Repository discussion

B. von Haller
M. Al-Turany , D. Berzano, G. Eulisse, P. Hristov, M. Richter

CERN

22.04.2016

Introduction

- Today's presentation goals
  - Present our proposal
  - Get feedback before implementing it in a branch

- In most cases, no « right » or « wrong »
  - Question of taste, experience, etc…
  - Consensus

Principles (1)

(order conveys no information)

1. A module is a set of code closely related and sharing an interface that can result in one or more libraries.
2. We favour extracting large common modules to their own repositories.
3. AliceO2 is a thin repo containing
   - Detector specific code
   - Commonalities
   - Global algorithms
4. The AliceO2 repo has a mixture of « per detector » and « per function » sub-modules with corresponding sub-structure

Principles (2)

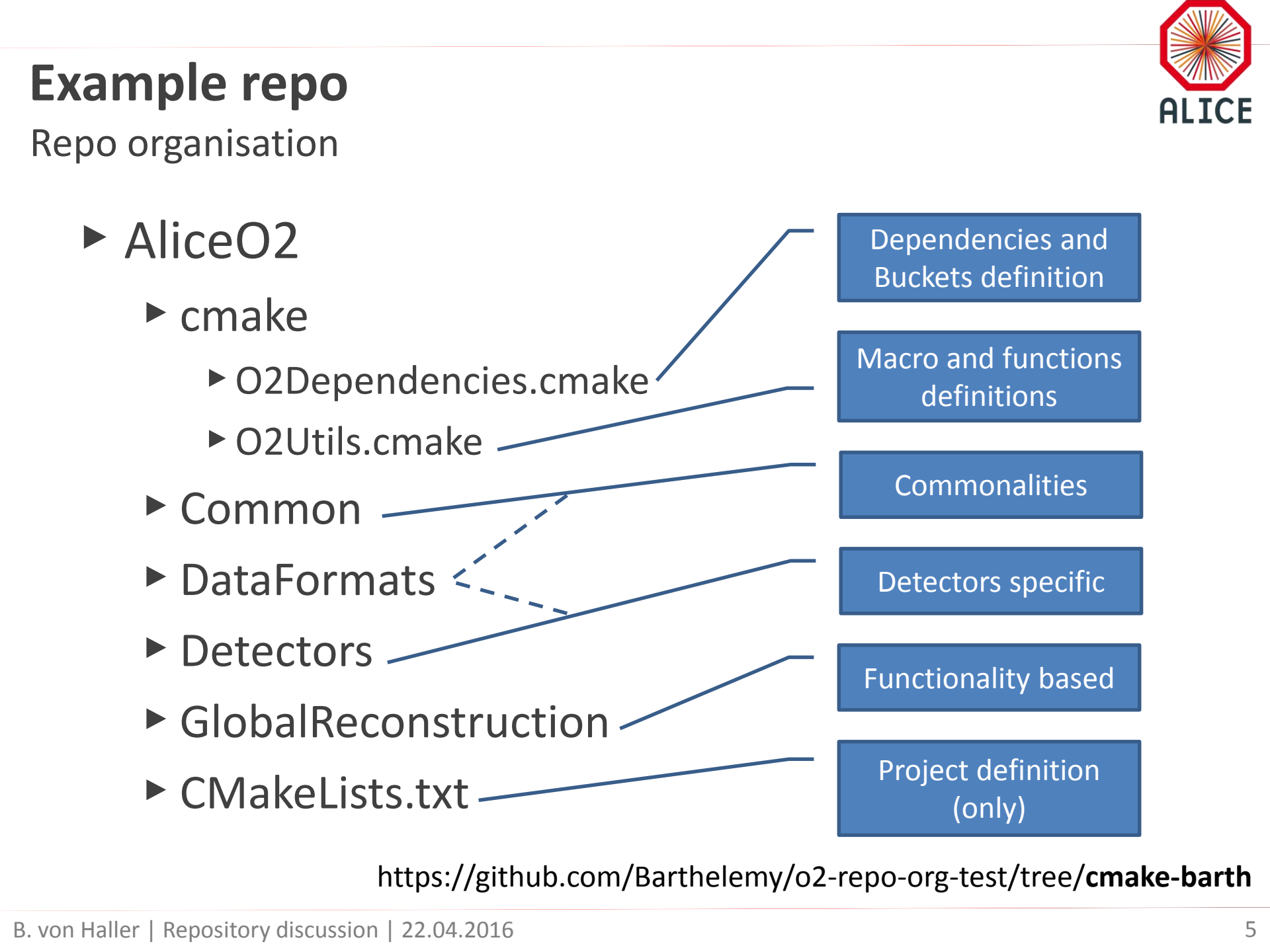
(order conveys no information)

