INSTITUT DE PHYSIQUE NUCLÉAIRE
ORSAY

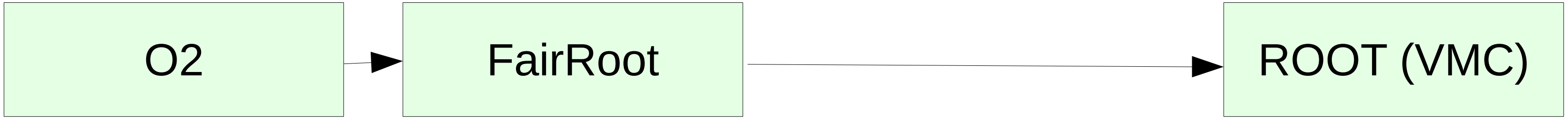
FairRoot/AliceO2 migration to Geant4 multithreading

I. Hrivnacova, IPN Orsay (CNRS/IN2P3)

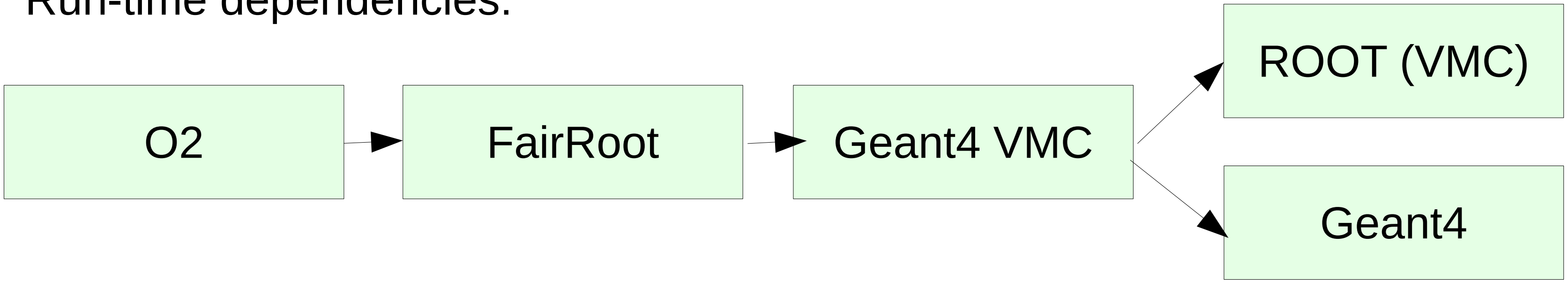
ALICE Offline Week,
29 -31 March 2017, CERN

O2 Simulation

Build dependencies:



Run-time dependencies:

