

# Application of TriBITS to Larger Componentized Scientific and Engineering Software Projects

Sandia National Laboratories

Roscoe A. Bartlett

Sandia National Laboratories, New Mexico, 87185



## Challenges

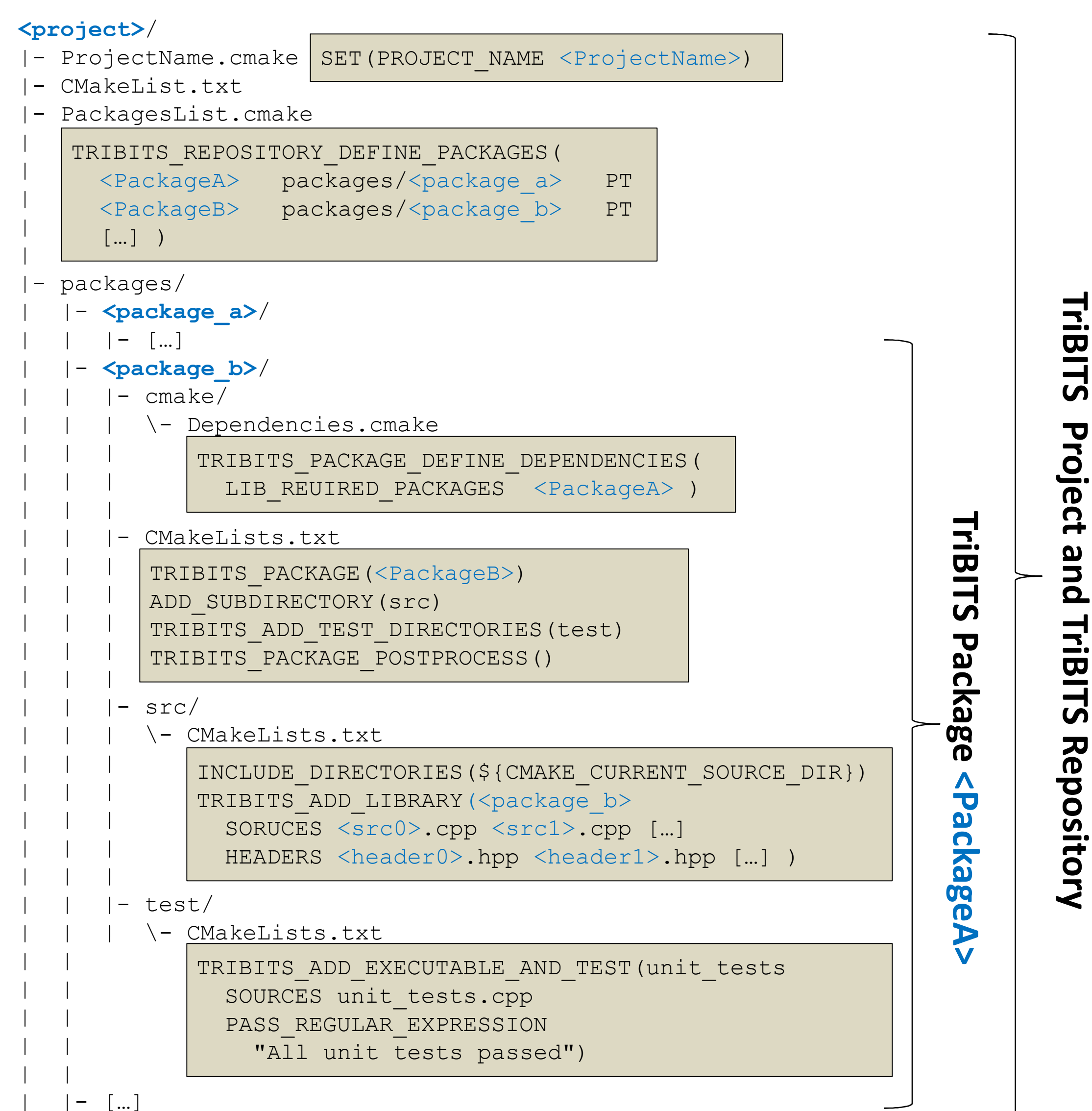
- Multiple software repositories and distributed development teams
- Multiple programming languages (C, C++, Fortran)
- Multiple development and deployment platforms
- Development, builds, testing, and/or deployments of different subsets of the software
- Co-development of multiple software packages

## Approach

### Tribal Build, Integration, and Test System (TriBITS) using CMake

- Framework for large, distributed multi-repository CMake projects
- Flexible package-based dependency and namespacing architecture
- Automatic package dependency handling using dependency DAG
- Reduced boiler-plate CMake code and enforce consistency across large distributed projects