5. Dependencies are defined centrally as buckets .
6. Each sub-module generates a single library linked against the dependencies defined in a single bucket.
7. Sub-modules' executable(s) link against the same bucket as the library and the library itself.
8. Horizontal dependencies are in general forbidden (between sub-modules at the same level)
9. Naming : camel-case
   - What is repeated / structural starts with a lower case letter (e.g. src, include, test)
   - The rest (labels, unique names) start with an upper case letter (e.g. Common, Detectors)

Example repo

Repo organisation

- AliceO2
  - cmake
    - O2Dependencies.cmake
    - O2Utils.cmake
  - Common
  - DataFormats
  - Detectors
  - GlobalReconstruction
  - CMakeLists.txt



https://github.com/Barthelemy/o2-repo-org-test/tree/**cmake-barth**

O2Dependencies.cmake

```
# Packages
find_package(
              Boost
       COMPONENTS
              unit_test_framework program_options
       REQUIRED
)
include_directories( ${Boost_INCLUDE_DIRS} )

# Buckets definition
o2_define_bucket(
       NAME
              GlobalTracking_Bucket
       DEPENDENCIES
              Common
              ${Boost_PROGRAM_OPTIONS_LIBRARY}
)
```

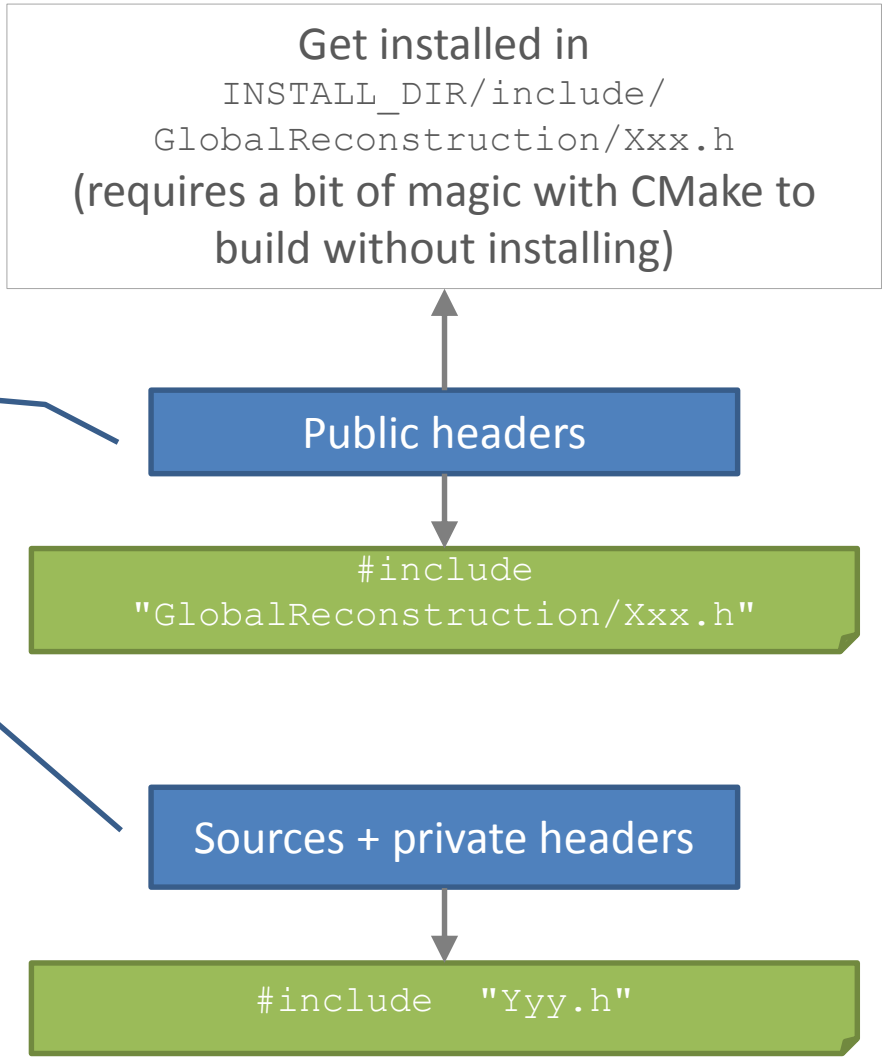
O2Utils.cmake

- macro(o2_define_bucket)
  - Used in O2Dependencies
- macro(o2_target_link_bucket)
  - Used in users' code, same as target_link_library

Example repo

Structure details (functionality based sub-module)

- GlobalReconstruction
  - cmake
  - doc
  - include
    - Something.h
  - src
    - Something.cxx
    - PrivateHeader.h
  - test
  - CMakeLists.txt



Example repo

Magic for the headers

- In AliceO2/CMakeLists.txt

```
# function to pre-install all headers in the build directory
pre_install_o2_includes()
