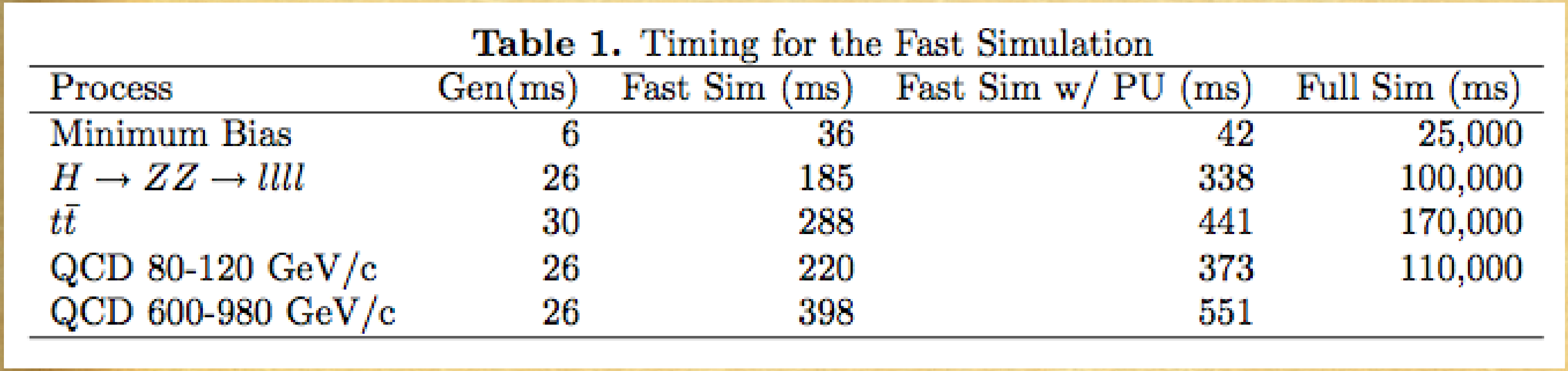
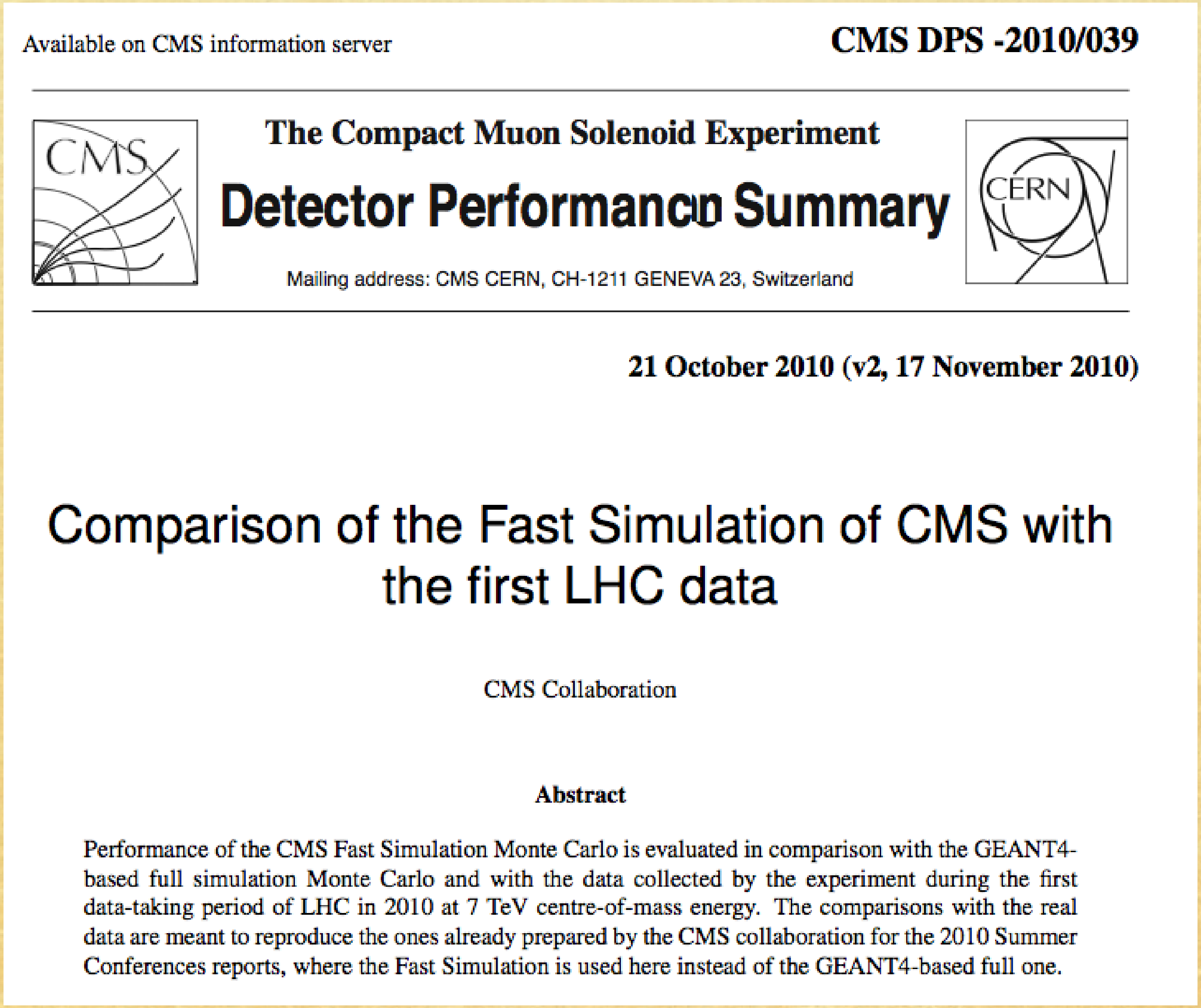


Table 1. Timing for the Fast Simulation

Process	Gen(ms)	Fast Sim (ms)	Fast Sim w/ PU (ms)	Full Sim (ms)
Minimum Bias	6	36	42	25,000
$H \to ZZ \to llll$	26	185	338	100,000
$t\bar{t}$	30	288	441	170,000
QCD 80-120 GeV/c	26	220	373	110,000
QCD 600-980 GeV/c	26	398	551	