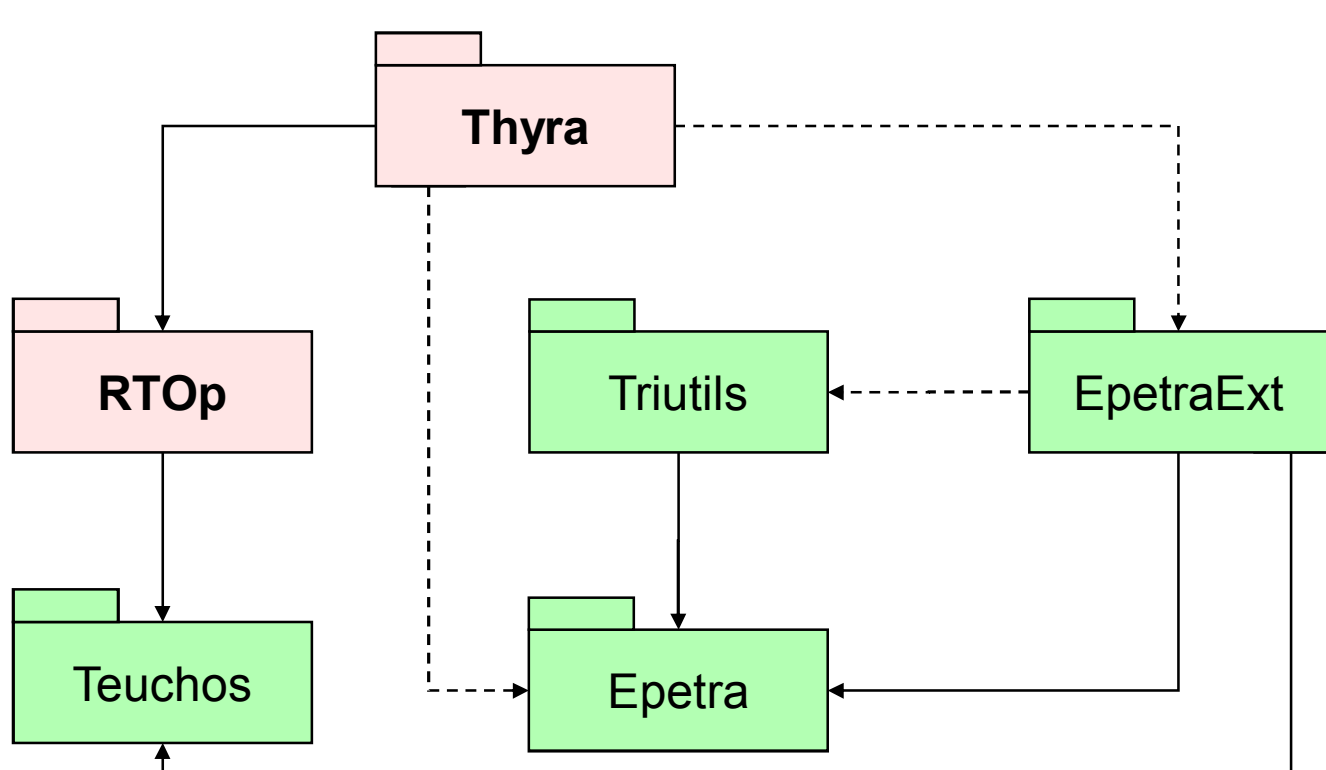
## Package-based Architecture



## Package Dependencies

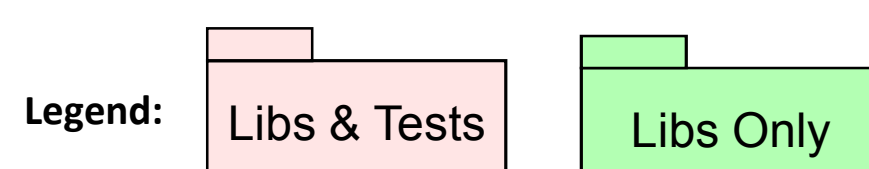
```

$ cmake \
-DTrilinos_ENABLE_RTOP=ON \
-DTrilinos_ENABLE_ALL_FORWARD_DEP_PACKAGES=ON \
-DTrilinos_ENABLE_TESTS=BOOL=ON \
../Trilinos
  
```



### Example: CI Testing

- Tests built and run for affected downstream packages only.
- Only libraries are enabled for upstream packages.
- Huge cost savings for most intermediate packages in dependency graph!**



