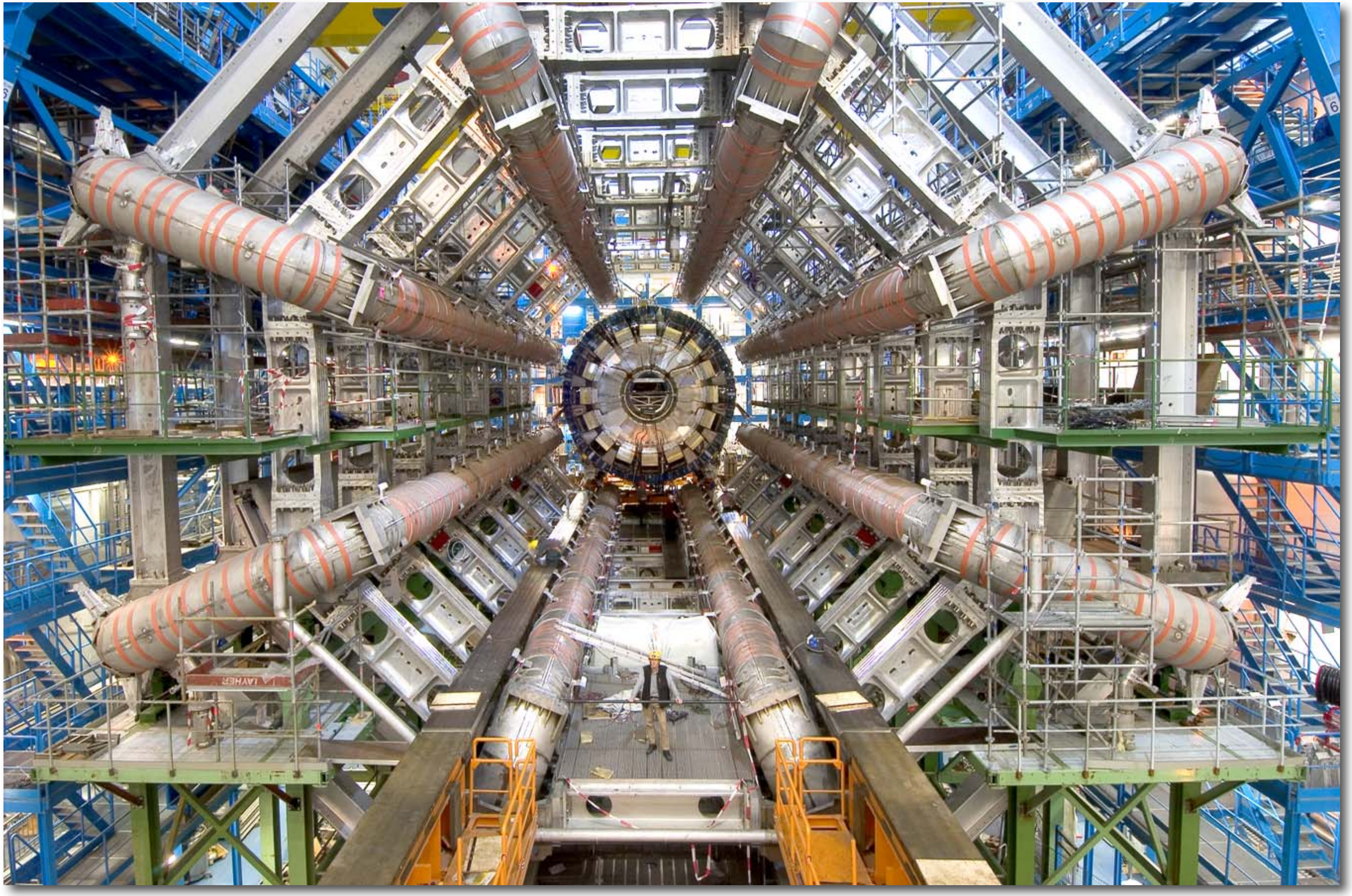
M. V. Gallas[(1)], A. Dell'Acqua [(1)], A. Di Simone [(1)], A. Rimoldi [(2)] , J.Boudreau [(3)], V. Tsulaia[(3)], D. Constanzo[(4)]

1- CERN (Switzerland), 2- University of Pavia & INFN (Italy), 3- University of Pittsburgh (USA), 4 - University of Sheffield (UK)

CHEP06 Computing in High Energy Physics International Conference, Mumbai, 13-17 February 2006



ATLAS Detector Simulation: an LHC challenge

ATLAS

detector

A reality on the way to be ready in the LHC tunnel (CERN,Geneva) after 15 Years of preparation development and construction

collaboration

34 countries, 152 institutions and 1770 scientific authors

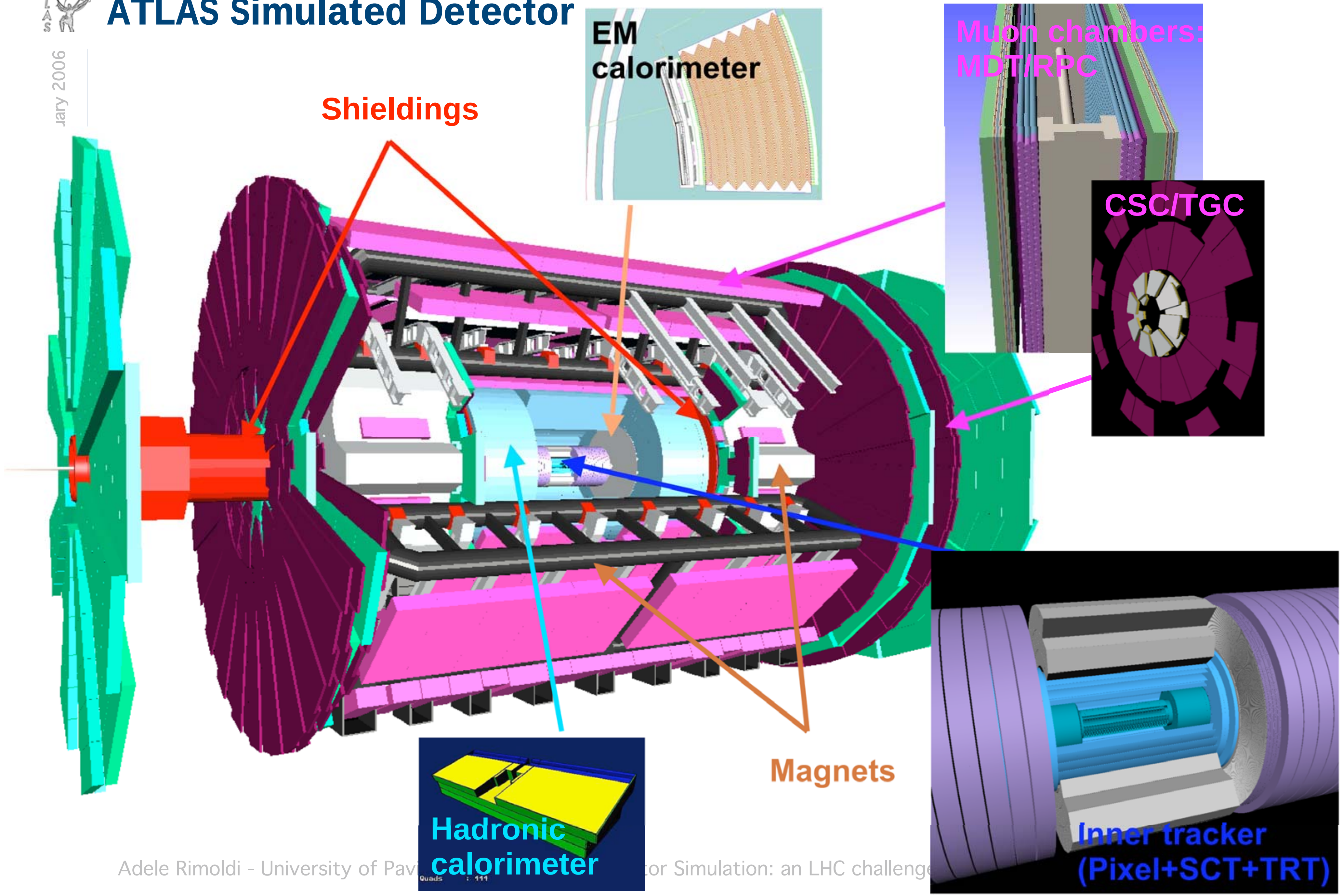
offline software handled by a common framework