FastMC in CMS



Available on CMS information server

CMS DPS -2010/039

The Compact Muon Solenoid Experiment

Detector Performance Summary

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

21 October 2010 (v2, 17 November 2010)

Comparison of the Fast Simulation of CMS with the first LHC data

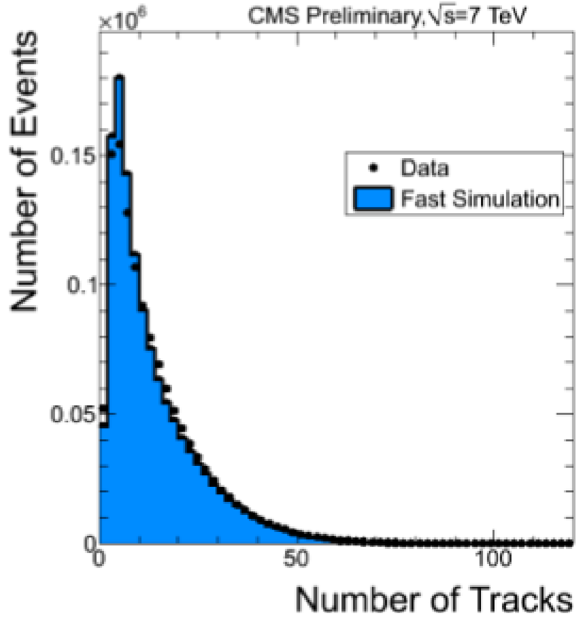
CMS Collaboration

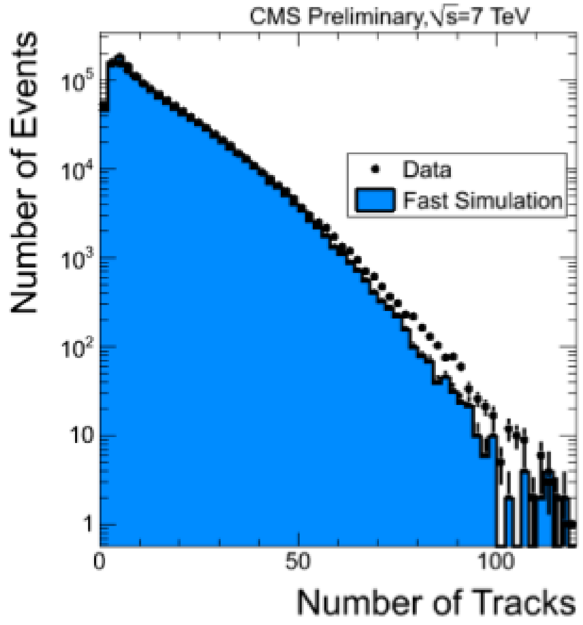
Abstract

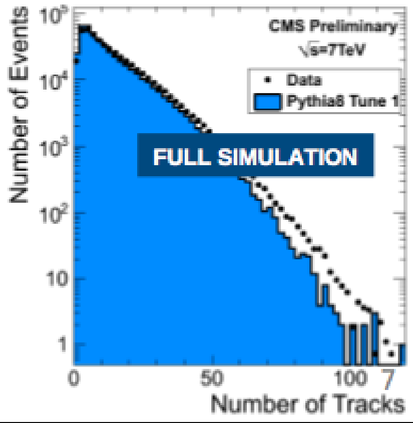
Performance of the CMS Fast Simulation Monte Carlo is evaluated in comparison with the GEANT4-based full simulation Monte Carlo and with the data collected by the experiment during the first data-taking period of LHC in 2010 at 7 TeV centre-of-mass energy. The comparisons with the real data are meant to reproduce the ones already prepared by the CMS collaboration for the 2010 Summer Conferences reports, where the Fast Simulation is used here instead of the GEANT4-based full one.

FastMC in CMS

Number of reconstructed tracks







Lower number of events with high track multiplicity in the Fast Simulation with respect to the data. The same also happens in Full Simulation: Pythia tuning.

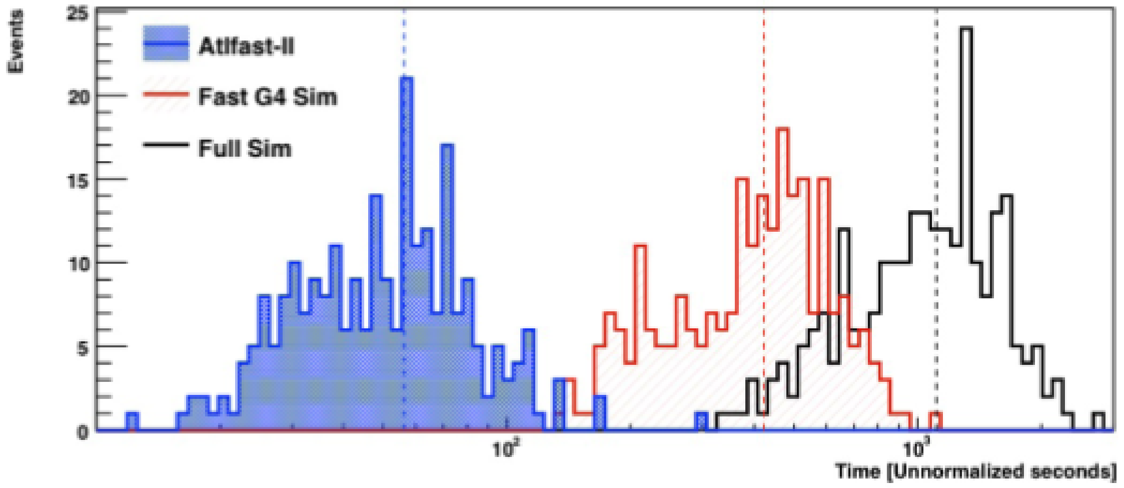
ATLAS Fast simulations

For some studies more statistics or faster turn-around are needed

CPU time consumption reduction by adopting fast simulations in critical productions

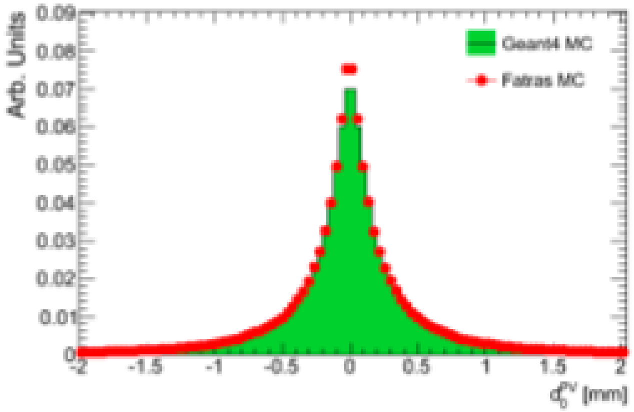
- ATLFAST-I
  - developed for physics parameter space scans — MC Truth
  - objects smeared by detector resolution
- ATLFAST-II
  - uses parameterised particle showers (FastCaloSim) and simplified detector description tracking (FATRAS)
- FAST G4
  - Standard full simulation uses pre-simulated shower libraries (energy 1 GeV->10 MeV) and single hit energy depositions at lower energy

ttbar events

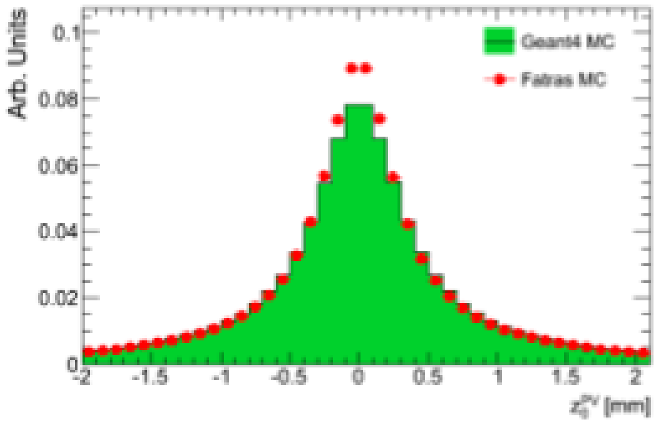


Fatras

***Fatras ID validation:* reconstructed impact parameter**
(minimum bias sample)



A bit narrower in Fatras
compared to full simulation
A good agreement nevertheless
with typical selection cuts (~1-2mm)



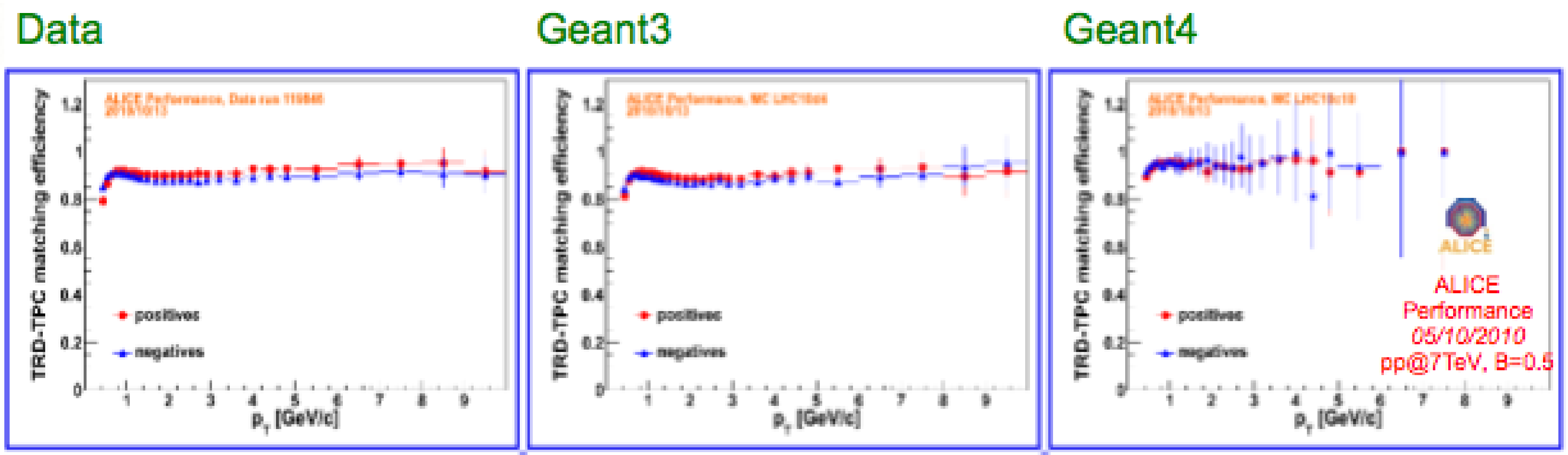
Timing performance:
(minimum bias sample)