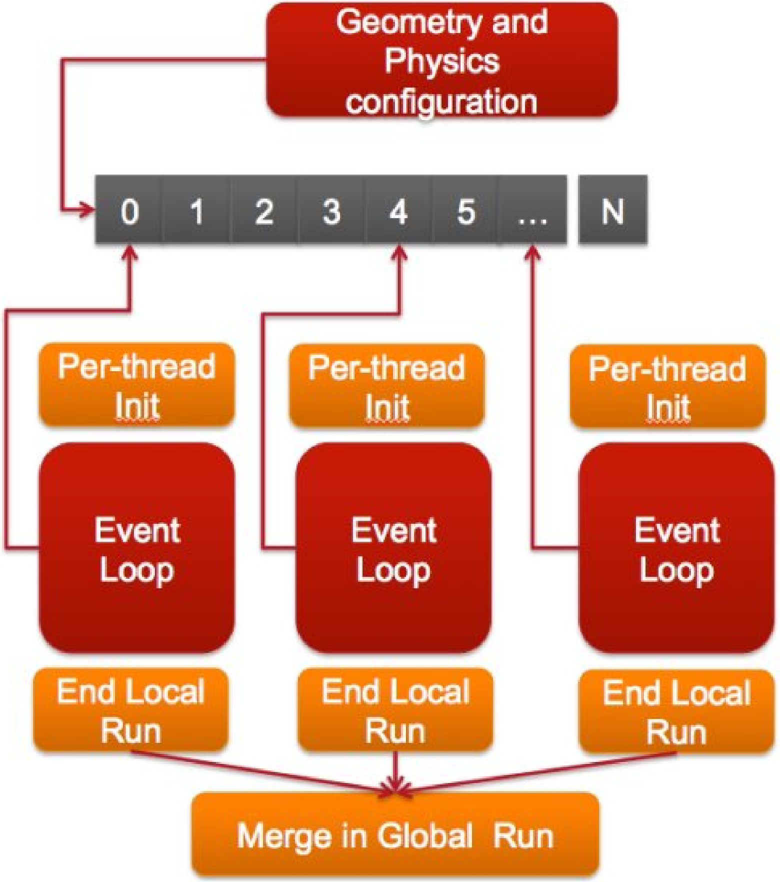


Outline

- Geant4 MT
- Geant4 VMC MT
- FairRoot
- O2

Geant4

- MT since Version 10.0, released on December 6th, 2013
- Event level parallelism
- Heterogeneous parallelism:
  - MPI (Message Passing Interface) works together with MT
  - Intel Thread Building Block (TBB): task based parallelism framework
- Geant4 Web site:
  - http://geant4.web.cern.ch/geant4/



Geant4 VMC

- MT since version 3.0 released on November 11th 2014
  - Beta version 3.0.b01 presented in ALICE Offline week in March 2014
  - Version 3.0 presented at CHEP 2015, in Okinawa
- Geant4 VMC code was adapted for multithreading using the same approach as in Geant4 MT
  - Replacement of the singleton objects in Geant4 VMC with singletons per thread, including the main classes: TVirtualMC and TVirtualMCApplication
  - New VMC package MTRoot – implements the ROOT output per thread with locking critical ROOT operations
  - G4Root (TGeo Geant4 navigation packages) migration – by A. Gheata

Migration of VMC Applications

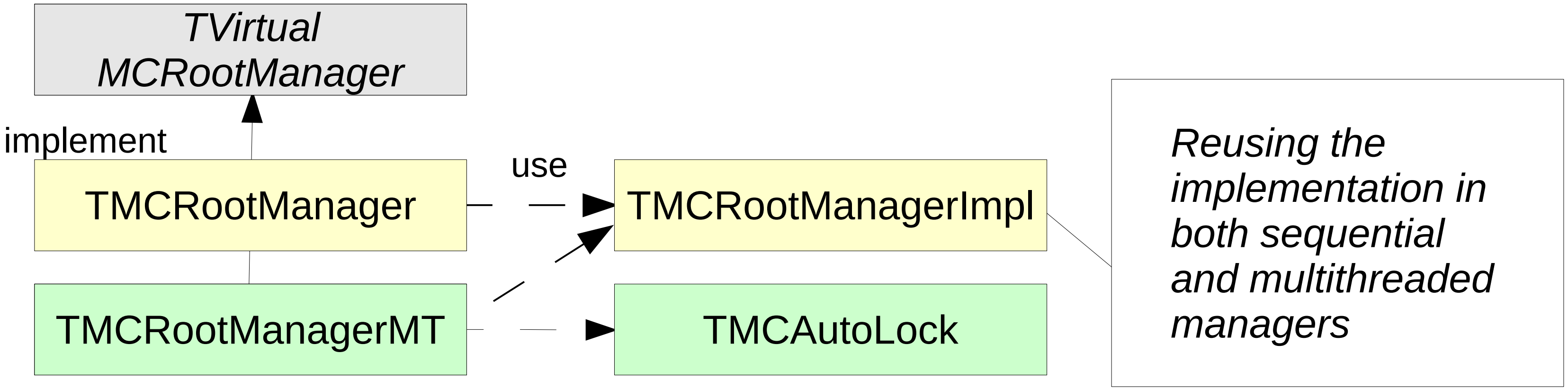
- Users need to implement new functions of TVirtualMCApplication which are then used to clone the application and its containing objects on workers
- Creating of the objects on worker threads is then triggered from the Geant4 VMC classes
  - Examples and more detailed instructions are available from the VMC Web site