- Generators
- Simulation
- Digitization
- Reconstruction (make ESD)
  this reconstructs particles, jets, tracks etc.
- AOD building– condense the ESD
- Physics Analysis

This talk is about

- G4Atlas simulation
- ATLAS Detector Description
- Job control
- The Python G4Atlas interface
- Examples
- Validation
- Performance measurements
- Conclusions and perspectives

ATLAS Simulated Detector



G4Atlas simulation

ATLAS offline software

- **common framework** for event processing (ATHENA) where to run applications
- **Applications**
  - sets of services and algorithms assembled and configured @ run-time
  - steered using jobOption scripts written in Python
- Python
  - as OO-scripting language
    - simple and intuitive
    - robust in an interactive use
    - introspection mechanism
      - user can interrogate the object, about type and internals
    - can add introspection and interactivity to the C++ layer

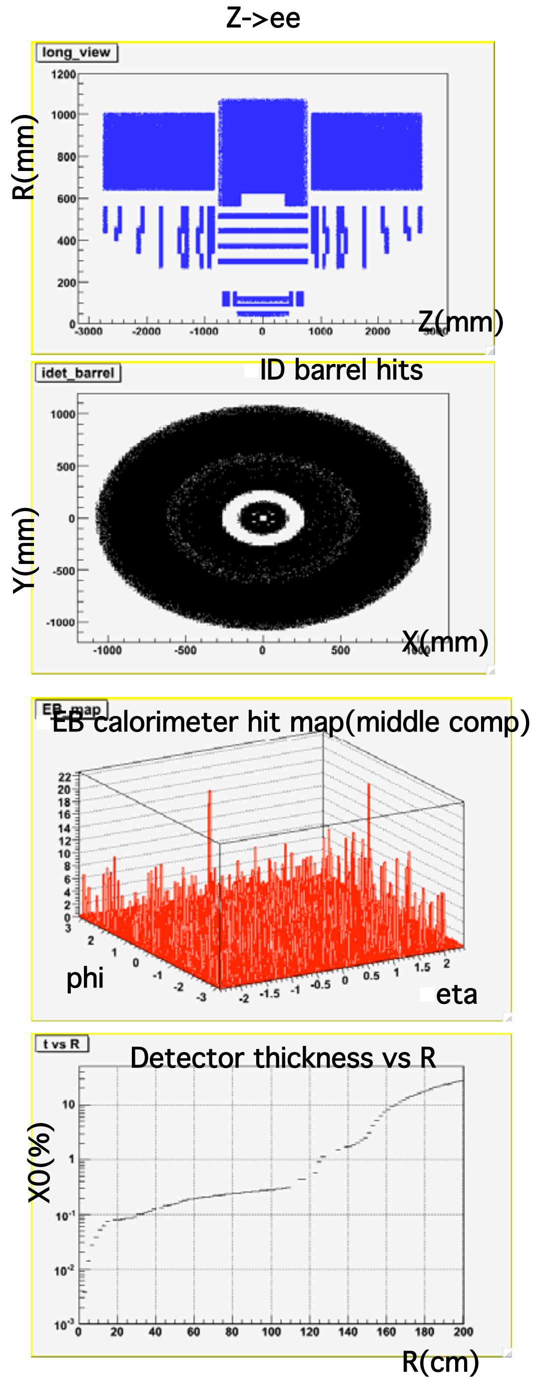
G4Atlas

- ATLAS simulation application
- Full featured OO **Geant4** simulation suite
- based on
  - dynamic loading and action-on-demand
  - All user requested functionality is added using plug-in modules
- PyG4Atlas
  - flexibility for configuring different setups
  - interactivity
  - introspection
- Robustness and flexibility
  - daily user-specific requests and different geometry configurations set @run-time
  - Robustness proved with negligible failures target after validation process is 1/10000
- massive GRID productions

2006 new production just started
2005 latest ATLAS workshop (8.6 Mevents)
2005 Combined Test Beam ->performance studies (4 Mevents)
2004 large scale production (DC2) (12 Mevents)