Full simulation + digitization	***> 1 minute / event***
Fatras simulation (+digi)	***< 1 sec / event***

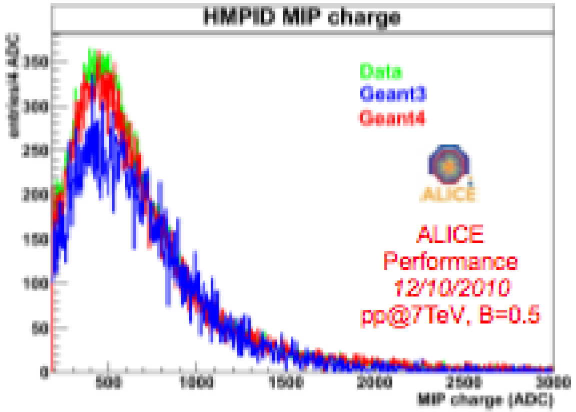
FastMCs
General Considerations

- The tuning of the full simulation requires a deep involvement from the experiment.
- This is even more true for the fast simulation. It requires a very deep knowledge of the detector and the physics goals to understand the kind of approximations to make without jeopardizing the physics results.
- In other words, the fastMC can only happen once the full simulation is understood and good agreement against data has been obtained.

VMC in Alice



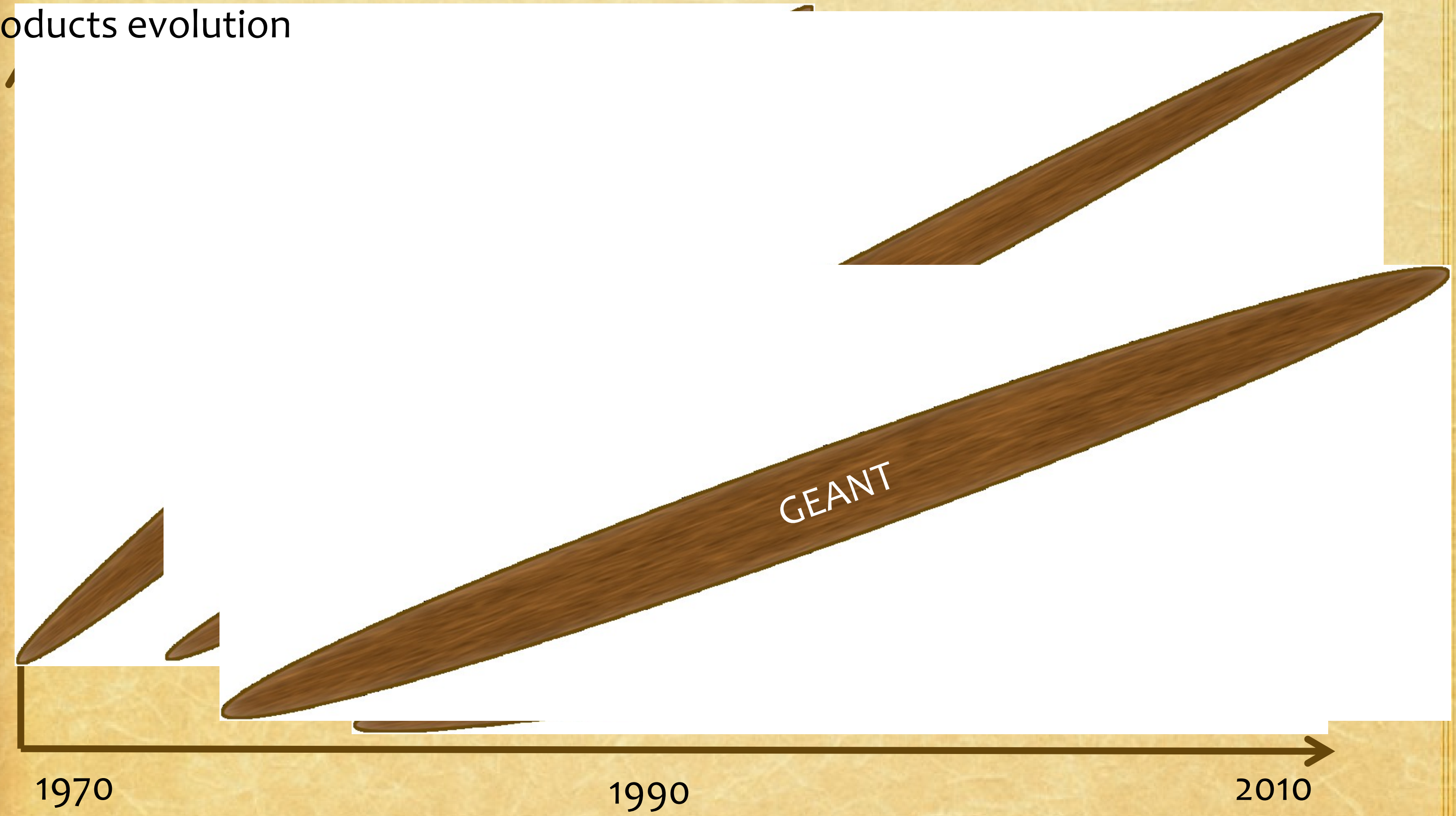
TPC-TRD prolongation efficiency



HMPID validation

The main MC families

Products evolution



The GEANT versions (I)

- The first version of GEANT appeared in 1974. It was a very simple framework for simulation between NA3, NA4 and Omega experiments. (about 5000 LOC)
- GEANT2 came in 1977 with more functions to control the initialisation, stepping phases (10000 LOC).
- GEANT3 came in 1981 in OPAL, then many experiments. It was a huge step. A powerful geometry system, electromagnetic physics based on EGS3/4 and interfaces with hadronic shower packages like Tatina, Gheisha and Fluka. (150000 LOC)