class TVirtualMCApplication

```
// required for running in MT
virtual TVirtualMCApplication* CloneForWorker() const;
// optional
virtual void InitForWorker() const;
virtual void BeginWorkerRun() const;
virtual void FinishWorkerRun() const;
virtual void Merge(TVirtualMCApplication* localMCApplication);
```

MTRoot

- A new set of classes which take care of locking critical operations (registering ROOT objects to trees etc.) in multithreading mode is introduced in new **mtroot** package
  - http://root.cern.ch/drupal/content/mtroot
- In previous Geant4 VMC versions, a single class Ex02RootManager for Root IO management was provided in examples



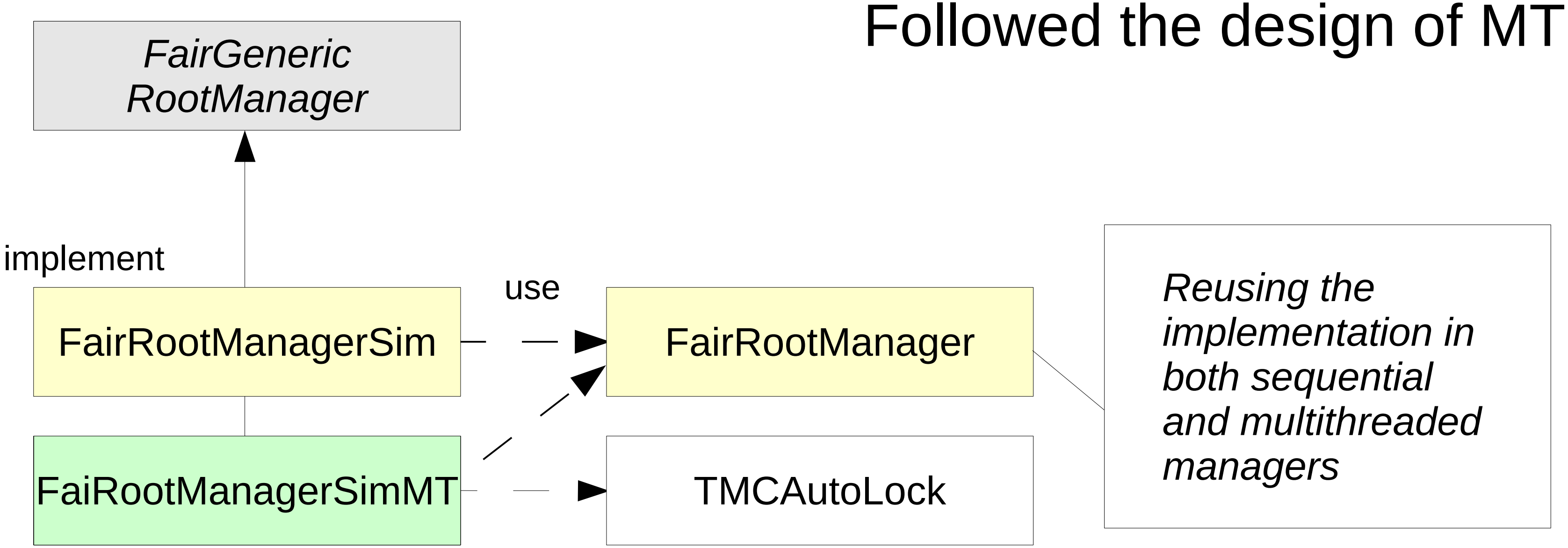
Reusing the implementation in both sequential and multithreaded managers