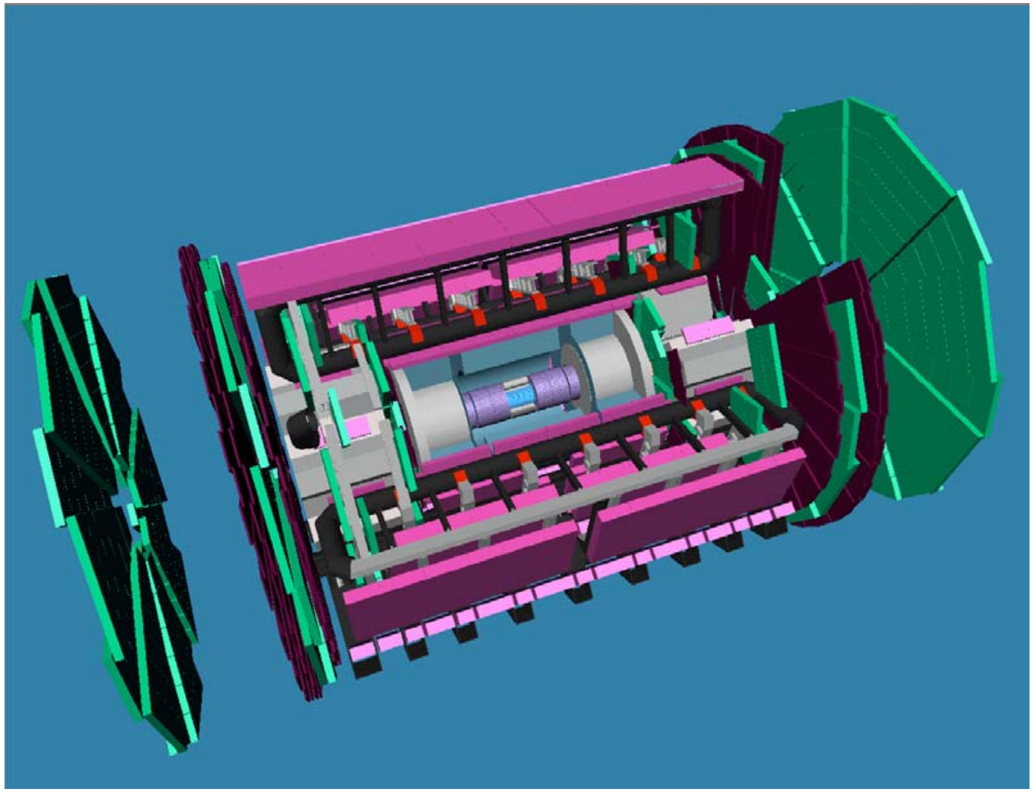
G4Atlas simulation

- Different setups handled similarly
  - full ATLAS simulation
  - cosmic-ray setups
  - Combined Test Beam and standalone test beams
- Consistency and validation effort kept throughout all the applications
  - the user can switch among applications with minimal effort
  - Non-ideal detector description in progress
  - geometry for detector as installed, misalignment, material services
- Validation tools
  - Algorithms and test tools
    - G4AtlasTest steering+ subalgorithms
      - to access detector “hits”
      - to perform material scans
      - Radiation and interaction length integrators



ATLAS detector description

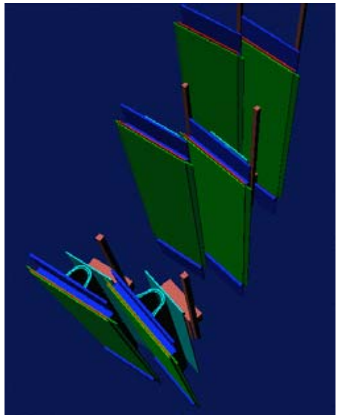
- Description of the complex geometry is decoupled from the simulation framework (G4Atlas)
  - two hierarchical geometry trees in memory at the same time: "GeoModel" and "Geant4"
- GeoModel provides a transient geometry representation built from primary numbers (OracleDB & versioning) and alignment constants
- Same geometry used in
  - Simulation
  - Digitization
  - reconstruction jobs
- Built @ runtime
- GeoModel description
  - optimized for a large number of volumes (~10^6) (use of parameterized volumes)
  - translated into a G4 geometry at the initialization phase and placed within re-sizeable and moveable G4 envelopes



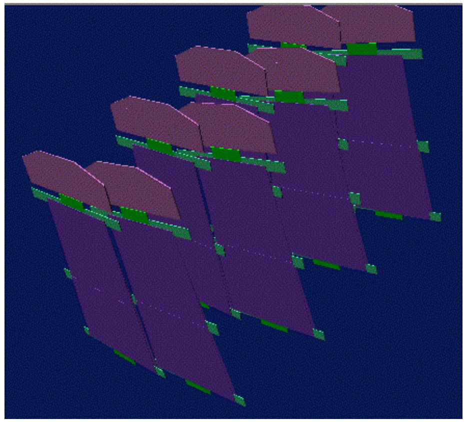
Task	Total memory (MBytes)
Pixel	5,6
SCT	9,1
TRT	3,1
InDet material	1,0
LAr	54,4
Tile	1,1
Muon	21,3
GeoModel total	95,7

ATLAS detector description

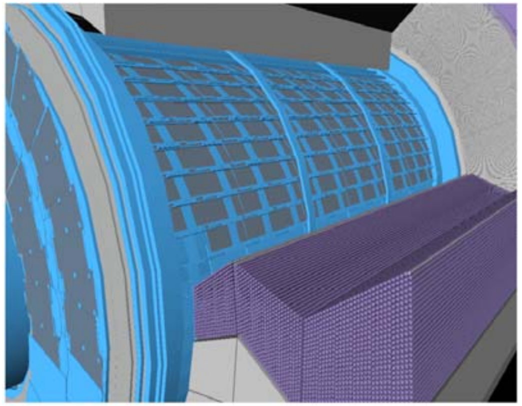
PIXEL : ~ 6000 volumes



SCT: ~40500 volumes

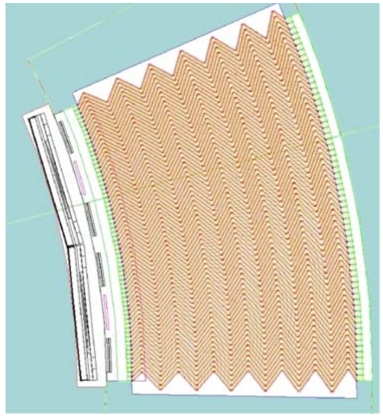


TRT: ~300000 volumes



(mostly parameterized)

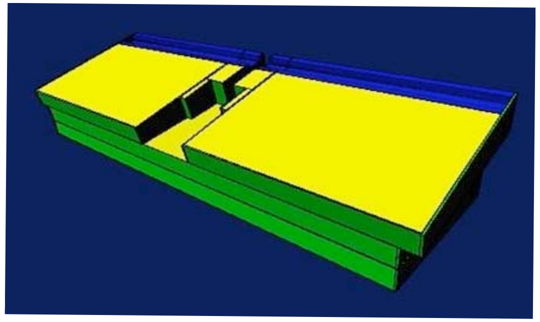
LAr Calo: ~142500 volumes



(partly parameterized)

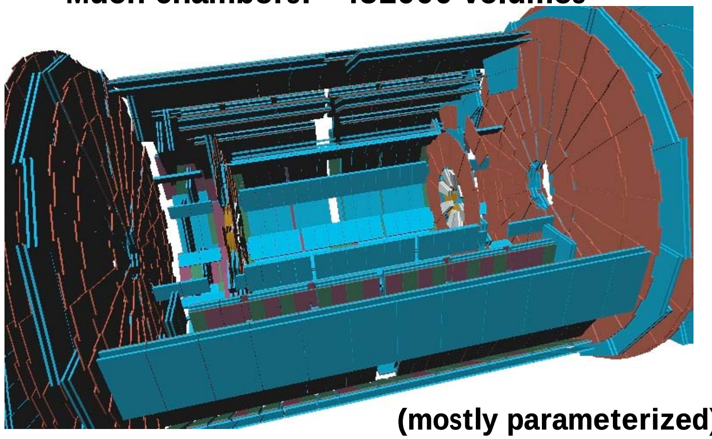
- Gallery of sub-detector geometries as seeing by G4ATLAS

Tile Calorimeter: ~8500 volumes



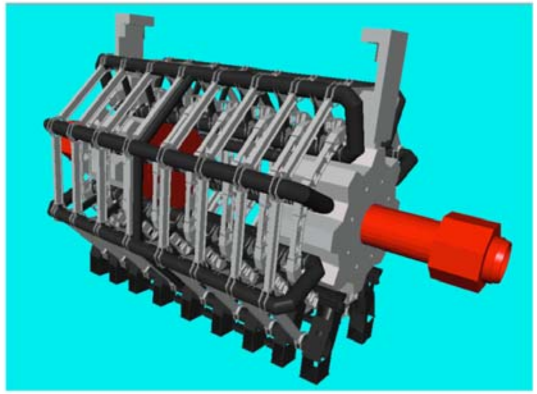
(mostly parameterized)

