Technische
Hochschule
Augsburg

Building Soar with CMake and Conan

Moritz Schmidt, 45th Soar Workshop

Outline

- ↗ Why a new build system?
- ↗ CMake & Conan vs Scons
- ↗ CMake Build System with Conan
- ↗ What’s missing?

Why a new Build System?

- Easy setup for `soar_ros`
- Dependency management
- Reduced set of Soar + dependencies required (no Java, tcl, SWIG, scons)
- Hard to understand Scons (documentation)
- CMake used by ROS 2 (known tooling)

CMake & Conan vs. Scons

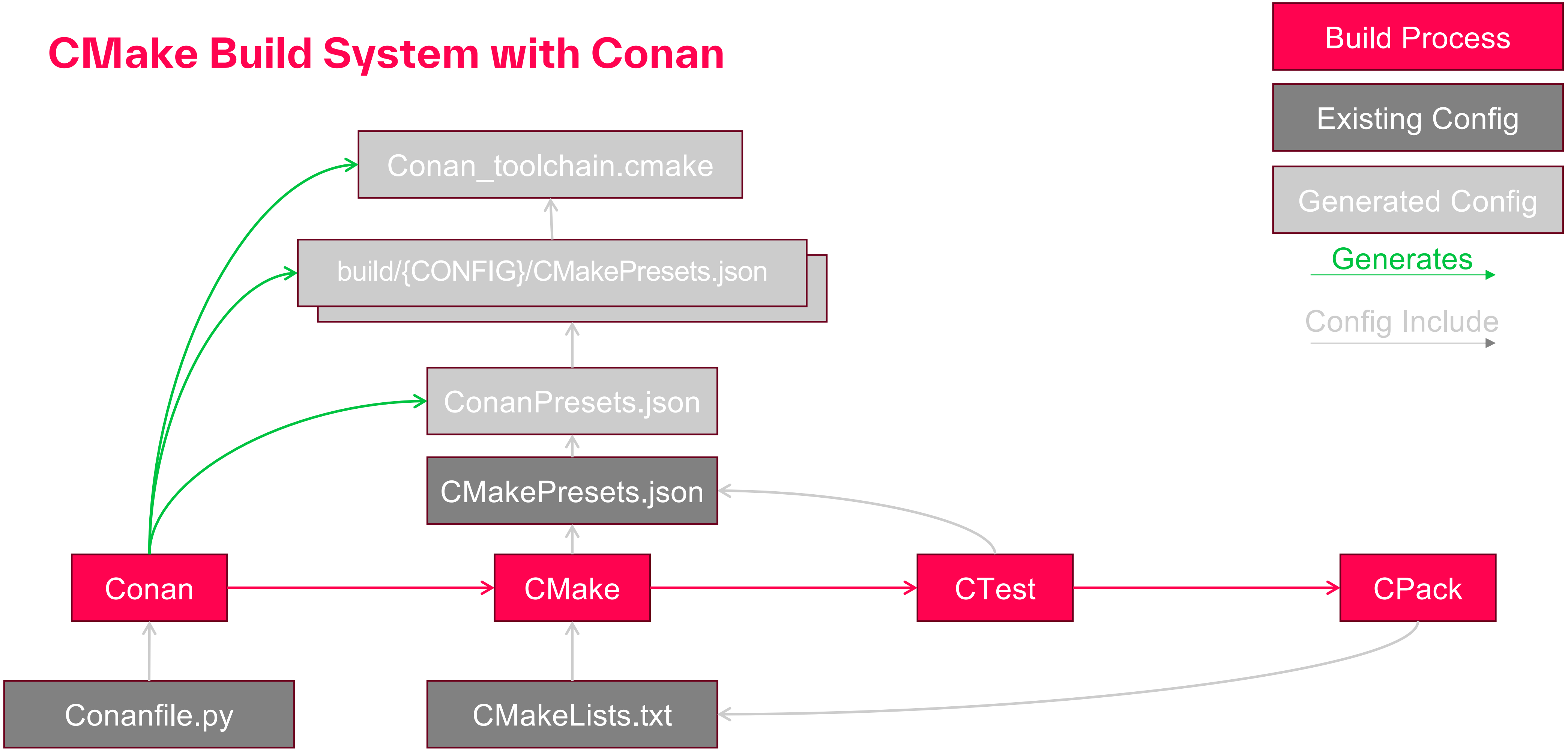
Scons

- Python-based
- Uses environment/ global variables may lead to frustration
- Hard to understand due to scope of scripts
- Lack of documentation/ usage/ examples