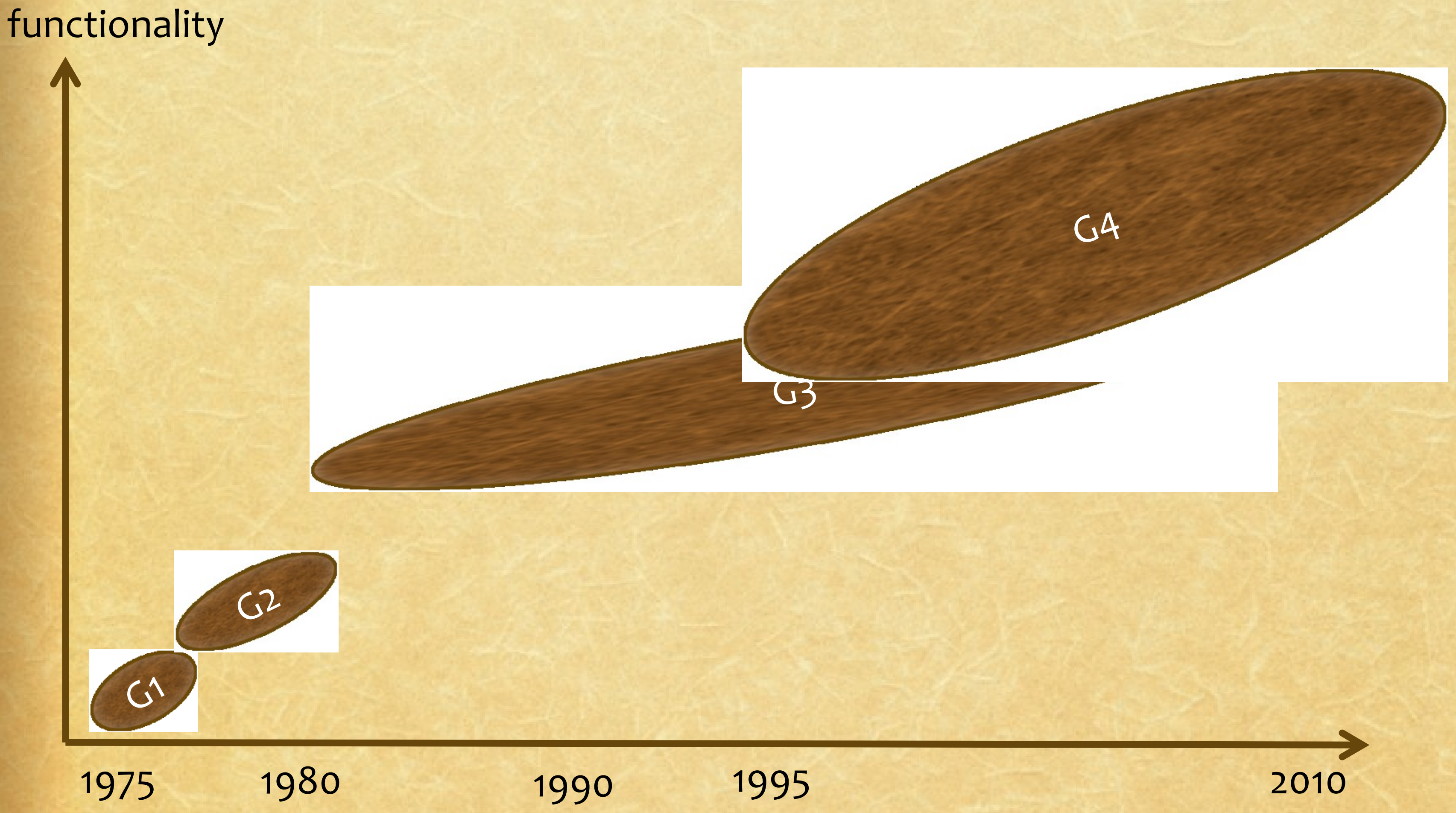
The GEANT versions (II)

GEANT4 came in 1995 following the directions taken by GEANT3 but written in C++. The geometry system was along the same lines as in GEANT3 and the electromagnetic physics was a continuation (with the same authors) of what was in GEANT3.

GEANT4 had a long list of developments and improvements in the physics sector, in particular hadronic physics and this work is going on.

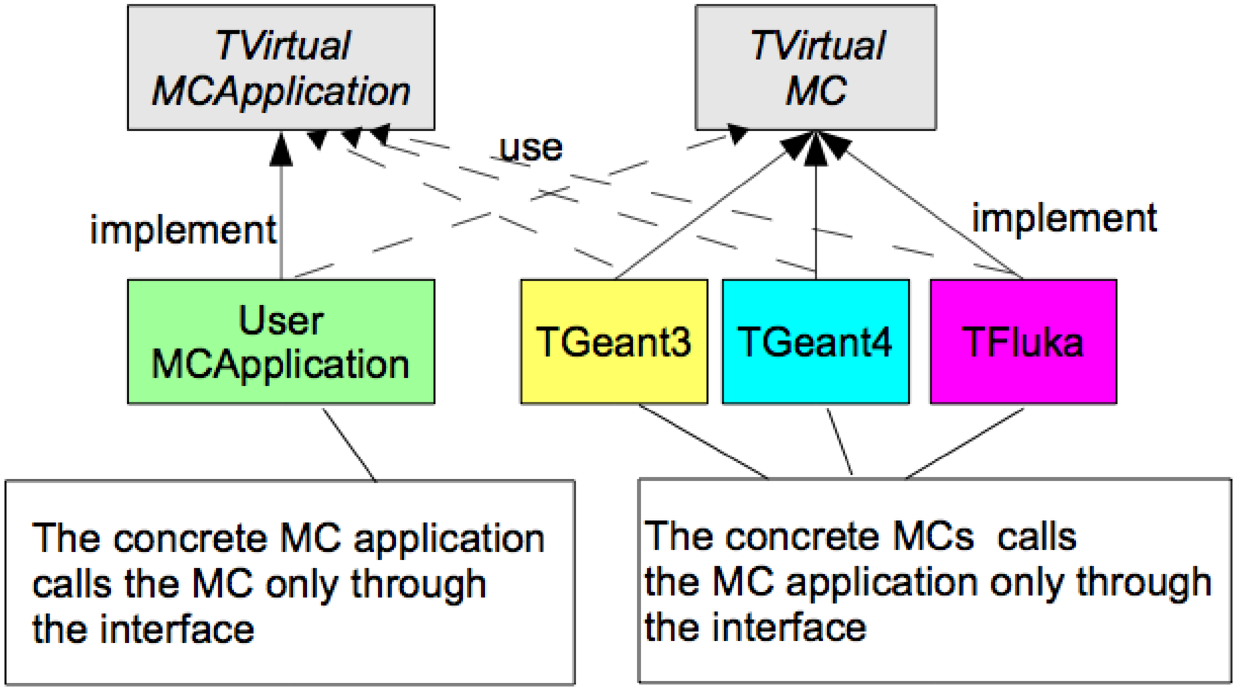
Recent reports from LHC have demonstrated the high quality of the simulations with GEANT4 physics.

The GEANT versions (III)



The VMC Design

- In VMC, we introduce the abstract interface both for the MC simulation program and for the user application
- In this way we decouple the dependence between the user code and the concrete MC


TVirtual MCApplication
TVirtual MC
use
implement
implement
User MCApplication
TGeant3
TGeant4
TFluka
The concrete MC application calls the MC only through the interface
The concrete MCs calls the MC application only through the interface

Comments on VMC

- A successful framework for testing different MC engines.
- It makes the experiment specific code independent of the transport engines.
- Useful as a migration tool too.
- Not designed for supporting multiple transport engines in the same job.

Strategy

Proposed Strategy (I)

- The current GEANT4 project continues with more focus on HEP problems, in a back compatible way and it will be the strong reference for the new project.
- A new project **GEANT** is launched with the idea of building a demonstration system in less than one year. It uses the GEANT4 physics at its core. The new project is developed in good synergy with G4 and ROOT, using most of its components, including the build system (at least at the beginning) and it offers new possibilities detailed in the next slides.

G4 Physics

- We start from the G4 physics because a considerable investment has been made there and the experiments seem to be happy in understanding the data with a right choice of the physics processes.
- However, a lot of work remains to be done to evolve the G4 Physics into a predictive tool.
  - See, for example the talk by Adam Para at CALOR2010, IHEP, Beijing, May 11,2010.
- The project could be an opportunity to refocus the developments and understand where to invest the efforts.

G4 Physics (II)

Some suggestions that I have already received

- Correctness of the physics modeling or at least a good documentation of the actual content and the degree of the veracity of the simulation should be on top of the list.
- Interaction with event generators can be vastly improved. In particular, given the expected new physics to be studied, one would like to be able to invoke some parton-level generators like COMPHEP, MADGRAPH, etc.. and hand over the partons to showering codes like PYTHIA and proceed to the detector simulation in some unified and consisted manner.
- Backgrounds (machine induced and physics-induced, i.e. overlays of other events) handling is an important feature.