Muon chambers: ~451000 volumes



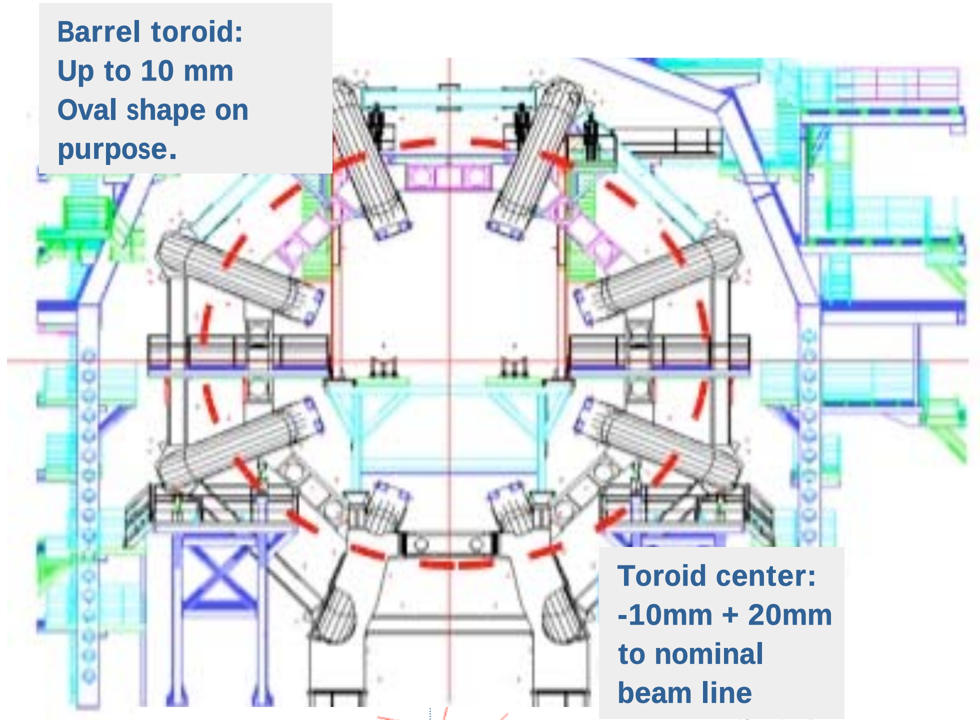
(mostly parameterized)

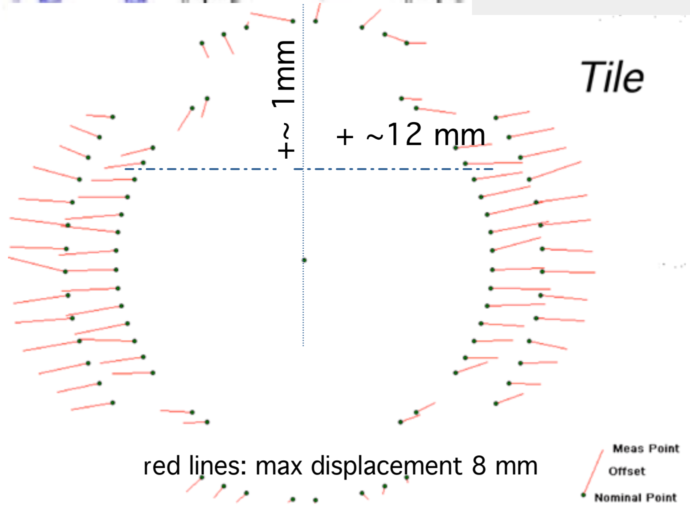
Toroids ~ 1000 volumes



ATLAS detector description

- Memory usage ~90 Mbytes (in spite of the optimizations) for full ATLAS
- GeoModel is also used by the CTB (2004) test beam (validation)
- All the past productions were done using ideal detectors with nominal positions
- New productions will include:
  - setup for detector as installed
  - misalignment
- Import all the misalignments and deformations into GeoModel
  - time variation associated with run number

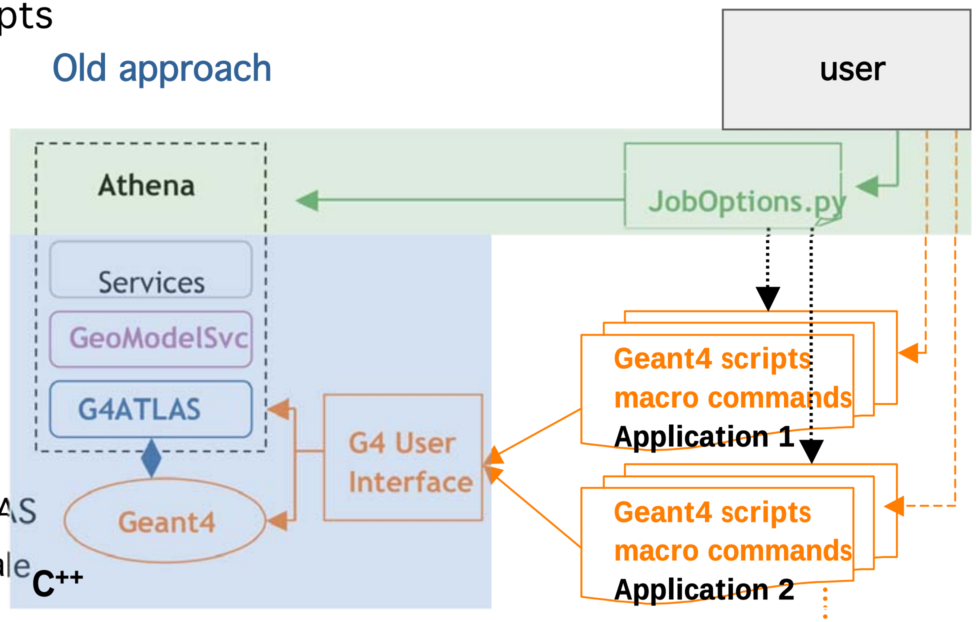




Job Control

- ATLAS offline software
  - all in the ATHENA based framework, born on the grounds of Gaudi
- ATHENA
  - uses the Python language as a front-end to the final users
    - JobOptions file(s) handle jobs via python scripts
      - Input/output, configurations, algorithms to run ..
- GEANT4
  - No native interface to python
    - simulation configuration is passed through specific command lines to Geant4
    - Grouping in macro files
      - Directly controlled by the user
      - Diverging number in a complex application as ATLAS
      - Maintenance quite impossible on the long time scale