FairRoot

- Migration of FairMCApplication and related simulation classes
  - December 2014: implementing cloning of the MC application and contained data (FairRun, Detectors, Stack)
  - Except for FairRootManager and FairLogger
- Migration of FairLogger - March 2016
- Migration of the remaining FairRootManager – this month
  - Followed the design of MTRoot but using the API for Register() methods from FairRoot

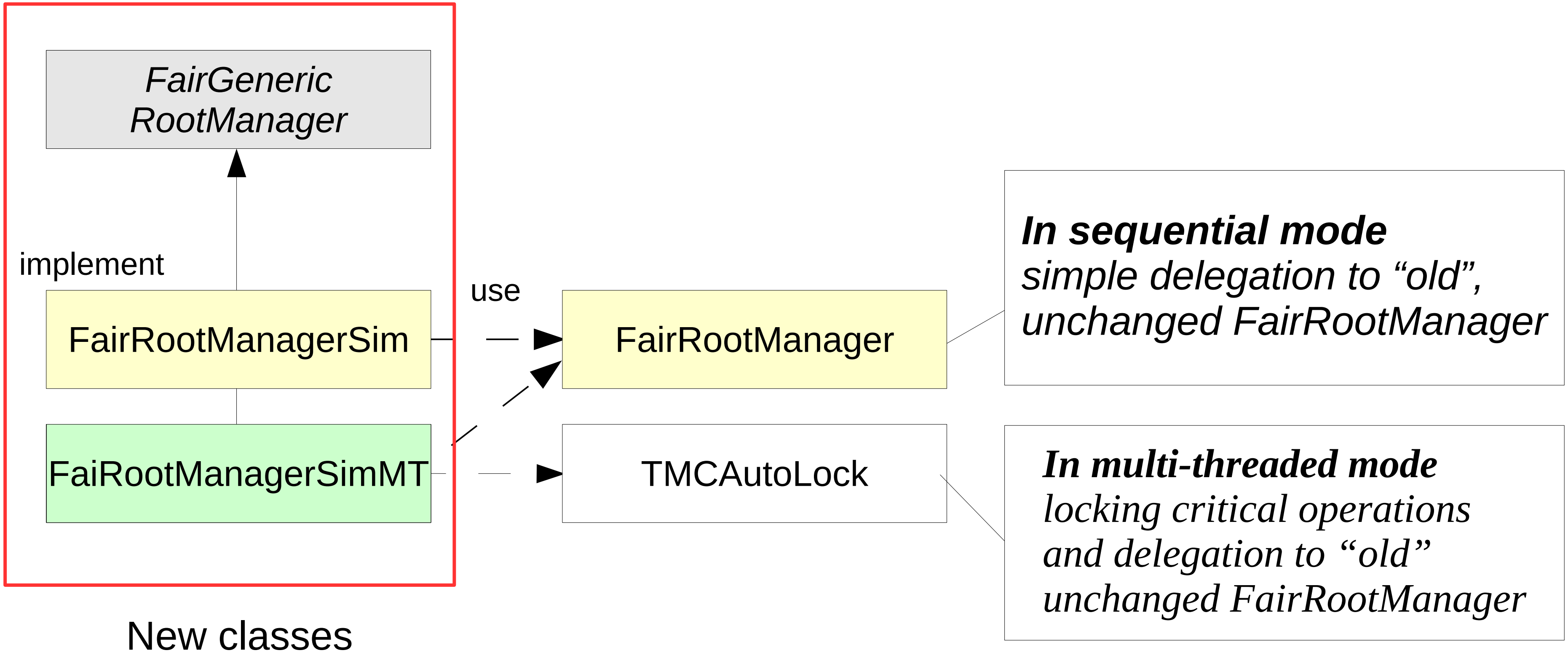
FairRoot – RootManager Classes

Followed the design of MTRoot



FairGeneric RootManager
implement
FairRootManagerSim
use
FairRootManager
FaiRootManagerSimMT
TMCAutoLock
Reusing the implementation in both sequential and multithreaded managers