Proposed Strategy (II)

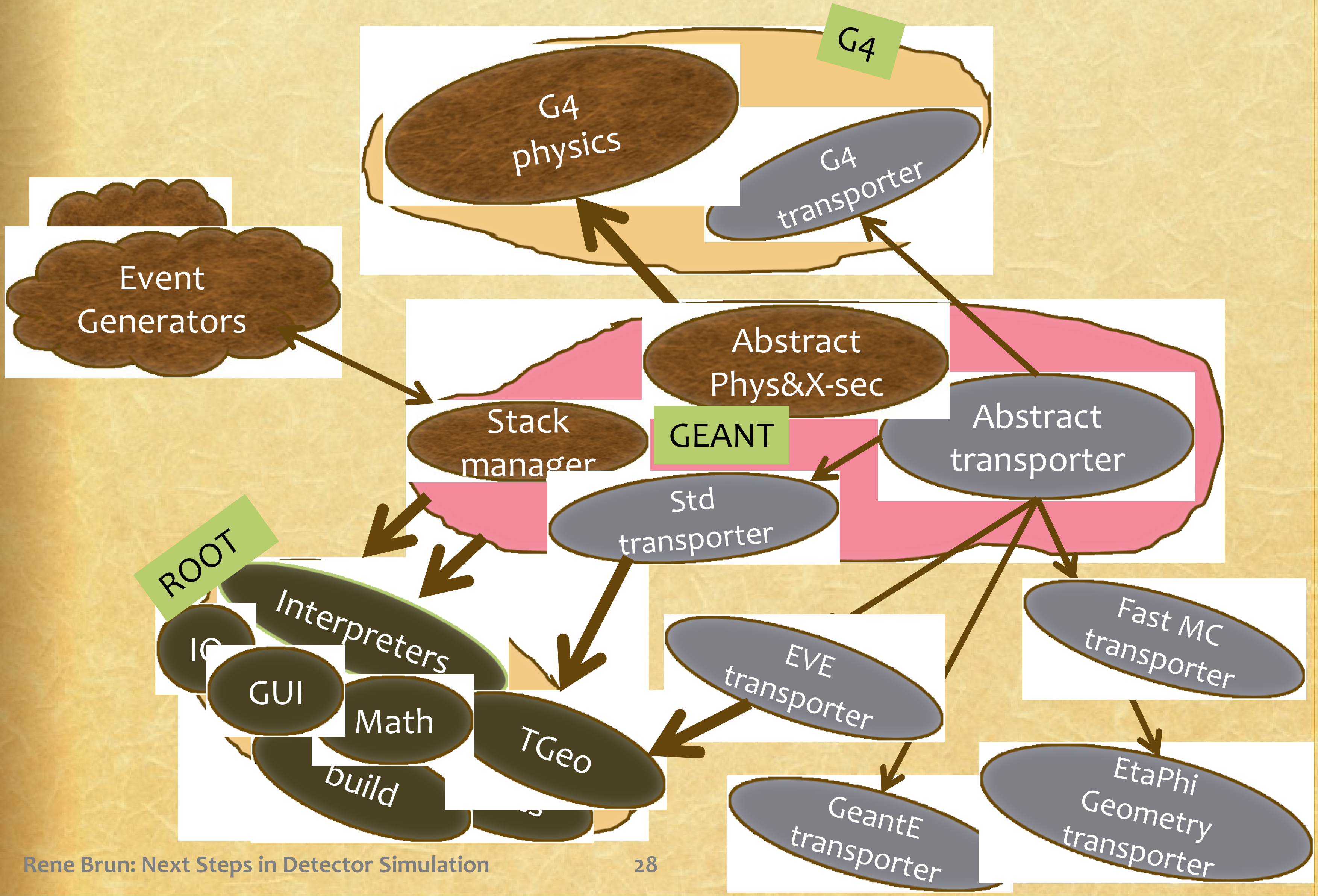
Because the goal is to show a demonstration prototype in a few months, we want to start with the most convenient mature tools and concentrate on the main new features.

Of course, these tools will evolve with time or better solutions may appear. The project will follow the developments in G4 and ROOT, eg build system, geometry, etc.

If the prototype is satisfactory, it will be further developed to reach a production quality level, and this may become GEANT version 5.

Architecture

New GEANT in one picture



Virtualisation of Transport

Changing physics model and cross sections should be easy and transparent.

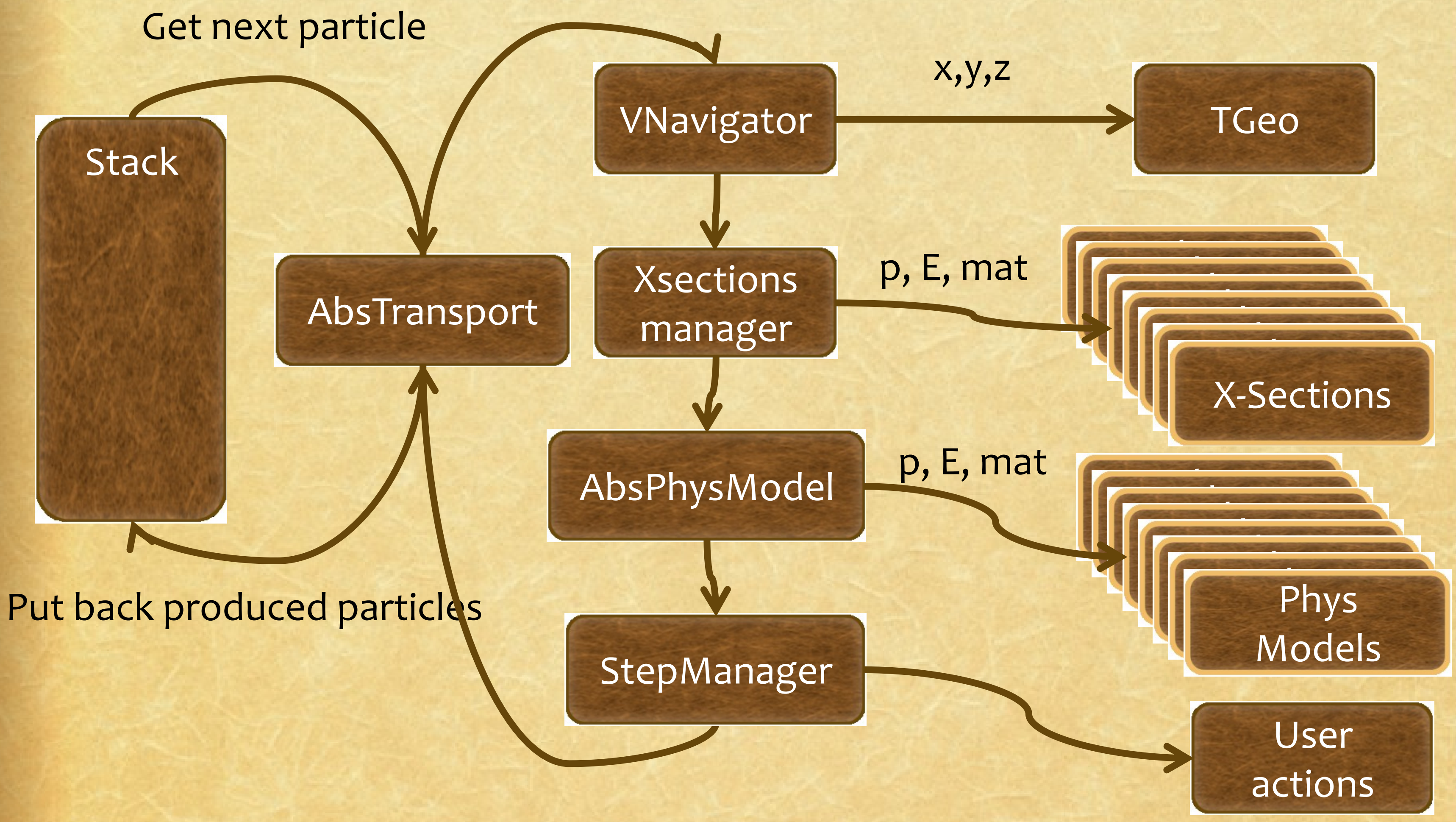
Transport model, cross sections and physics model should be all interfaced to virtual classes that are called by the framework.

More than one instance of each can be available at any moment during the running of the programme.

