Follow up on debugging

- ~/.rootrc

```
Root.Stacktrace:    no
Browser.Name:       TRootBrowser
```

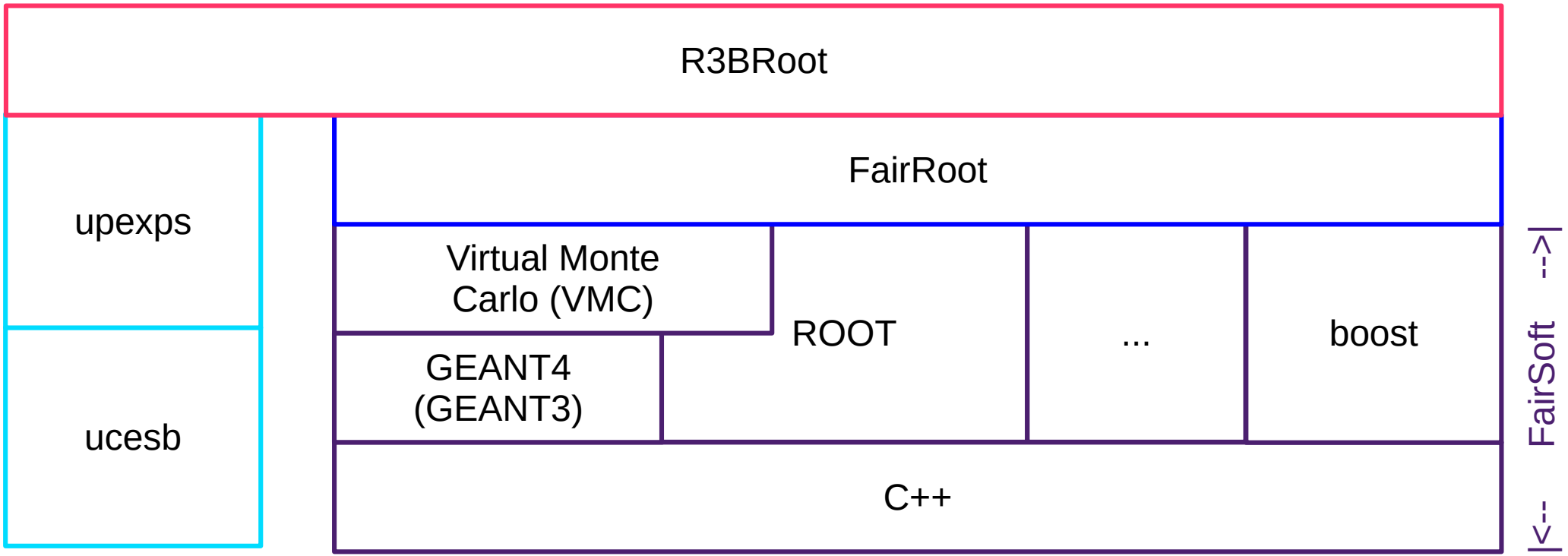
- updated slides, solution branch of yesterday in github now (by the time of presentation)

R3BRoot Overview

Featuring everyones most favorite detector

FairSoft, FairRoot, ucesb, R3BRoot

simplified model



Installing R3BRoot dependencies
overview

- Install FairSoft ($SIMPATH)
- Install FairRoot ($FAIRROOTPATH)
- Install ucesb (${UCESB_DIR}) + upexps
- Compile R3BRoot

Installing FairSoft

https://github.com/FairRootGroup/FairSoft

- FairSoft: bundle of FairRoot dependencies: ROOT, boost, CLHEP, yaml-cpp, VMC, ...
- Two ways to install:
  - legacy way
  - spack way

Latest release:
nov22

- Bring plenty of disk space, time, patience
- Avoid compiling it if you can (->cvmfs version)

Installing FairRoot

https://github.com/FairRootGroup/FairRoot

- export SIMPATH=/path/to/FairSoft
- export PATH=$SIMPATH/bin/:$PATH (for shiny new cmake version)
- git clone ...; cmake ... ; make install
- Takes less time & disk space than FairSoft
- Using precompiled version from /cvmfs/ is still a good alternative