Old approach



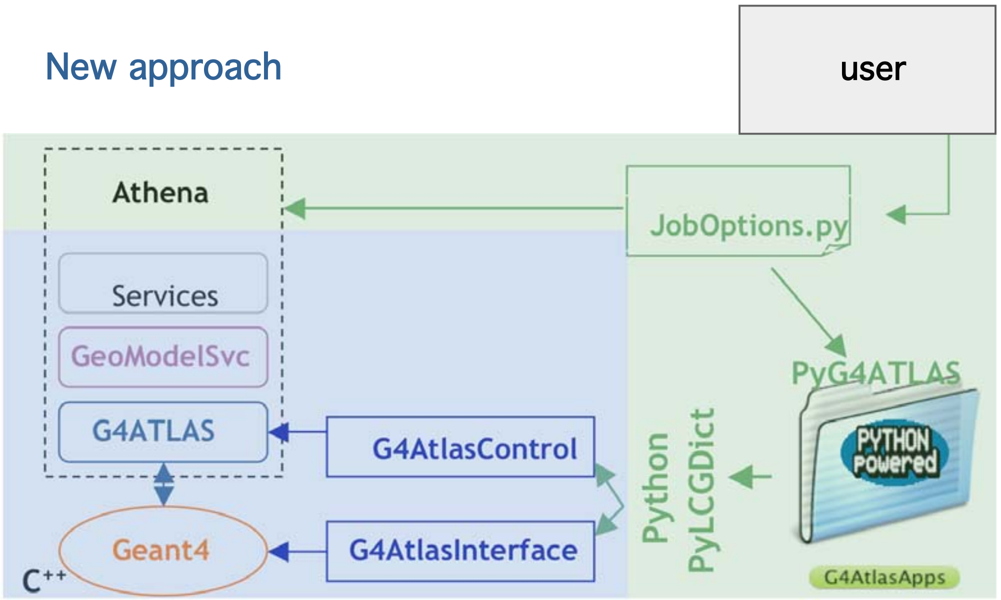
- Python layer added and developed
  - From the ATHENA prompt a simulation job can be controlled and configured @runtime

The Python G4Atlas interface

- G4Atlas has a Python interface, PyG4Atlas, which
  - provides enormous flexibility for configuring and maintain different setups
  - improves the usability by adding interactivity and introspection
- PyG4Atlas is a Python module that uses the **PyLCGDict** binding to the LCG C^{++} dictionaries

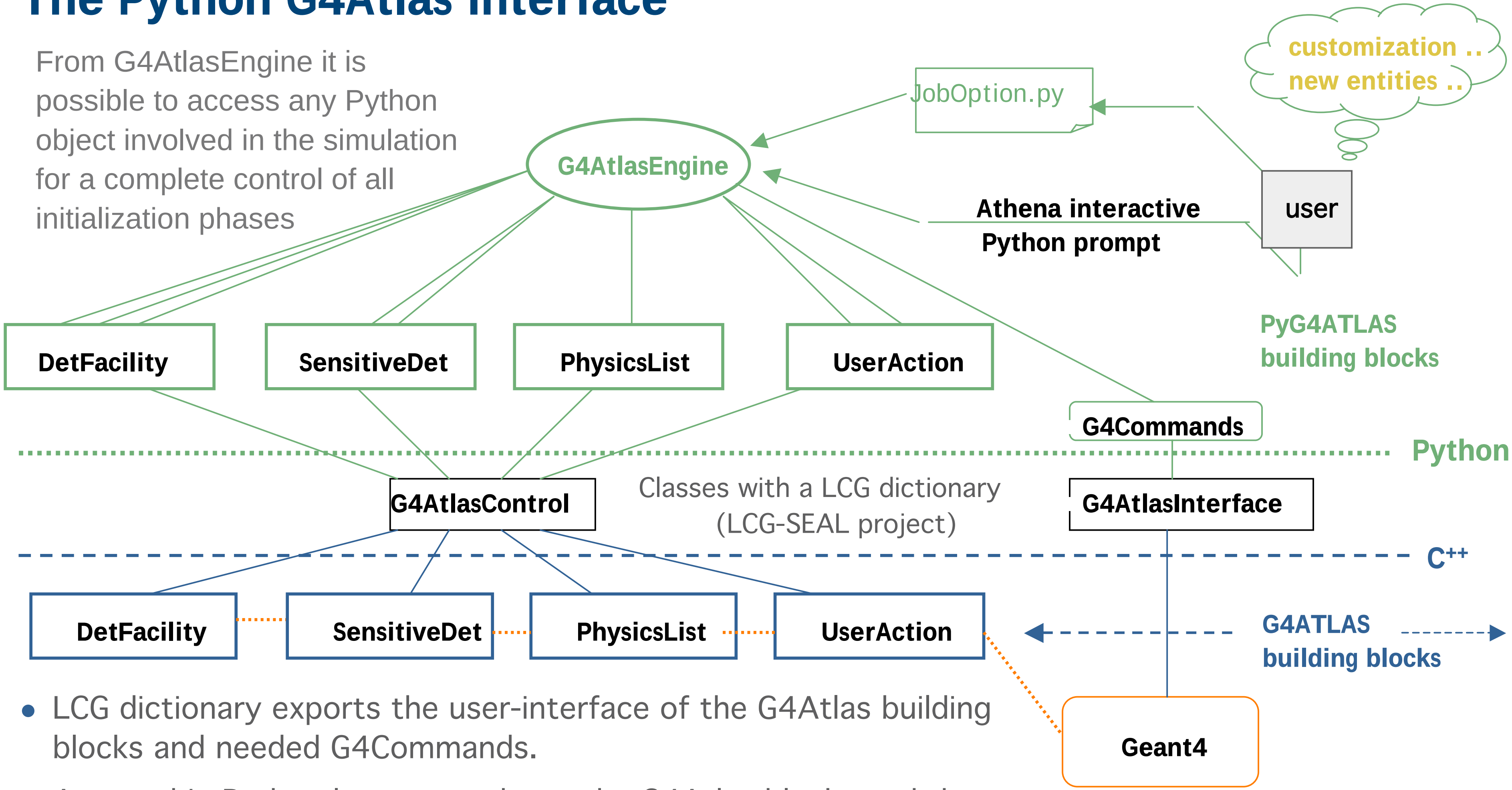
- Macro files dropped
  - Substitution with G4AtlasControl classes with public interfaces exported to Python
- all G4 Commands are available
  - G4AtlasInterface
- SEAL PyLCGDict mechanism used to export C++ classes to Python
- G4AtlasApps
  - uses the created dictionaries
  - Can profit of additional ATLAS or external dictionaries (CLHEPDict)

New approach



The Python G4Atlas interface

From G4AtlasEngine it is possible to access any Python object involved in the simulation for a complete control of all initialization phases

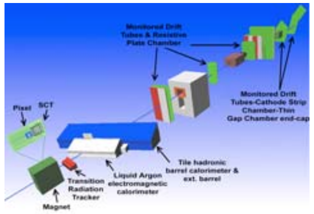


- LCG dictionary exports the user-interface of the G4Atlas building blocks and needed G4Commands.
- A very thin Python layer reproduces the G4Atlas blocks and the user can access the simulation engine, profits from the introspection and customizes it
- PyG4Atlas always selectively import Python modules, libraries, and dictionaries depending on the user requirements

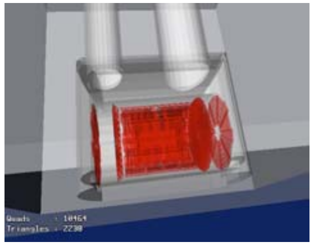
G4Atlas applications

- The simulation framework offers a set of pre-configured simulation applications:

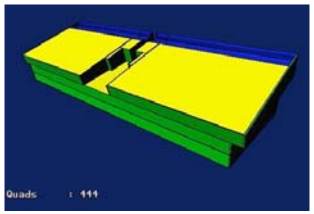
Full ATLAS simulation



Combined Test Beam



Cosmic ray setups



Old stand-alone Tile test beams

G4AtlasApps is a Python-coded package that can set up and run, within Athena, the Geant4 ATLAS full simulation or any other ATLAS G4 simulation like the ATLAS barrel Combined Test Beam or the different cosmic ray setups

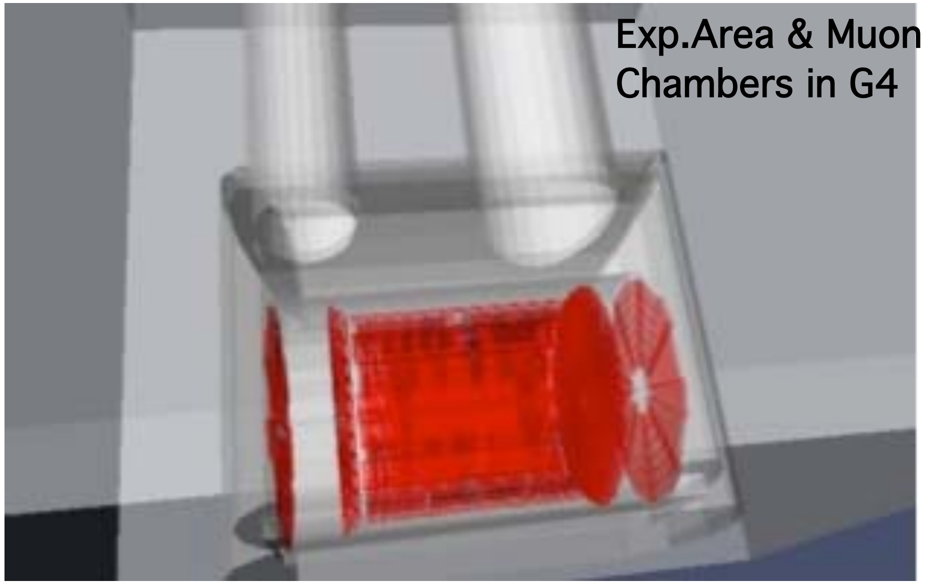
- Several layouts are available in each application
- Sub-detector specific studies and user customizations can be easily achieved from the pre-configured applications
- The pre-configured applications are exercised daily (automatic nightly tests)
- The feedback (improvements and new features) from the users is included in the pre-configured applications and maintained centrally for the community

Examples

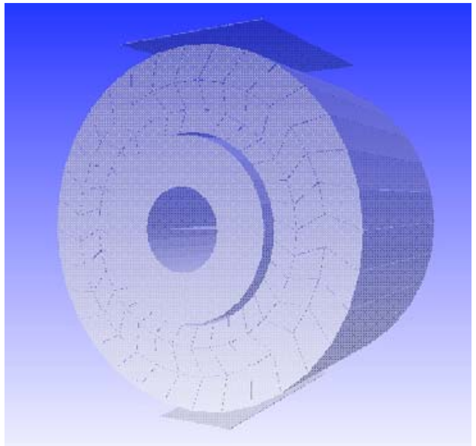
Cosmic-ray simulation

- During 2006 the priority is the cosmic-ray data
- Full support for the cosmic-ray simulation is in place:
  - description of the exp.area: rock overburden and surface buildings
  - primary cosmics from dedicated CosmicGenerator (only μ)

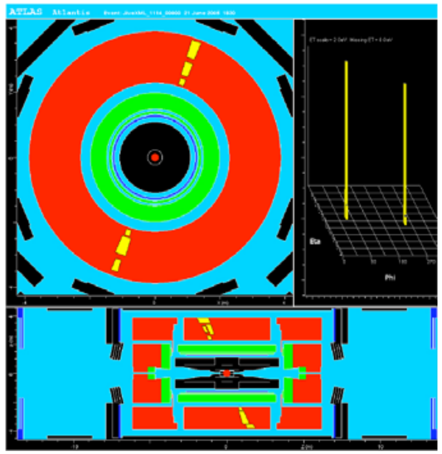
- each envelope is used as a scoring layer
  - Particles at the entrance are recorded
  - the external envelope (muons) saves the particles propagated through the rock overburden, before entering into ATLAS detector -> re-start the simulation from that point



Exp.Area & Muon Chambers in G4



Experimental setup for the Inner detector cosmic-ray studies in the surface not in the pit.



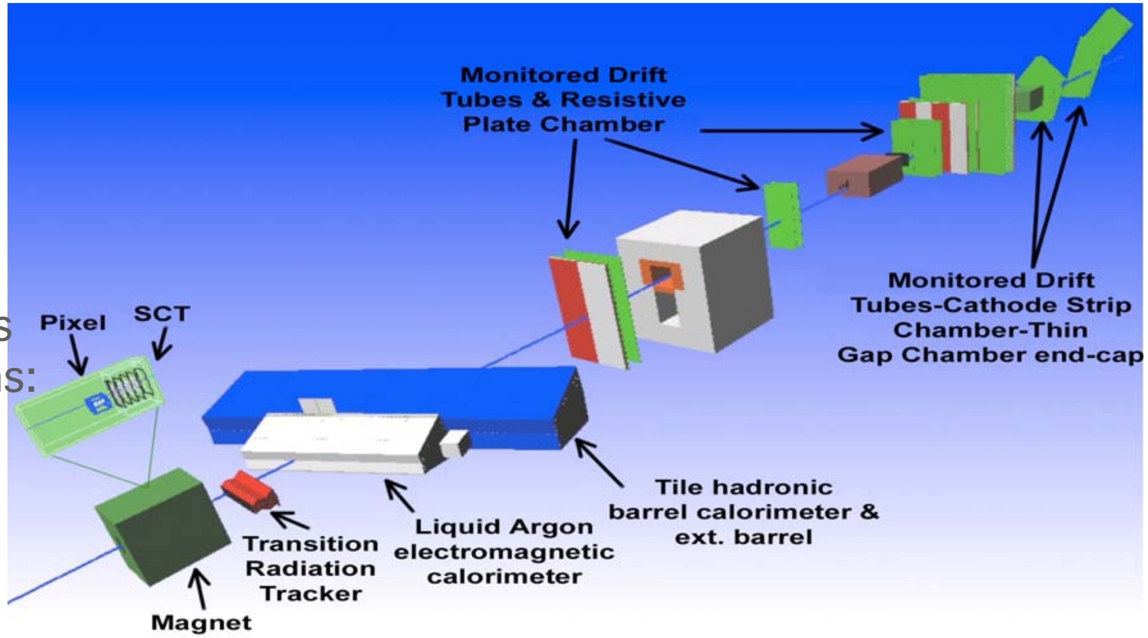
Cosmic in the Tile Barrel, this is not simulation!!