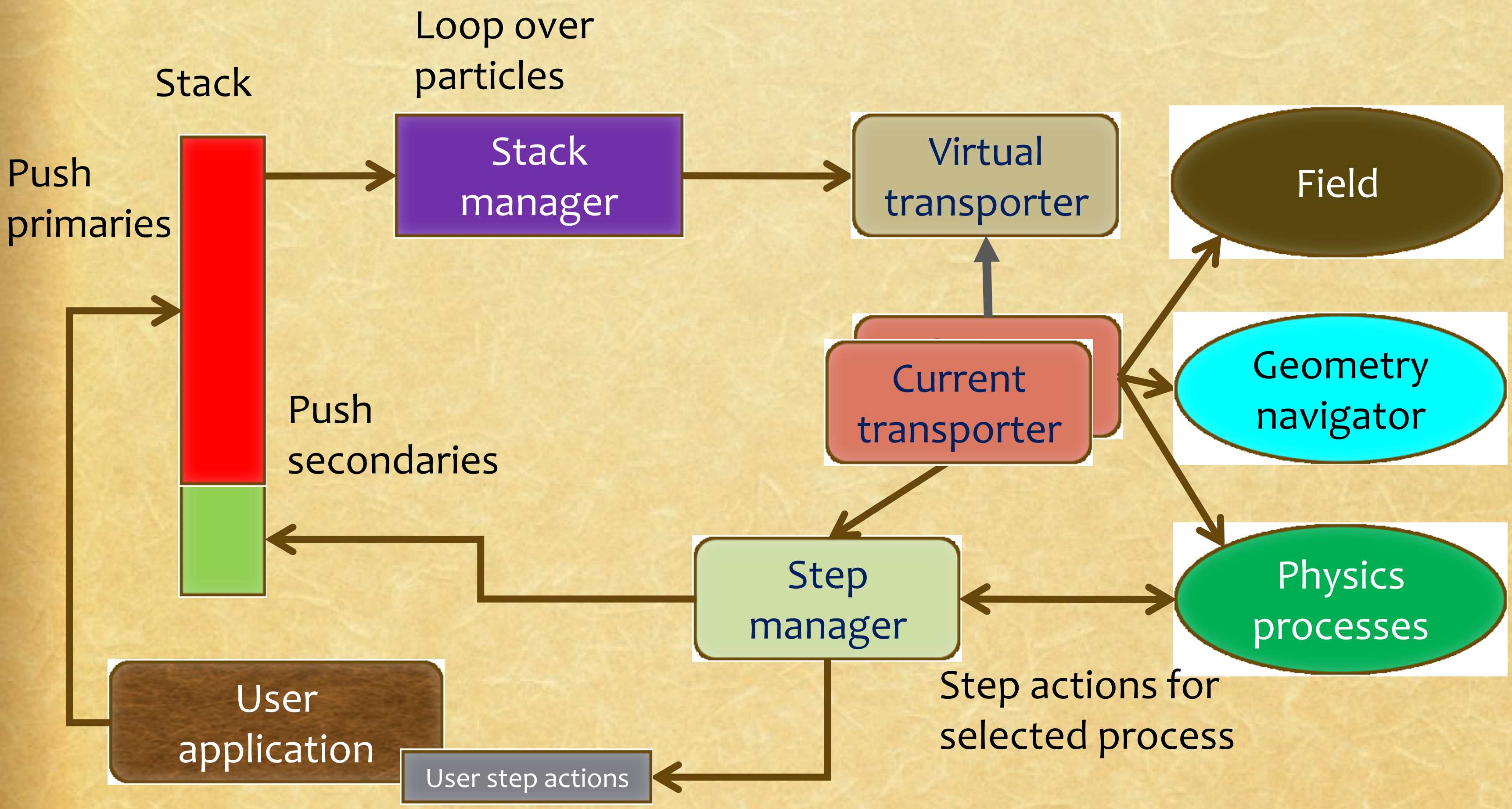
All elements should be virtualised and modular, even if we should pay attention to their internal coherence.

Reference implementations will be provided.

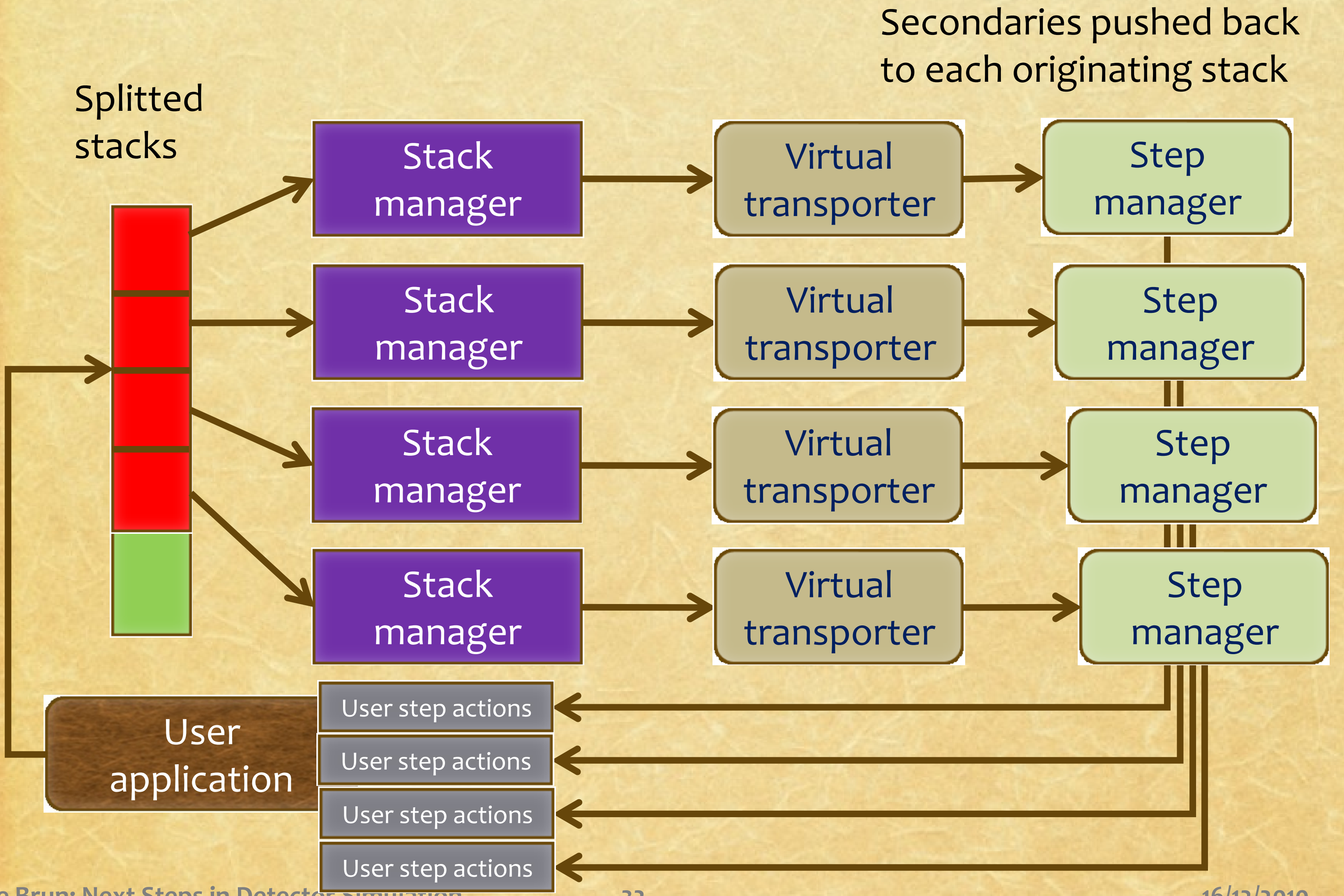
High level view



Event loop and stacking



Parallel approach



Abstract Transporter

We want **GEANT** to be easily customizable to several functions for full or/and fast simulation, used in reconstruction and event visualization.
Transporters are dynamically loaded and more than one transporter can be active.

Std Transporter

- This default transporter is provided as a GEANT library. It is designed for a full simulation application.
- This transporter uses the GEANT4 physics classes and the ROOT TGeo classes (or new solid geometry when it will be available)

G4 Transporter

- At the same level as the std Transporter, but based on the existing G4 geometry classes.
- As this may imply several changes in the existing G4 propagator, we need to collect experience with the std transporter before implementing it.

FastMC Transporters

- The ideas behind the current fast MCs (FastCMS, FATRAS) could be exploited and adapted to be consistent with the virtual transporter.
- This will require a direct participation from the people involved with these tools.
- Ideas could also be combined with the EtaPhi transporter.