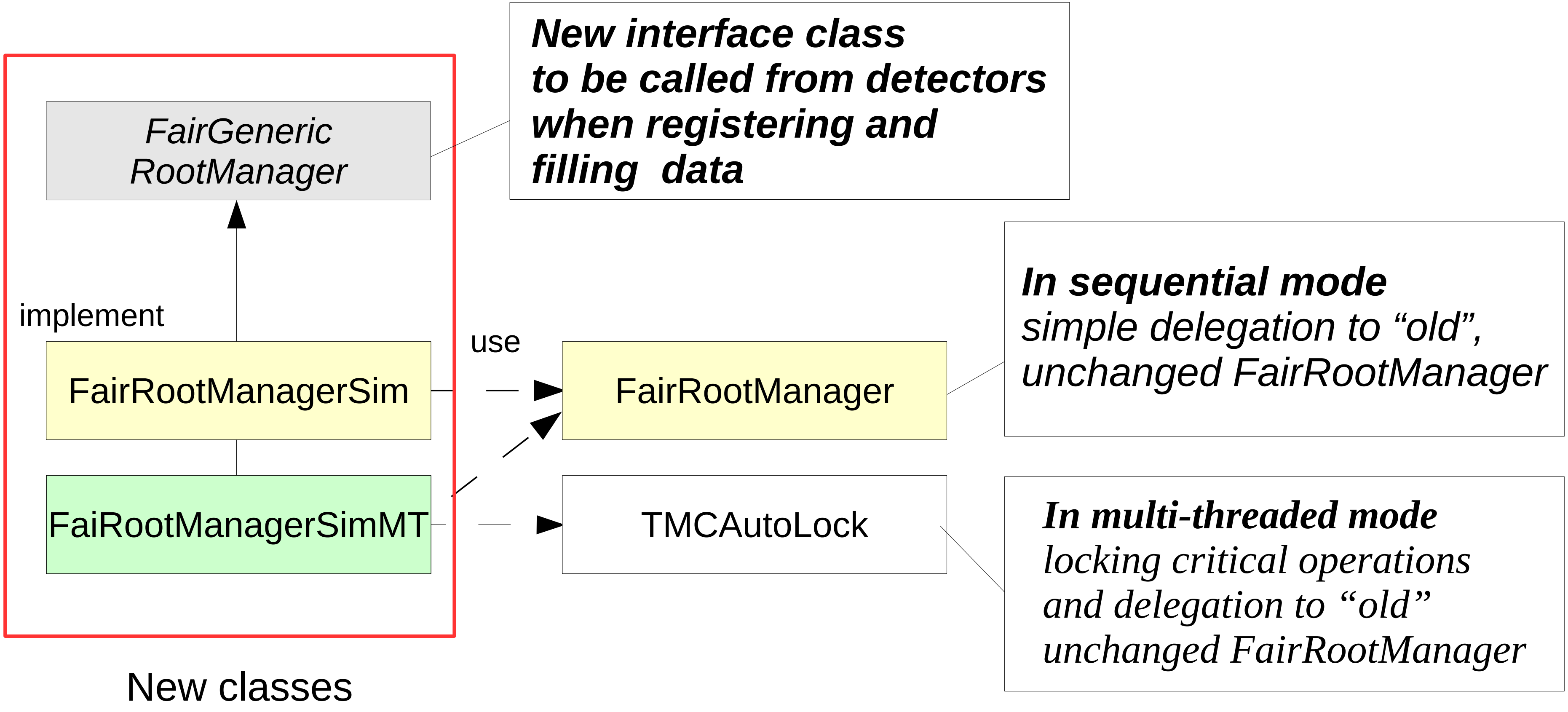
FairRoot – RootManager Classes



In sequential mode
simple delegation to "old",
unchanged FairRootManager

In multi-threaded mode
locking critical operations
and delegation to "old"
unchanged FairRootManager