Latest release: v18.8

Installing ucesb, upexps

- ucesb:
  - git clone https://git.chalmers.se/expsubphys/ucesb.git
  - source $SIMPATH/bin/thisroot.sh # for ROOT output
  - make
- upexps
  - not publicly hosted (steal a copy from land, perhaps?)
  - cd 202302_s000 ; make
- not on cvmfs -> /u/land/fake_cvmfs/10/*extra
- more on this in the ucesb talk by Audrey

building R3BRoot

- export SIMPATH=...
  export FAIRROOTPATH=...
- mkdir build; cd build
- cmake /path/to/R3BRoot
- make (make -j for the impatient)

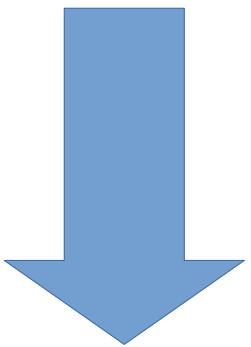
using R3BRoot

- source path/to/build/config.sh
- root -l -q -x myfavoritemacro.C
  (or compile your code like a grown-up, as diskussed yesterday)
-

R3BRoot CMake build system

- R3BRoot
  - CMakeLists.txt
    - add_subdirectory(califa)
  - califa
    - CMakeLists.txt

R3BRoot/CMakeLists.txt
[cmake]
build/Makefile
[make]
build/libs



califa/CMakeLists.txt

```
....
set(INCLUDE_DIRECTORIES
     ${R3BROOT_SOURCE_DIR}/tracking ...)
set(SRCS
     ./ana/R3BCalifaCrystalCalDataAnalysis.cxx ...)

set(LINKDEF CalifaLinkDef.h)

set(LIBRARY_NAME R3BCalifa)

set(DEPENDENCIES
  Spectrum R3BBase R3BPassive R3BData
  R3BTracking Boost::regex)

GENERATE_LIBRARY()

add_subdirectory(test)
```

allows us to include headers from R3BRoot/tracking

List all the source files we want to compile
Feeding rootcint

What libraries does our library require?

The CMake macro GENERATE_LIBRARY will implicitly take the value of LINKDEF, SRCS DEPENDENCIES etc

We also have R3Broot/califa/test/CMakeLists.txt

Making macros work: CalifaLinkDef.h

- GENERATE_LIBRARY will call rootcint with $LINKDEF so it can make the relevant classes available to the root interpreter
- CalifaLinkDef.h

  template support???!

```
#pragma link C++ class R3BCalifaGeometry+;
#pragma link C++ class R3BCalifa+;
#pragma link C++ class R3BCalifaDigitizer+;
```

- If your header is broken, you may get an error from rootcint instead of from your trusted system compiler
- also why we use #ifndef R3BCALIFA_H instead of #pragma once

Component overview -- Califa

omitted lots of stuff here: sim, parameter files, geometry, ...

- R3BRoot/
  - r3bdata/califaData
    - CalifaHitData, CalifaCrystalCalData, CalifaClusterData
  - r3bsource/califa/
    - ext_h101_califa.h
    - CalifaFebexReader # for experiment data, upexps interface
  - califa
    - FairTask: R3BCalifaDigitizer # for simulation data
    - FairTask: CalifaMapped2CrystalCal # for experiment data
    - FairTask: CalifaCrystalCal2Cluster # always

First task

- Install FairSoft and FairRoot from the scratch
- You can pick either legacy or spack FairSoft
- For bonus credits:
  - Install a custom compiler first and use it with that
  - tweak at least five CMAKE_* variables

First task

- Install FairSoft and FairRoot from the scratch
- You can pick either legacy or spack FairSoft
- For bonus credits:
  - Install a custom compiler first and use it with that
  - tweak at least five CMAKE_* variables

Just kidding