# make it available to the sub-modules
include_directories(${CMAKE_BINARY_DIR}/include)
```

- In AliceO2/GlobalReconstruction/src/Something.cxx

```
#include "Common/base.h"
#include "PrivateHeader.h" // in same module
```

Example repo

CMakeLists.txt of a sub-module

- # Define the library

```
add_library(${LIBRARY_NAME} SHARED ${LIB_SOURCES})
```

- # Link against a bucket

```
o2_target_link_bucket(TARGET ${LIBRARY_NAME}
                      BUCKET "GlobalTracking_Bucket")
```

- # Define application

```
add_executable(${APPLICATION_NAME} ${APP_SOURCES})
```

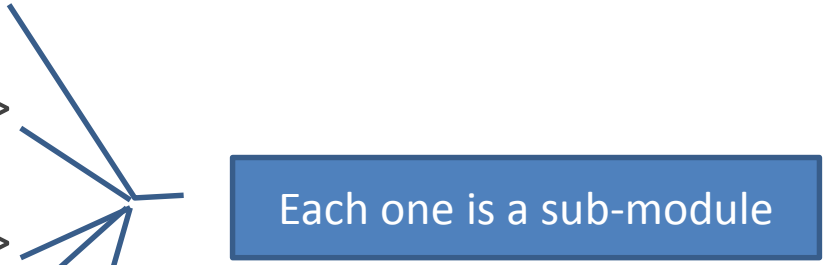
- # Links the application against the bucket

```
o2_target_link_bucket(TARGET ${APPLICATION_NAME}
                      BUCKET "GlobalTracking_Bucket")
target_link_libraries(${APPLICATION_NAME} ${LIBRARY_NAME})
```

Example repo

Structure details (detectors based sub-modules)

- Detectors
  - ITS
    - reconstruction
      - <same as previous slide>
    - simulation
      - <same as previous slide>
    - base
      - <same as previous slide>
  - TPC
    - reconstruction
      - More sub-folders…
    - simulation
      - <same as previous slide>



Extra notes

- Extracted modules in AliceO2 group follow the same principles and in particular
  - Coding guidelines
  - General structure
  - 1 lib per sub-module
- Need scripts to create new modules and submodules
- Global definitions in O2Utils.cmake

Next steps

- Reorganize AliceO2
  - Add directories such as Tutorial, Topologies, …
  - Regroup and split
  - Rename
- Apply dependencies principles to AliceO2
  - Modify CMakeLists.txt

- Add examples for documentation and tests
- Boiler code -> remove/extract
- Write scripts / template to generate new modules

Feedback

- Time for questions and feedback