Examples

Combined Test Beam

- CTB is a big source for **detector performance studies** and **physics validation** at LHC
- The simulation infrastructure deals with all the different configurations:
  - combined mode
  - photon beams
  - material studies
  - η scans (0-1.2)
  - calibration
  - different ancillary detectors

(layout evolutions over the 24 weeks of run)



- 90 million events, 4.5 TB
- 22 million events in combined mode

$e^{\pm}$, $\pi^{\pm}$, $\mu^{\pm}$ from 1 to 350 Gev

- GRID production of 200 validated runs
- 4 M events

- The different layouts are set specifying only the run number
- Single particle generator is used in most of the cases although Hijing can be used as well for speed up material studies

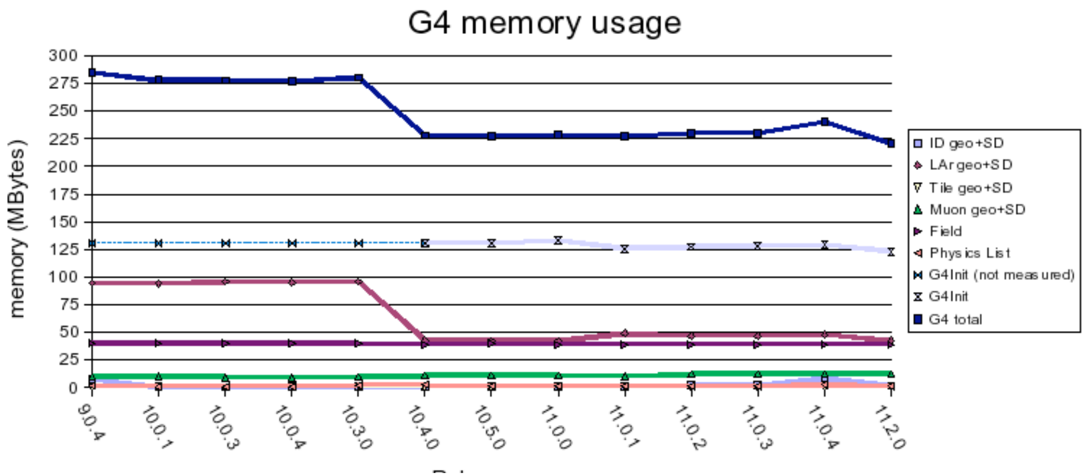
Validation

- Process parallel to the simulation development
- The aim is to spot as soon as possible any non optimal performance, internal inconsistency or inaccurate description of the detectors or physical process
  - G4 charged particle propagation in a magnetic field
    - Momentum shift observed in ID due to the default G4 tracking parameters
- Split in three main domains:
  1. continuous measurement of the performance in terms of the CPU time and memory consumption.
  2. comparison with real data from:
     - old stand-alone test beams for the different sub-detectors
     - ATLAS combined test beam (CTB-2004)
     - in a near future cosmic ray tests
  3. physics performance studies by reconstruction of full physical events

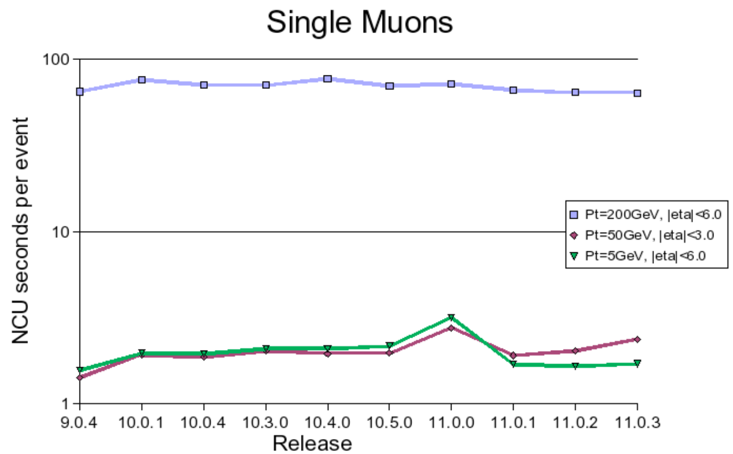
Validation

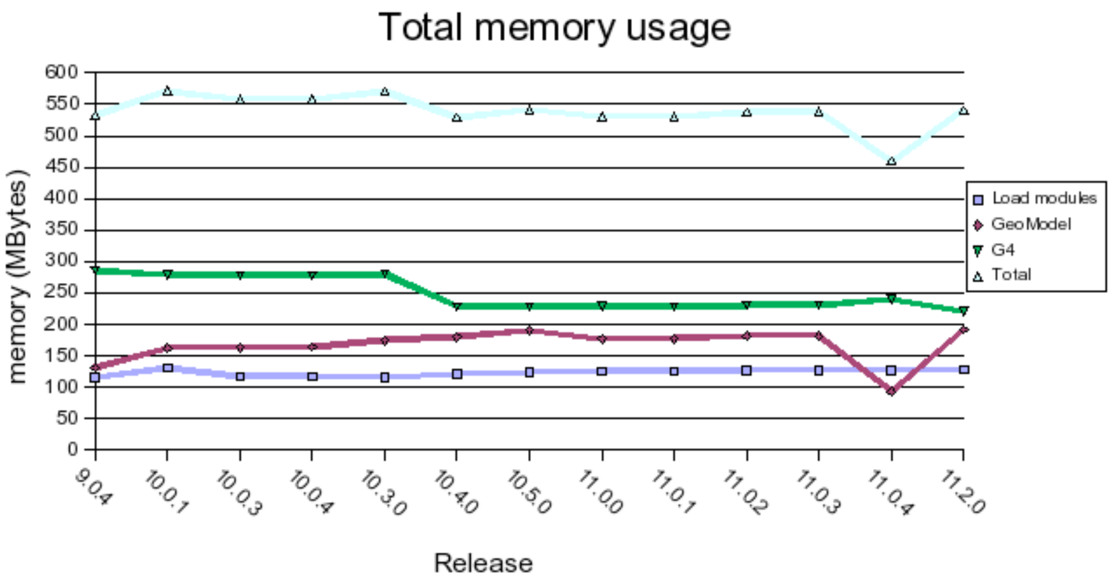
Performance

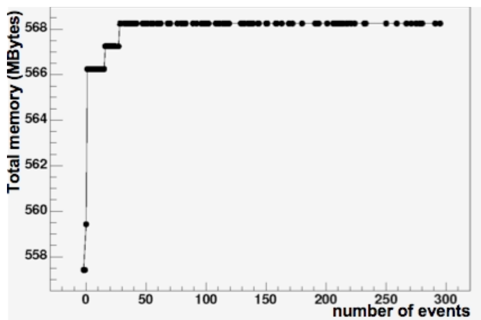
- CPU time per event and memory usage at run-time is daily monitored



- Detailed measurements for single particles and physical events are done in each new release