FairRoot – RootManager Classes

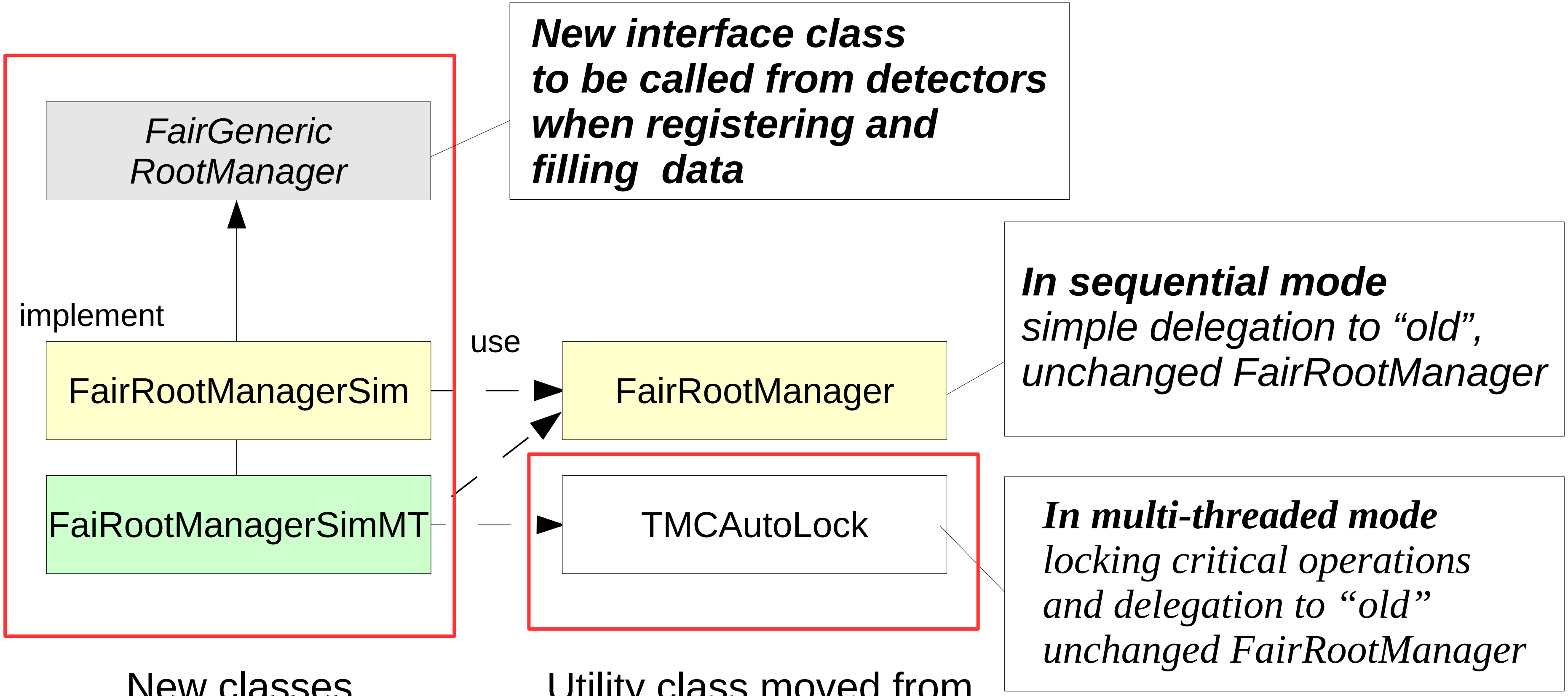


New interface class to be called from detectors when registering and filling data

In sequential mode
simple delegation to "old", unchanged FairRootManager

In multi-threaded mode
locking critical operations and delegation to "old" unchanged FairRootManager

FairRoot – RootManager Classes



New interface class to be called from detectors when registering and filling data

In sequential mode
simple delegation to “old”, unchanged FairRootManager

In multi-threaded mode
locking critical operations and delegation to “old” unchanged FairRootManager

