

# CMake Clarifié : Maîtrisez vos Projets C/C++



Luc Badinga  
Ingénieur Systèmes embarqués

# Sommaire

## Introduction à CMake

### 1. Installation

### 2. Cmake, Test et Outils

- Module 1 - Les Fondamentaux CMake
- Module 2 - Gestion de Projets simples et complexes avec CMake
- Module 3 - Tests unitaires et et option d'optimisation
- Module 4 - CMake Avancé pour Projets C/C++
- Module 5 - Outils

## Conclusion et Ressources

# Introduction à CMake

- Définition CMake



- Pourquoi utiliser CMake et exemples des projets

# 1. Installation sur Windows

Les outils à installer sur Windows:

- Visual Studio

<https://code.visualstudio.com/Download>

*Attention : Bien choisir le kit de développement*

<https://learn.microsoft.com/en-us/cpp/build/vscpp-step-0-installation?view=msvc-170#step-4---choose-workloads>

- Cmake

<https://cmake.org/download/>

- Git ou GitDesktop

<https://git-scm.com/downloads>

<https://desktop.github.com/download/>

- Doxygen

<https://www.doxygen.nl/download.html>

- Python : Version 3.8 ou la plus récente

# 1. Installation sur Linux

Les outils à installer sur Linux:

- `sudo apt-get update`
- `sudo apt-get upgrade`
  
- `# Recommender`
- `sudo apt-get install gcc g++ gdb`
- `sudo apt-get install make cmake`
- `sudo apt-get install git`
- `sudo apt-get install doxygen`
- `sudo apt-get install python3 python3-pip`
  
- `# Optionnel`
- `sudo apt-get install lcov gcovr`
- `sudo apt-get install ccache`
- `sudo apt-get install cppcheck`
- `sudo apt-get install llvm clang-format clang-tidy`
- `sudo apt-get install curl zip unzip tar`
- `sudo apt-get install graphviz`

# 1. Installation sur Mac

Les outils à installer sur Mac:

- Xcode

<https://ics.uci.edu/~pattis/common/handouts/macclion/clang.html>

- lldb --version
- clang --version
- clang++ --version

- Brew

- Installation : <https://www.igeeksblog.com/how-to-install-homebrew-on-mac/>
- Guide complet  
<https://www.igeeksblog.com/how-to-install-homebrew-on-mac/>

Installation :

- brew install git
- brew install make
- brew install cmake
- brew install doxygen
- brew install lcov
- brew install gcovr
- brew install ccache

# 1. Setup VScode avec WSL

Les outils à installer sur Windows:

- Installer VS Code

<https://code.visualstudio.com/download>

- Installer WSL

<https://learn.microsoft.com/fr-fr/windows/wsl/install>

- Set up pour WSL dans VScode



# Ressources du cours

- Dépôt Github

<https://github.com/mbadinga1/CoursCMake>

- slide du cours disponible

# Organisation du cours

- VScode avec WSL
- Ressources sur Git
- slide du cours disponible



# Module 1 – Les Fondamentaux CMake

## Mots clés :

CMake, CMakeLists.txt, cmake\_minimum\_required, add\_executable, add\_library, build, make

## Objectifs:

- Etre capable de créer un projet basique avec CMake
- Connaître les commandes basiques pour lancer build et l'exécutable
- Apprentissage par projet – Conversion température

# Module 2 : Gestion de Projets C/C++ – CMake

## Mots clés :

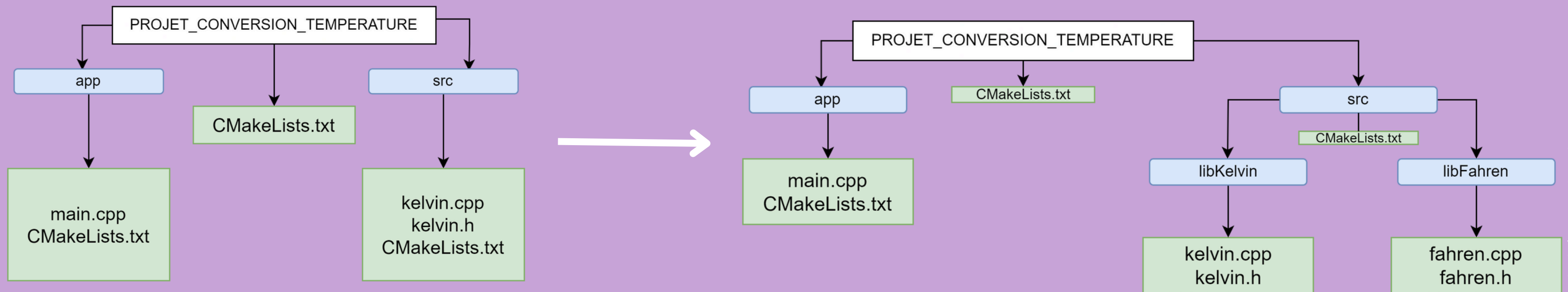
target\_link\_libraries, add\_subdirectory, find\_package, \$, set