## Raw CMake vs. TriBITS

### Raw CMakeLists.txt

```

# Build and install library
SET(HEADERS hello_world.lib.hpp)
SET(SOURCES hello_world.lib.cpp)
ADD_LIBRARY(hello_world_lib ${SOURCES})
INSTALL(TARGETS hello_world_lib
  DESTINATION lib)
INSTALL(FILES ${HEADERS} DESTINATION include)

# Build and install user executable
ADD_EXECUTABLE(hello_world
  hello_world_main.cpp)
TARGET_LINK_LIBRARIES(hello_world
  hello_world_lib)
INSTALL(TARGETS hello_world DESTINATION bin)

# Test the executable
ADD_TEST(test
  ${CMAKE_CURRENT_BINARY_DIR}/hello_world)
SET_TESTS_PROPERTIES(test PROPERTIES
  PASS_REGULAR_EXPRESSION "Hello World")

# Build and run some unit tests
ADD_EXECUTABLE(unit_tests
  hello_world_unit_tests.cpp)
TARGET_LINK_LIBRARIES(unit_tests
  hello_world_lib)
ADD_TEST(unit_test
  ${CMAKE_CURRENT_BINARY_DIR}/unit_tests)
SET_TESTS_PROPERTIES(unit_test PROPERTIES
  PASS_REGULAR_EXPRESSION
    "All unit tests passed")
  
