

Therapeutic beam particle generator based on IAEA phase space characteristic

HOW TO SETUP C/C++ BUILD PIPELINE TO CREATE PYTHON LIBRARIES

GOALS

- Simplify calibration proces of equipment used in radio therapy
- Reduce data volume from 1 TB -> suitable size for storing and rerun simulation (PHSP files)
- Simplify computer simulation of the therapeuetic beam
- Modernization of IAEA standard
- Reduce redundant computation (include metadata in files)

Get right tool for the job



CPP

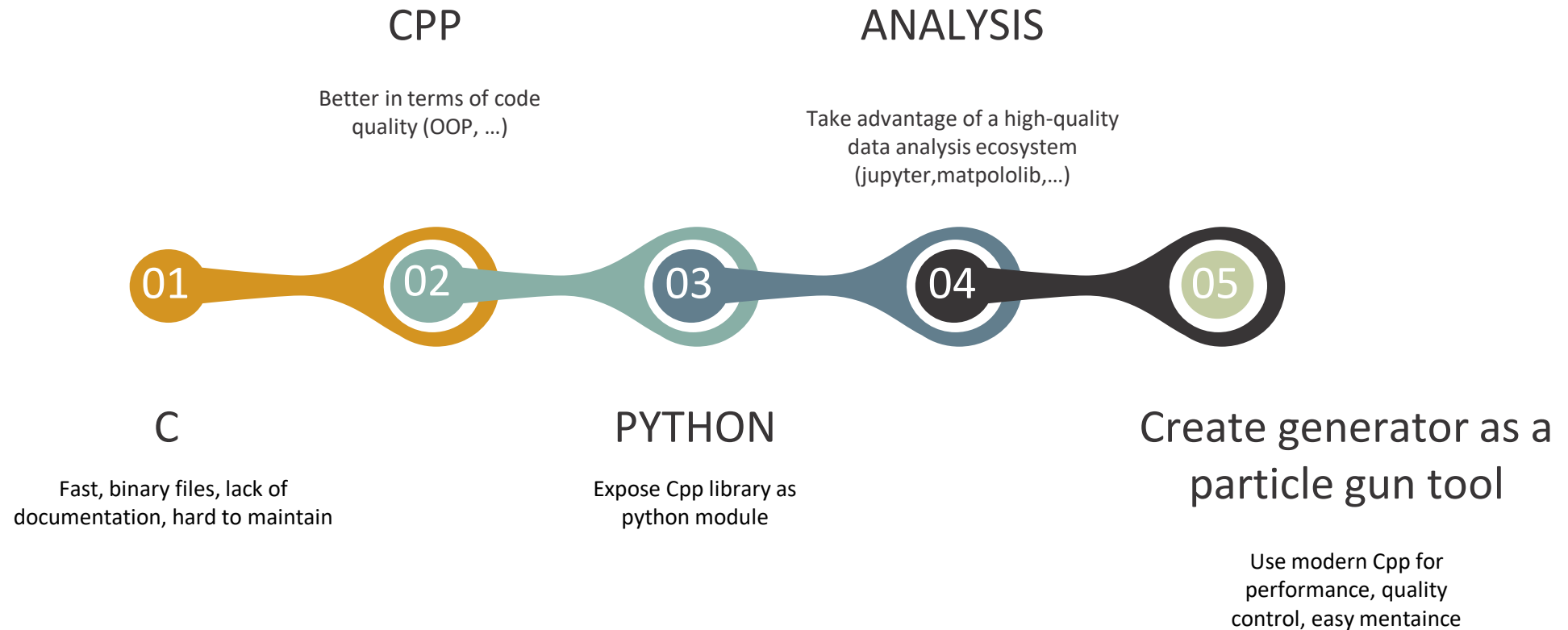
- compiled
- fast execution
- better at complex projects
- lack of standard in development tools



PYTHON

- interpreted
- fast prototyping
- better community support (libraries, documentation, forums)
- greater for integrating other technologies
- official tools (package manager, deployment system)