New classes

Utility class moved from geant4_vmc/mtroot in root/vmc

Fixing Problems

- In FairRoot, non-const functions of FairMCApplication need to be called on workers, while the interface defines only const functions
  - No problem for thread safety as they are called behind locks
- New non-const functions were added in TVirtualMCApplication (in development version of ROOT)
  - To keep backward compatibility, they are introduced with new names

class TVirtualMCApplication

```
// NEW non-const functions
virtual void InitOnWorker();
virtual void BeginRunOnWorker();
virtual void FinisRunOnWorker();
```

- **Development versions of ROOT and Geant4 VMC are needed for running simulation in MT mode**
  - But **not required for FairRoot build**

O2

- The only migration item required is replacing FairRootManager::Instance() with FairGenericRootManager::Instance()
  - In the code called by VMC (detectors, stack)
- More fixes may be needed in the implementation of Detector::Clone() method:
  - This method is used to clone the detector objects from master on workers
  - All data (hits) collections must be created in the cloned objects, as they are filled in parallel during event processing
  - Done in ITS::Detector

O2 Test

- O2/macros/run_its_sim.C
  - The only O2 macro, running with Geant4 without errors (mft: requires AliRoot installation, tpc: wrong media)
- Activation of multi-threading mode in g4Config.C:

```
Bool_t mtMode = true;
TG4RunConfiguration* runConfiguration
  = new TG4RunConfiguration("geomRoot", "QGSP_FTFP_BERT",
                            "stepLimiter+specialCuts", false, mtMode);
```

- Run 10 events/2 threads successfully
  - Still work in progress
  - More exercising will be needed to spot possible thread-safety problems

Building O2

- AliBuild O2 defaults:
  - ROOT 6.08/02
  - Geant4 10.01.p03
  - Geant4 VMC v3.2.p1

- O2 MT simulations require:
  - The next ROOT tag (6.09/04?)
  - Geant4 10.03.p01
  - The next Geant4 VMC tag (v3.5)

- Can we update the O2 aliBuild configuration defaults to these versions (when available) ?

AliBuild Problems

- Several attempts *without success* to use aliBuild to build O2 with the whole chain of packages important for simulation as development packages:
  - ROOT, FairRoot, Geant4, Geant4 VMC, VGM
- JIRA ticket opened 9 March
  - https://alice.its.cern.ch/jira/browse/OCCF-30
  - Work in progress
- Own customized build with disabling the packages above and using CMake to find them on the system paths
  - Not ideal, but working solution

Geant4 aliBuild Configuration

- Suggestions for changes in alidist/geant4.sh
  - -DGEANT4_BUILD_MULTITHREADED=ON
    - The user code can be run in sequential mode also against Geant4 libraries built in MT mode; in O2 this option is handled in Detectors/gconfig/g4Config.C macro
  - G4SYSTEM variable is not needed with CMake build
  - G4*DATA environment variables can be set using geant4.sh script available in Geant4 installation
    - Many data files versions depend on Geant4 version
    - This avoids updating them when changing Geant4 version

The Next Steps

- O2:
  - Add the run_sim_its.C test with Geant4 MT in the standard testing suite
  - Fix the [Geant4] simulation of the other detectors
    - Could we avoid AliRoot dependence in MFT ?
  - When all existing detectors are migrated make the multithreading mode default
- FairRoot
  - A few FairMCApplication data member are not yet cloned on workers:
    - fRunInfo, fFairTaskList, fRadGridMan
- Output buffering per thread like G4cout in Geant4

Conclusions

- Migration of FairRoot to Geant4 MT simulation is completed (except for minor rests)
- O2 run_sim_its.sh can run in MT including producing the particles and hits output
- At present development versions of ROOT and Geant4 VMC are required
- When the tags of these packages are provided:
  - Update the O2 aliBuild configuration defaults to these versions
  - A test with MT Geant4 simulation should then be also added in the standard testing