GEANE Transporter

- GEANE is a special transporter that propagates a particle (forward or backward) on a pure statistics base without generating secondary particles.
- However, it computes a covariance matrix and hence the track parameters errors. This is specially useful in reconstruction programs or HLT systems.

EtaPhi Transporter

- An average geometry is generated by sampling techniques from the complex geometry (eg in pseudo-rapidity and phi cells).
- Several parameterized quantities are stored per cell (cross-sections, mag fields, energy loss, multiple-scattering).
- Tracking is performed in this fast geometry and the hits/digits stored in the complex geometry.
- This technique had already been tried with GEANT3, but machines did not have enough memory at the time.

EVE Transporter

- Useful for event displays or event generators.
- Using only the event kinematics, tracks are extrapolated in mag fields (conventional event displays like Fireworks or AliEve).
- A special version with a log(time) is used to visualize and understand events with short decay particles. This is useful for teaching purposes.

More on Transporters

- materials + medium properties, in relation with physics processes affecting the current particle. A transporter that ignores them is useful for event displays.

- magnetic field in relation with geometry. Typically while propagating in magnetic field, some (or all) sub-steps need to be vetoed by geometry (e.g corners in the trajectory). One may have a mode where the field is ignored (maybe correlated with particle momentum) and another one where the geometry is ignored (assumption that the current step AB does not cross boundaries). The second is useful if you have a coarse geometry approximated by voxel cells in eta-phi) .

- Geometry transporter - this may be used as switch from detailed to coarse geometry representations, maybe used in the same global simulation.

New architectures

- The transporters must be designed such that they can run in a highly parallel environment (many cores or/and GPUs).
- Support for event (or packet of events) and track level parallelism.
- This puts special requirements on the architecture, in particular the data structures.
- A GSI group and M.Kosov team are interested by this activity.

Interface with ROOT

We expect that a small fraction of the ROOT manpower will be used in the starting phase to setup the system of make interfaces with the sub-systems of ROOT that can exhibit important advantages at a small cost.

Build System

- We propose to use the ROOT build system. The GEANT installation will follow the same technique as all other modules of ROOT via module.mk, etc. When the ROOT build system will move to something new , we will follow too.
- The GEANT libraries will be dynamically loaded via the ROOT plug-in manager.
- The html documentation (with dynamic graphics) will use the THtml system.
- The web site geant.cern.ch will use the same CMS (Drupal). A forum page similar to the ROOT forum will be created.

SVN repository

- A SubVersion project will be created.
- GEANT will follow the ROOT coding conventions and use the same tools (checker, coverity, nightly builds).
- The new classes will be in the namespace GEANT.
- We will discuss with the G4 team/collaboration the best way to access the G4 physics classes to avoid divergent structures.
- New classes will not include a version number (G5)

Math Libraries

- We intend to use the functions from the ROOT Math libraries (random numbers, physics vectors, matrices).
- Best way to do it? Ifdefs in the physics code to support both options (CLHEP for G4). See CMS.
- Special look at random number generators with optimization of the sampling techniques.
- Participation to the performance evaluation.

Dictionaries and IO

- We will provide dictionaries for all imported classes such that we can:
  - -document them
  - -do I/O
  - -use them from the interpreters (CINT/PyRoot)
- Dictionaries will be in separate libs.
- Persistency will be provided for classes like cross-section tables that take time to compute.
- The configuration of a GEANT application will be via a config.C, config.py file

Event IO

- Kinematic classes must be streamable in an efficient way in ROOT Trees in split mode. Back compatible changes to HepMC are proposed to make the IO more efficient.
- Examples of hits/digits detector collections will be given.

Geometry

- Evaluation of the existing systems with the idea that the geometry can be used independently of the simulation, eg in reconstruction, event displays or event generators or fast MCs.
- investigate the interface with Physics and navigation, eg best techniques to approach and cross boundaries.
- support for existing geometry inputs (xml, mysql).
- Continue the discussions on a "geometry for solids".

TGeo Geometry

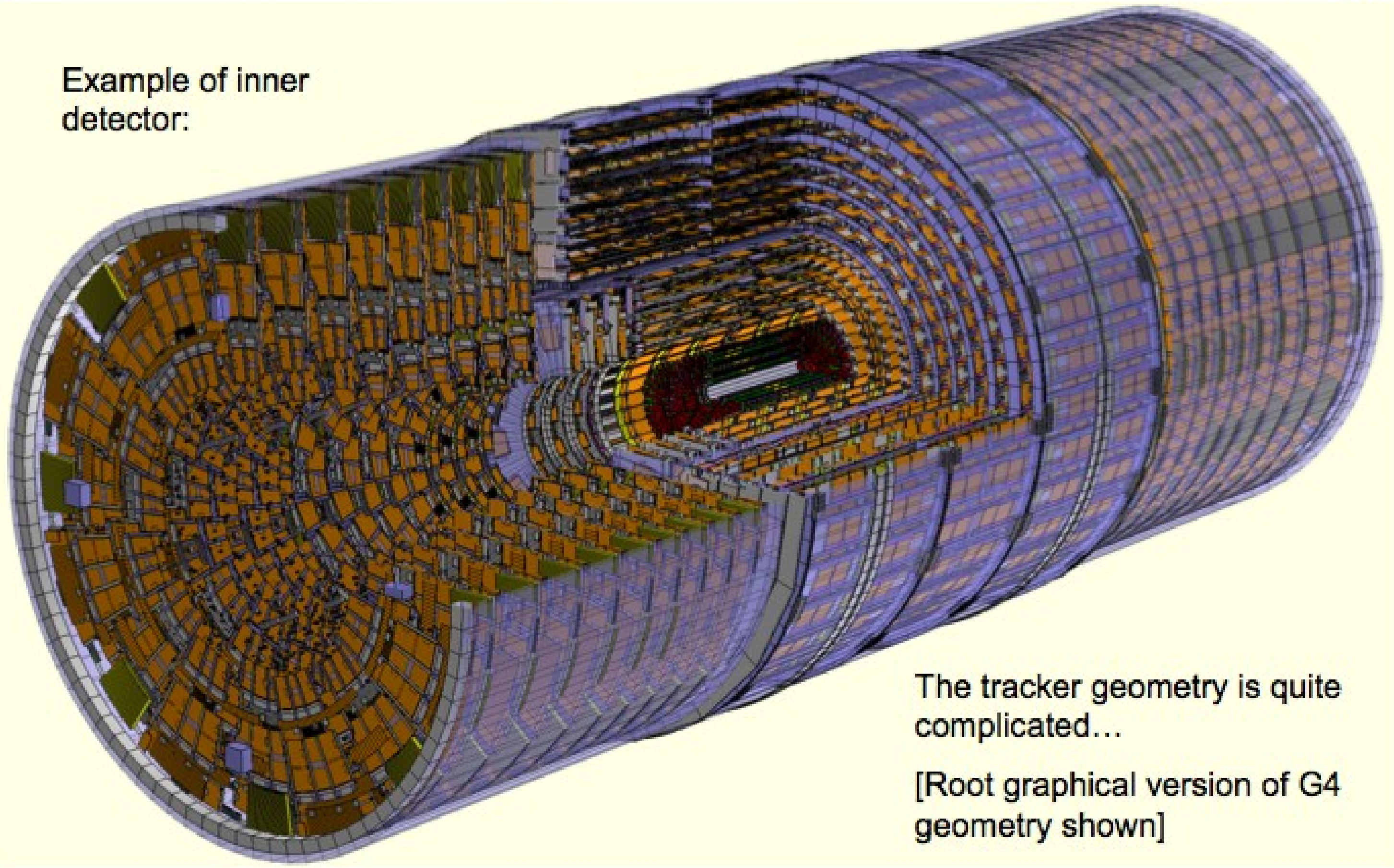
- Very powerful geometry used in simulation and reconstruction.
- Tested with most HEP detectors. 35 detectors tested in the ROOT nightly builds

ALEPH
ALICE
AMS
ATLAS
BABAR
BELLE
BES
BRAHMS
BTEV
CDF
CMS
D0
E907
FELIX
GANIL
HADES
HERMES
Integral
MINOS
NA35
NA47
NA49
PHENIX
PHOBOS
SDC
SLD
STAR
T2k-Nd280
Tesla
WA91

Geometry II

Overview of CMS

Example of inner detector:



The tracker geometry is quite complicated…

[Root graphical version of G4 geometry shown]

(4) 20 October, 2010 Mike Hildreth - CHEP 2010, Taipei, Taiwan

UNIVERSITY OF NOTRE DAME

Fermilab

GUI and 2D graphics

- A simple GUI will be provided to control the application such that :
  - User-created histograms can be visualized while the application is running.
  - Cross-section tables, field-maps can be visualized.
  - Infinity of possibilities.