## Objectifs:

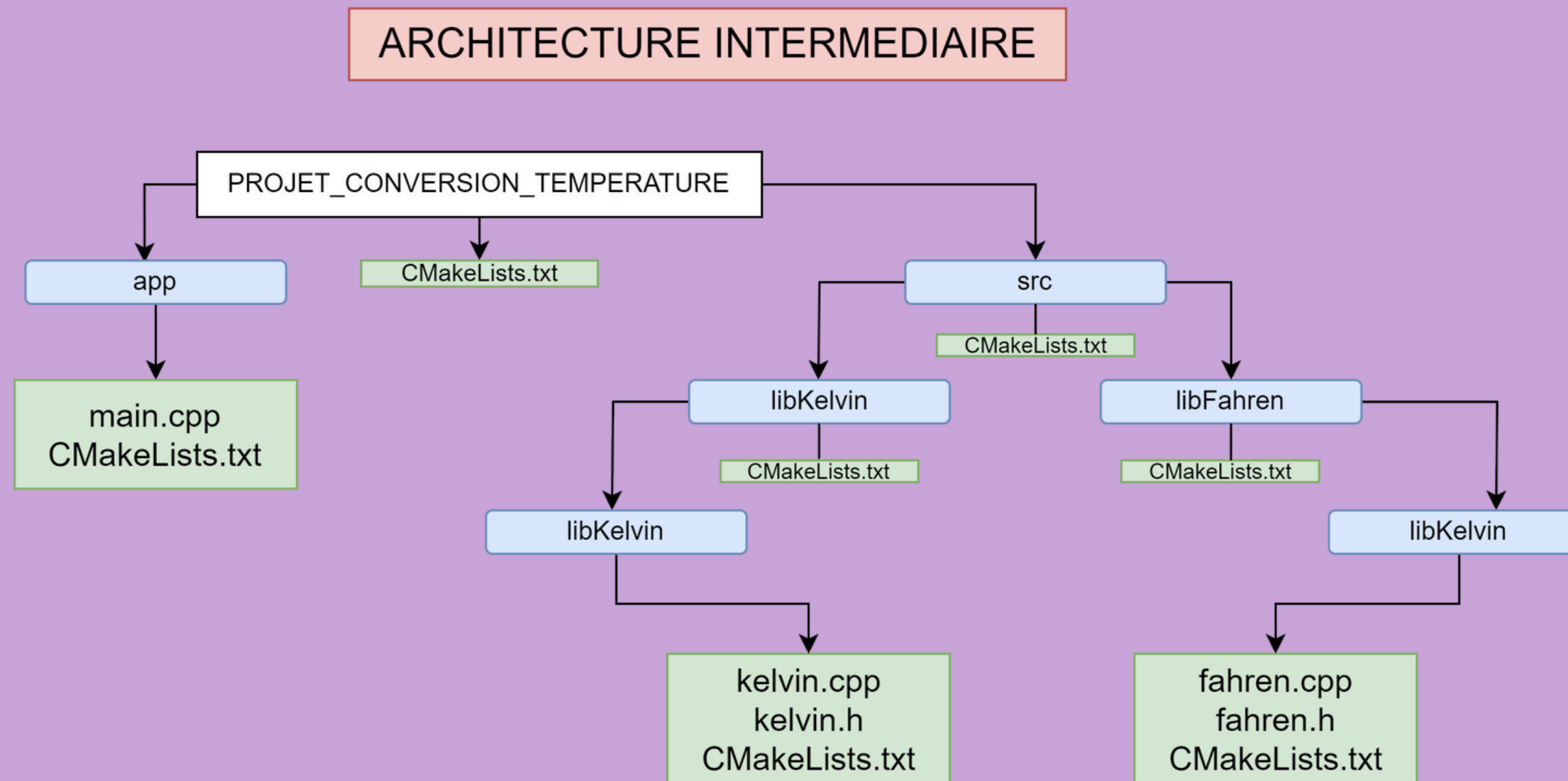
- Etre capable de créer un projet simple avec plusieurs dossiers
- Autre type d'architecture de projet
- Comprendre les variables et options de compilations
- Structurer avec les variables SOURCE et HEADER

# Module 2 : Gestion de Projets C/C++ - CMake

## ARCHITECTURE



# Module 2 : Gestion de Projets C/C++ – CMake



# Module 3 : Ajout librairies externes

## Mots clés :

find\_package, FetchContent, ExternalProject\_Add, git submodule, Conan, pkg-config, vcpkg

## Objectifs:

- Etre capable d'ajouter une librairie externe à votre projet
- Comparaison et comment choisir

# Module 3 : Ajout librairies externes

## Conclusion

# Module 3 : Comparaison – Libraires externes

GIT SUBMODULE	FETCHCONTENT	CPM (PACKAGE MANGER)	CONAN (PACKAGE MANGER)	VCPKG
<p>git submodule add "repo"</p> <p>git submodule update --init -- recursive</p> <p>Utile pour les projets non-cmake</p>	<p>include(FetchContent) FetchContent_Declare</p> <p><b>Cas d'usage:</b> Projet CMake <b>Github/ Gitlab</b></p>	<p>Release sur Github include(cmake/CPM.c make) CPMAddPackage</p> <p><b>Cas d'usage:</b> Projet CMake <b>Github/ Gitlab</b></p>	<p>pip install conan Cloner le repo Build le package conanfile.py</p> <p><b>Inconvénients:</b></p> <ul style="list-style-type: none"><li>• Installation package, python</li></ul>	<p>cloner le repo lancer .bat etc.</p> <p><b>Inconvénients:</b></p> <ul style="list-style-type: none"><li>• gestion vcpkg.cmake</li><li>• Récupération builtbaseline</li></ul>



# Module 4 – Tests unitaires et option d'optimisation

- Tests Unitaires avec CTest
- Option de Compilation et Débogage
- Optimisation des performances

# Module 5 – CMake Avancé pour Projets C/C++

- Gestion des dépendances Externes
- Création de Modules et Scripts CMake
- Bonne pratiques et techniques avancées

# Module 5 – Outils

# Conclusion