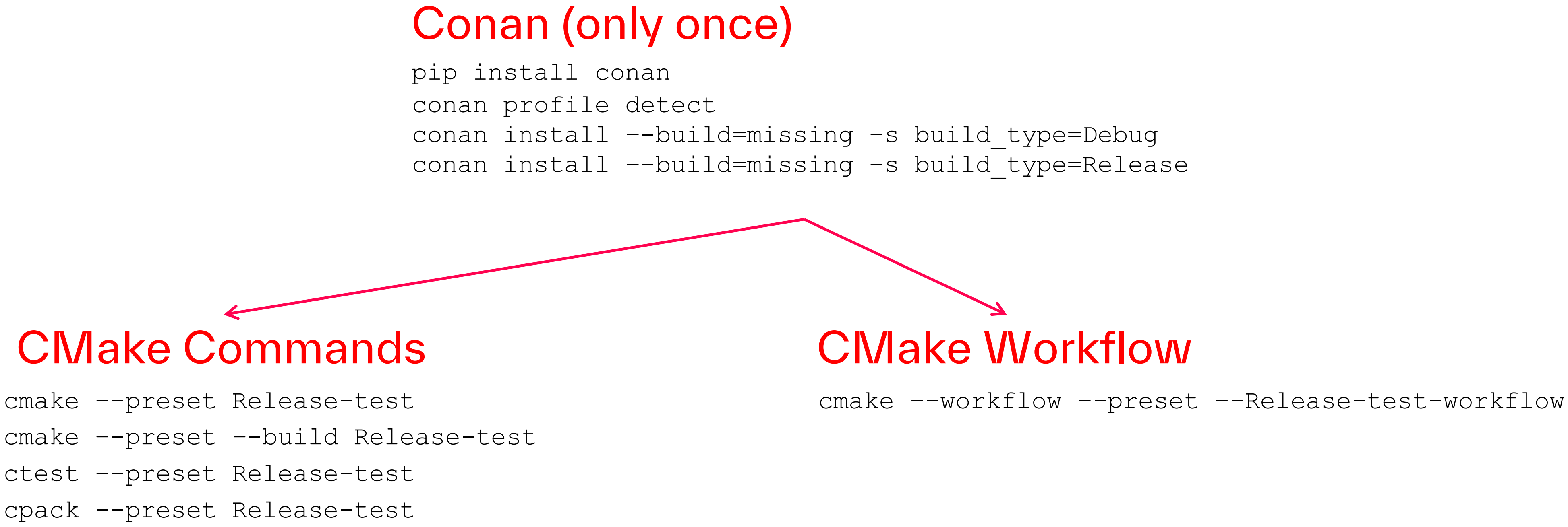
CMake + Conan

- CMake language
- Wide adoption
- Documentation, Tutorials, Forum, ...
- Integrations (e.g. Conan)
- Packaging, Testing, IDE Integrations
- Dependency management
- De-facto CPP standard
- Use of Ninja in the background
- (Faster build times)

CMake Build System with Conan



CMake Build System with Conan



Conan (only once)

```
pip install conan
conan profile detect
conan install --build=missing -s build_type=Debug
conan install --build=missing -s build_type=Release
```

CMake Commands

```
cmake --preset Release-test
cmake --preset --build Release-test
ctest --preset Release-test
cpack --preset Release-test
```

CMake Workflow

```
cmake --workflow --preset --Release-test-workflow
```

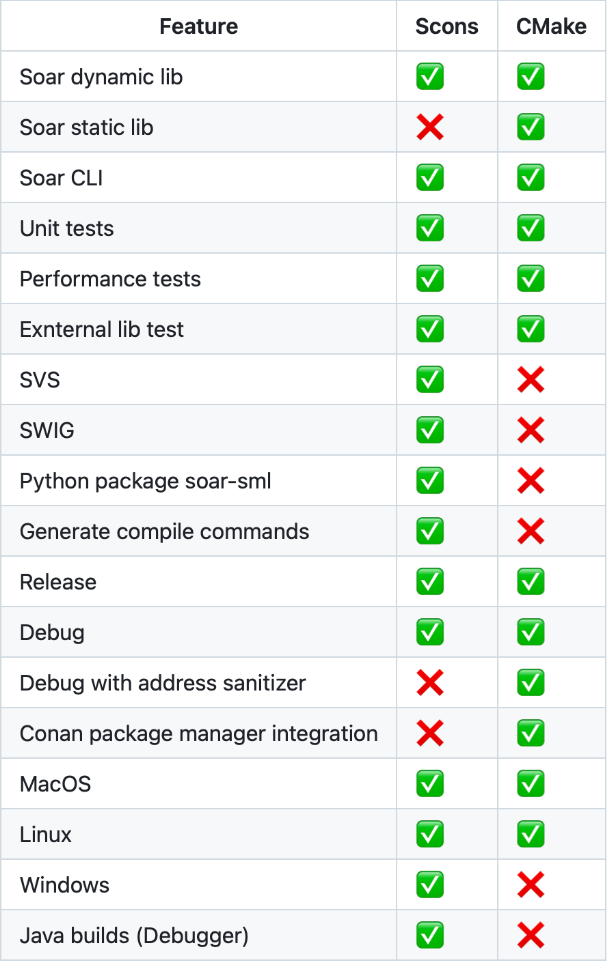
What’s the benefit?

```
include(FetchContent)
FetchContent_Declare(
    soar
    GIT_REPOSITORY https://github.com/soargroup/soar
    GIT_TAG development
)
FetchContent_MakeAvailable(soar)

target_link_libraries(<your package> soar_lib)
```

* This does not install dependencies, e.g. SQLite3. → Publish to conan

What’s missing?



Feature	Scons	CMake
Soar dynamic lib	✅	✅
Soar static lib	❌	✅
Soar CLI	✅	✅
Unit tests	✅	✅
Performance tests	✅	✅
Exnternal lib test	✅	✅
SVS	✅	❌
SWIG	✅	❌
Python package soar-sml	✅	❌
Generate compile commands	✅	❌
Release	✅	✅
Debug	✅	✅
Debug with address sanitizer	❌	✅
Conan package manager integration	❌	✅
MacOS	✅	✅
Linux	✅	✅
Windows	✅	❌
Java builds (Debugger)	✅	❌

Comparison as of **f9219a2** (2025-05-03), up-to-date version see https://github.com/soargroup/soar

Technische
Hochschule
Augsburg

Moritz Schmidt

Faculty of Electrical Engineering
Research Associate

Technical University of Applied Sciences Augsburg
An der Hochschule 1
D-86161 Augsburg
T +49 821 5586 1010
moritz.schmidt@tha.de
www.tha.de