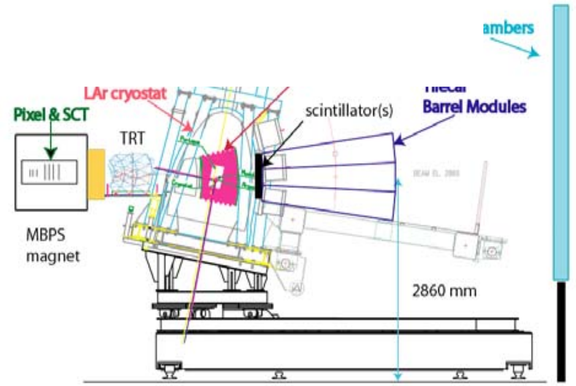


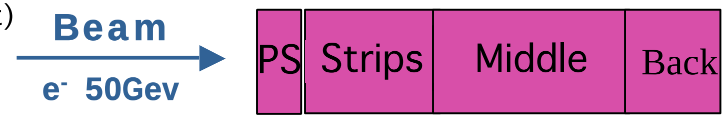


Validation

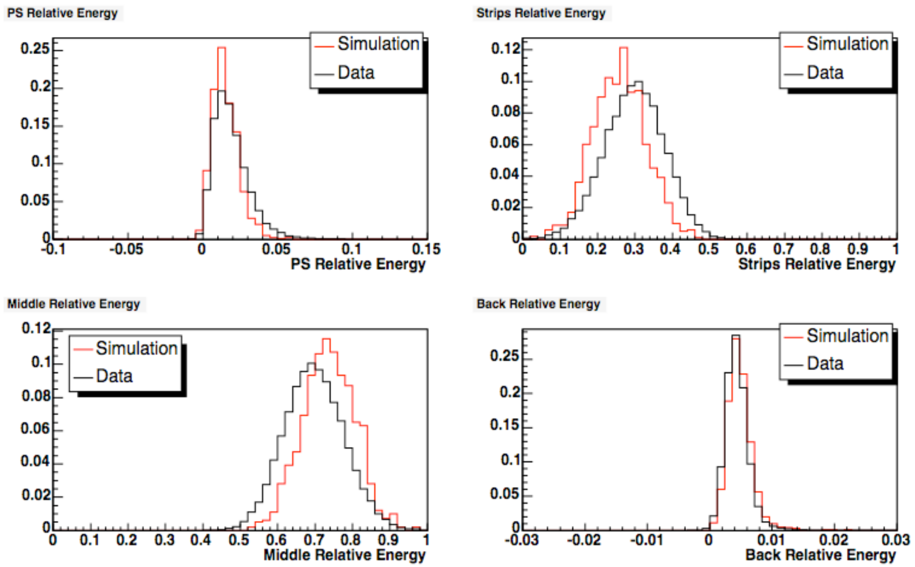
From the CTB(2004)

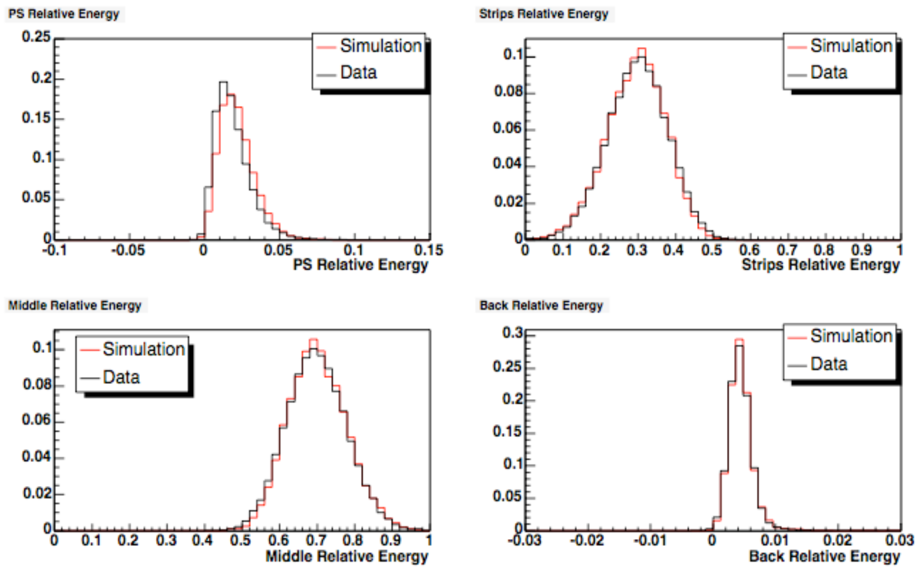
- The MC-Data comparison for **Liquid Argon longitudinal electromagnetic shower** shows under-estimated material
- The user can add material interactively and see the effect
  - from the Python prompt, inserting re-sizeable box
- Under-estimated material:
  - 13.15 mm of Al in the far upstream (the beam line was not simulated to this extent)
  - 15 mm Al can reproduce the absorption of the Liquid Argon in the FOAM material





Energy Deposit per Layer





[ATLAS preliminary, thanks to W. Lampl, S. Paganis et alt.]

(after addition of the missing material)

Conclusions

- G4Atlas is the Geant4-based simulation framework of the ATLAS experiment. It has been successfully and largely used in several massive GRID productions

✓Performance of the G4 simulation is systematically monitored

✓Memory usage and CPU time per event are measured for different event types

✓Since one year of fast and cumbersome development no significant increase on the memory requirements is observed on the G4 side

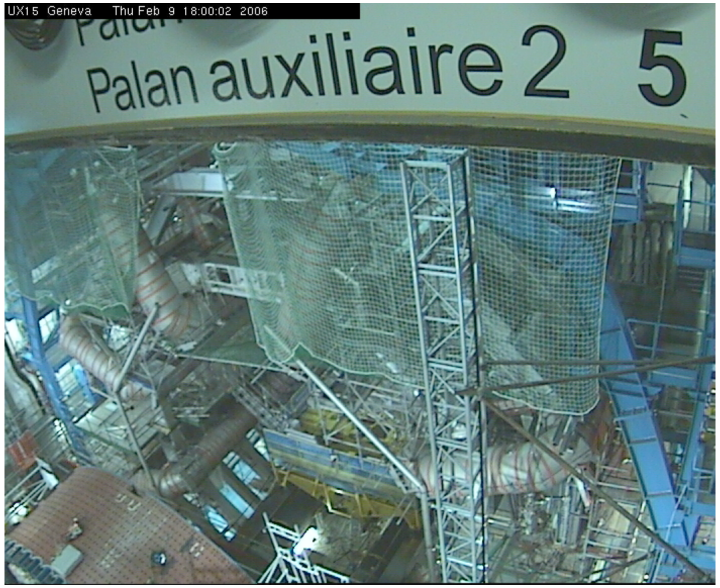
✓All performance results are coming as soon as a new release is distributed

✓Automatic testing in place for a pre-validation

✓Crashing rate 1/10000 maximum

- The detector geometry description is done by GeoModel and automatically translated to G4
  - The detector description is being described according to reality (detector as installed and misalignment)
- PyG4Atlas interface provides the flexibility and configurability required for the full ATLAS and test beam setups (maintenance and usability were achieved)
- PyG4Atlas uses the LCG dictionaries through PyLCGDict Python binding to connect the Python and C^{++} layers
- The data from the Combined Test Beam is a good source for the study of detector performance and Geant4 physics validation

Perspectives



- the long-term success will be measured by the user-friendliness and performance of the ATLAS simulation
  - In large-scale coordinated data challenges
  - On the desktop
- There is a lot more that will go into this program over the course of its very long life
- A user community of 2000 people, will soon make this one of the most scrutinized computer programs in HEP, checked with real data in every corner of phase space
- Optimization and a lot more..

On behalf of the ATLAS Simulation Core Team, thanks.

M. V. Gallas, A. Dell'Acqua, A. Di Simone, J.Boudreau, V. Tsulaia, D. Constanzo