Event displays

- ROOT provides a rich set of classes (EVE) for events and detector visualization.
- One must be able to show events directly from the event generators (with corresponding transporter).
- One must be able to use the graphics for debugging purposes.
- See many examples in AliEve, FireWorks, Linear collider DRUID (Manqui), fairroot,ilcroot,etc

Use of PROOF-Lite

- One obvious application of PROOF-Lite will be a GEANT application running packets of events in parallel on a multi-core system.
- This application will be one of the first users of the *parallel buffer merge algorithm* currently being designed/implemented.

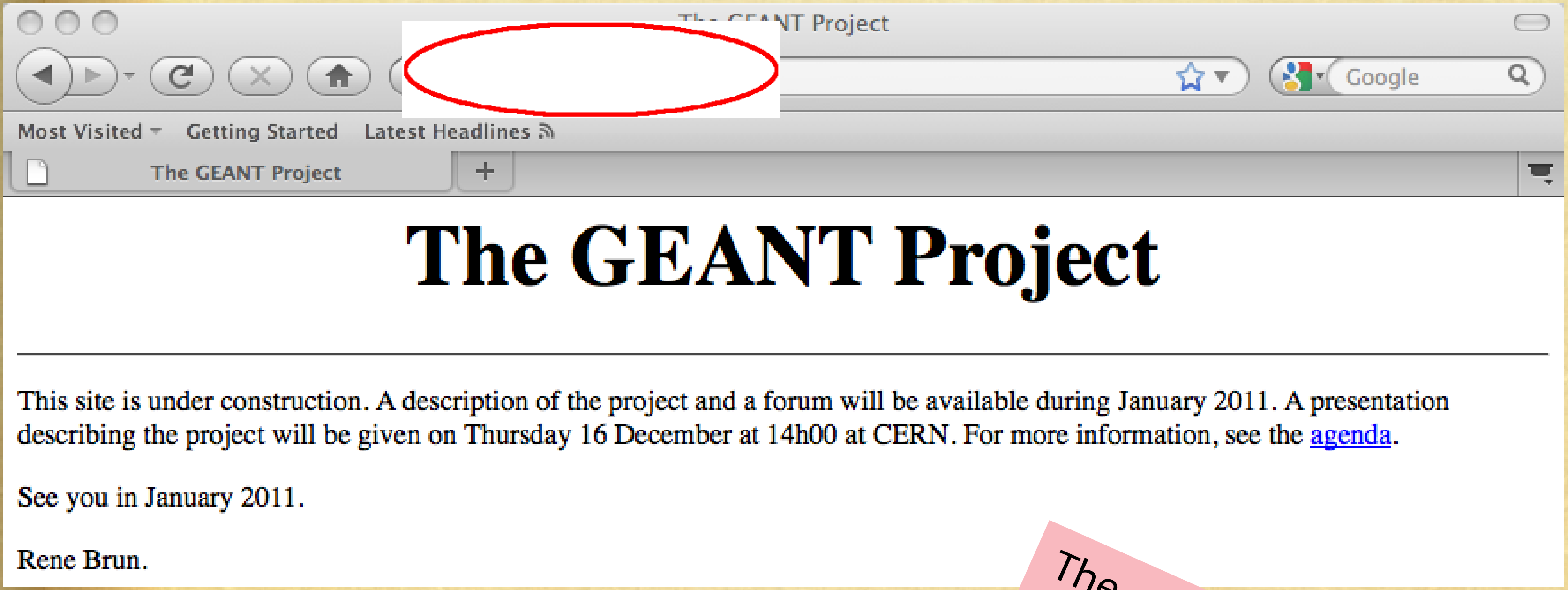
Benchmarking

Principle

- From the very beginning it is important to develop an infrastructure to monitor the progress with the prototype and compare results with G4 and FastMCs.
  - Comparing physics results
  - Comparing power of geometries
  - Comparing hits/digits
  - Validation procedure with experiments
  - Showing new features
  - Comparing Time Performance

Milestones

Project	Definition, explain goals,attract
Collaboration	Packages, teams, testers, validation
Global Design	Iterative, propose/critic/review
G4 Physics	Tech aspects, import model, evolution
Core	ROOT interface, plug-ins, IO, graphics
Web site	A la ROOT/drupal, forum, SVN
Geometry	Tgeo, G4, libs for solids, and evolution
Transporters	Std, G4, eve, fastMC, geane, etaphi
Stack Manager	Event & track level //
Kinematics	hepMC++



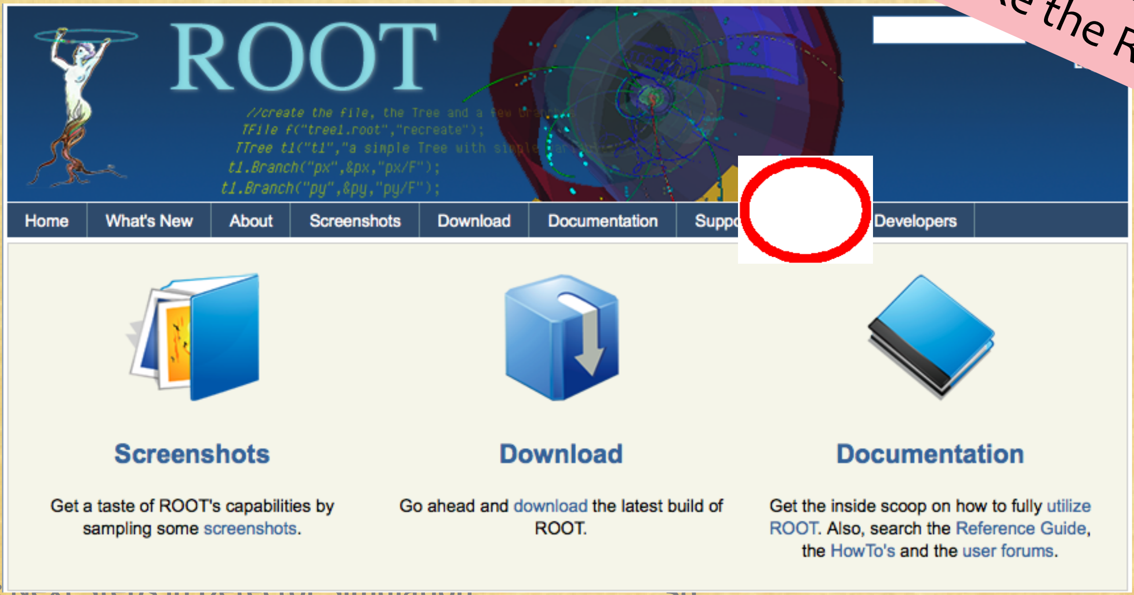
The GEANT Project

This site is under construction. A description of the project and a forum will be available during January 2011. A presentation describing the project will be given on Thursday 16 December at 14h00 at CERN. For more information, see the agenda.

See you in January 2011.

Rene Brun.

The new web site will be structured like the ROOT web site



Screenshots

Get a taste of ROOT's capabilities by sampling some screenshots.

Download

Go ahead and download the latest build of ROOT.

Documentation

Get the inside scoop on how to fully utilize ROOT. Also, search the Reference Guide, the HowTo's and the user forums.

… step by step …

- Now step by step build the project.
- We must be fast (time is running against me :☺) and slow to make sure we have a good design.
- This is going to require a lot of discussions with the experiments and the G4 team.
- Many thanks to *Sergio, Philippe, Livio, Pere, John, Federico* for the N+1 hot discussions creating this project.
- Many thanks to the many people who have already given their opinion and interest to participate.

It is now the time to make another

GEANT step