```

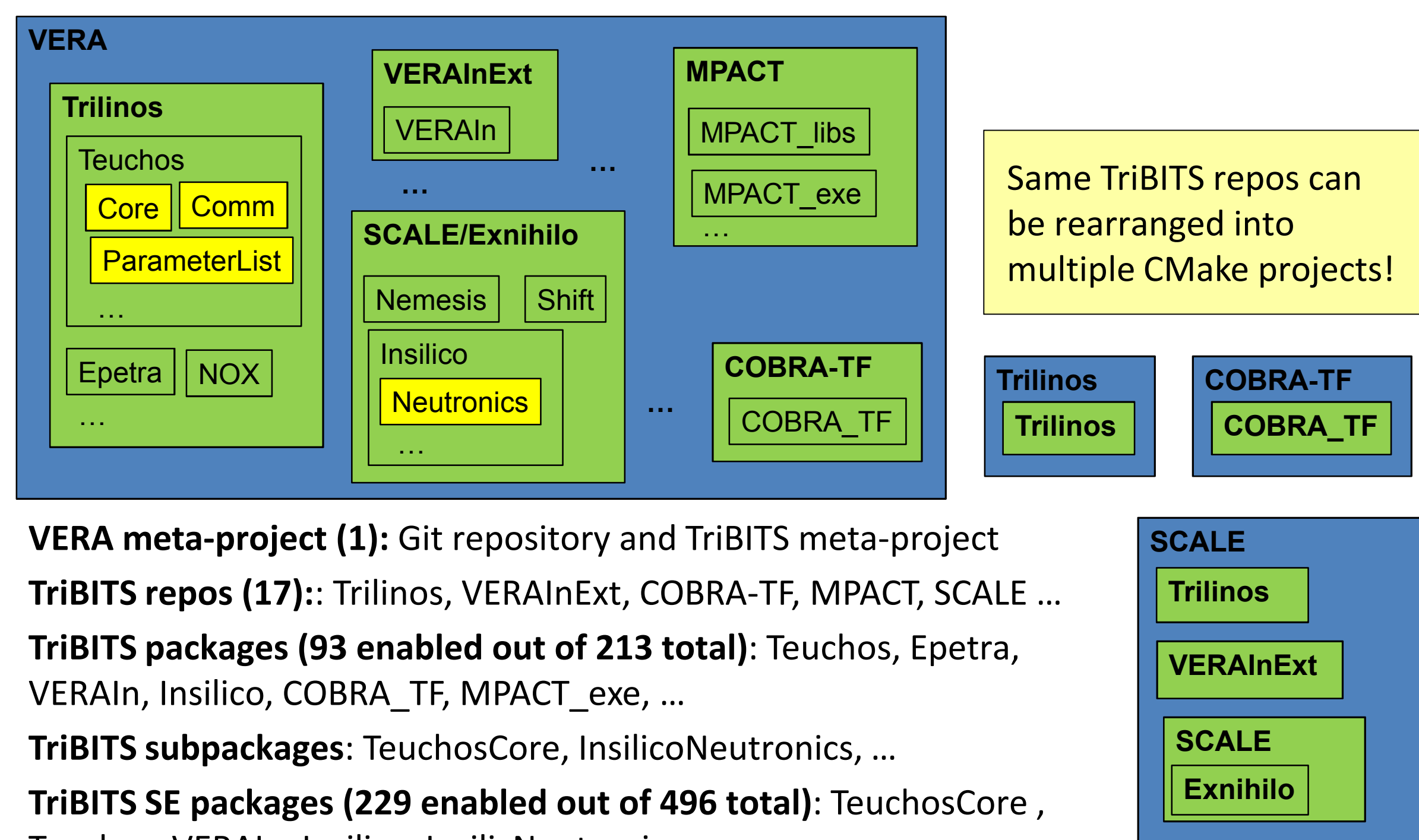
### TriBITS CMakeLists.txt

```

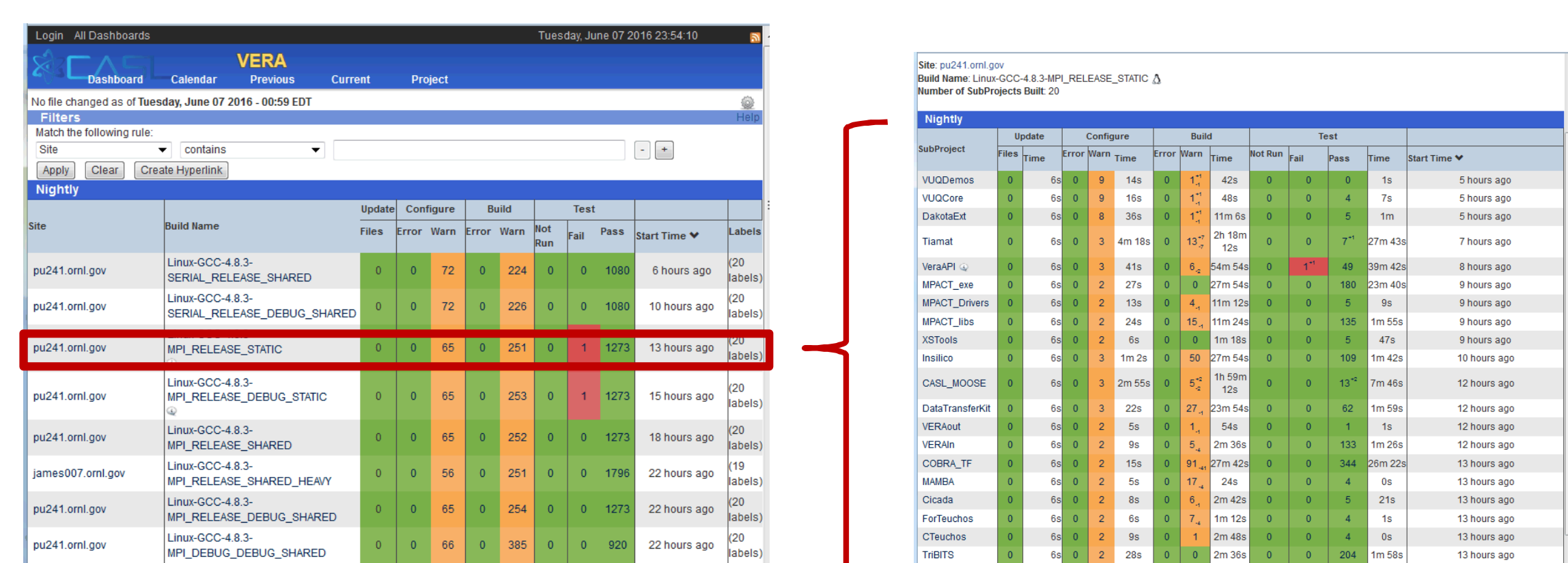
TRIBITS_ADD_LIBRARY(hello_world_lib
  HEADERS hello_world.lib.hpp
  SOURCES hello_world.lib.cpp)
TRIBITS_ADD_EXECUTABLE(hello_world NOEXEPREFIX
  SOURCES hello_world_main.cpp INSTALLABLE)
TRIBITS_ADD_TEST(hello_world NOEXEPREFIX
  PASS_REGULAR_EXPRESSION "Hello World")
TRIBITS_ADD_EXECUTABLE_AND_TEST(unit_tests
  SOURCES hello_world_unit_tests.cpp
  PASS_REGULAR_EXPRESSION
    "All unit tests passed")
  
```

- Less duplication and boiler-plate
- Fewer commands
- Fewer lines of CMake code
- Installs by default (most common)
- Library linking automatically handled
- Automatic namespacing of test & test executable names. (**All target and test names must be globally unique!**)
- Enforced uniform behavior across all CMakeListst.txt files

## Example: Trilinos & CASL



## Package-based CDash board



## Usage and Availability

**Used By:** Trilinos (SNL), Drekar (SNL), MPACT (U.Mich./ORNL), COBRA-TF (PennState/ORNL), SCALE/Exnihilo (ORNL), DataTransferKit (ORNL), ...

**Availability:** GitHub, 3-clause BSD license

**See:** <https://tribits.org>, <https://github.com/tribitspub/tribits>

We acknowledge the contributions of Brent Pershacker (SNL), Nico Schlomer (T.U. Berlin) and others who have helped to develop and extend TriBITS.