ROAD MAP



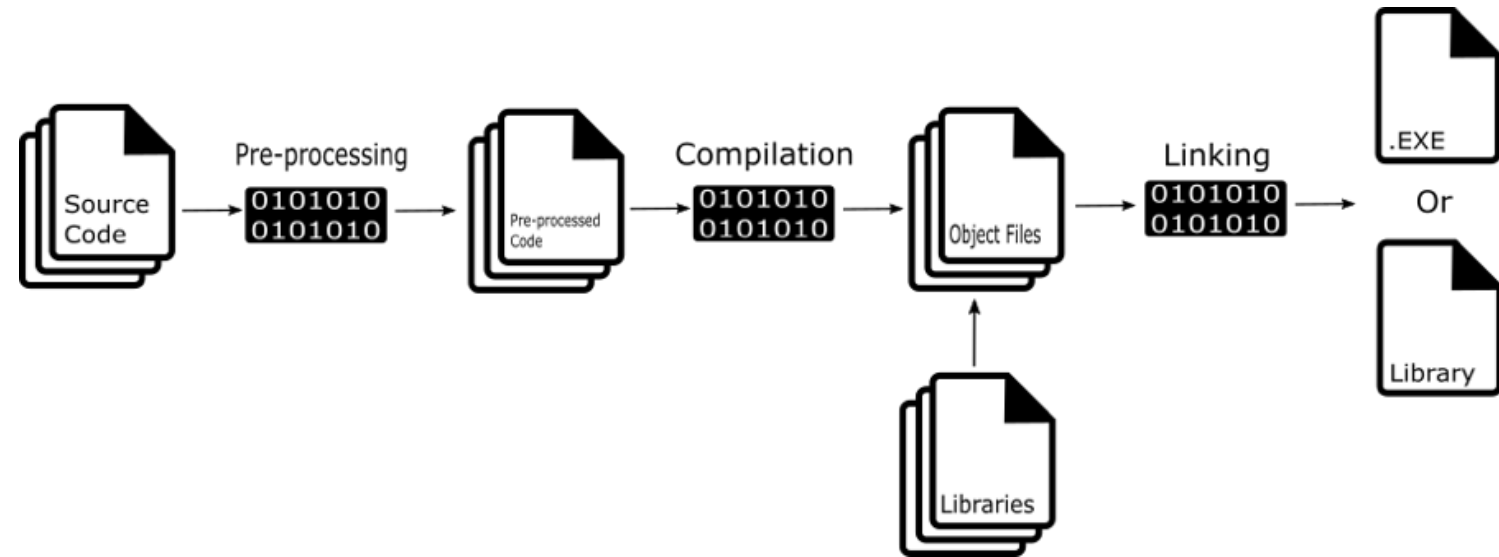
Build systems

CPP:

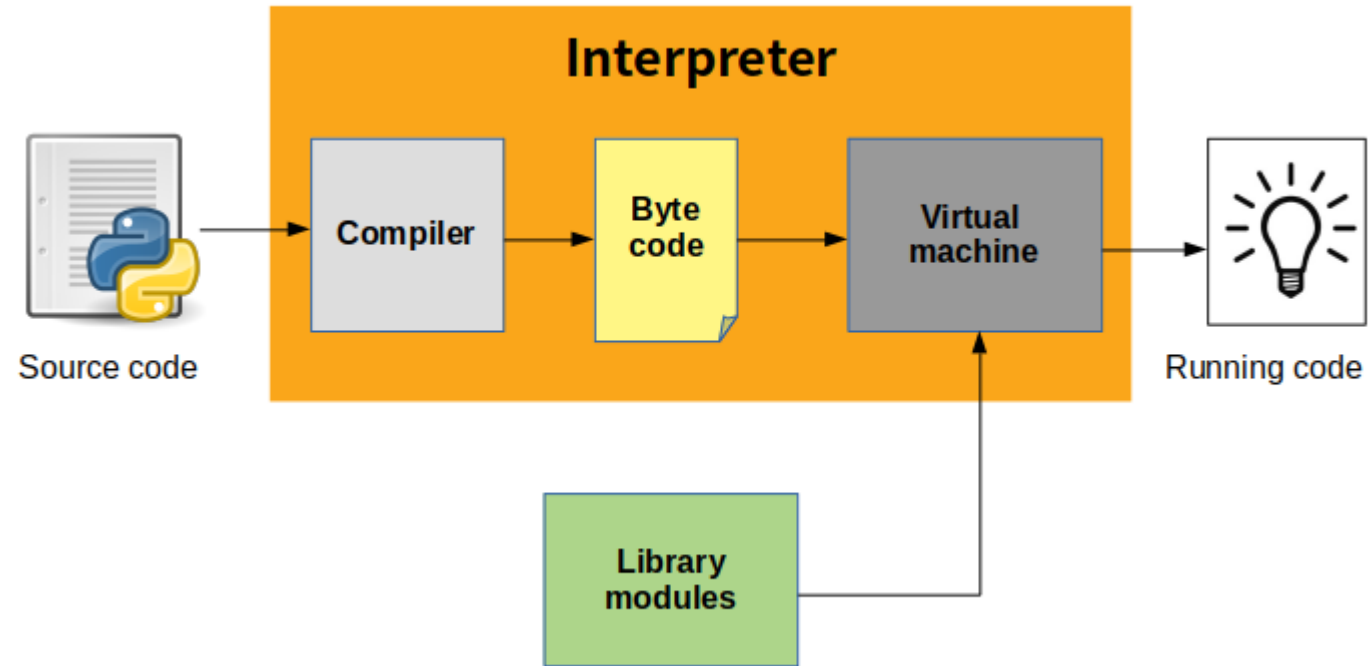
- Make
- Ninja
- MSBuild
- And dozens more

Meta Build Systems:

- CMake



Python interpreter



Python bindings

CYTHON

The approach Cython takes to creating Python bindings uses a Python-like language to define the bindings and then generates C or C++ code that can be compiled into the module.

Low-level, code written with intention to run only in python environment

BOOST.PYTHON

Library from huge(a lot of dependencies) project BOOST.

Compatible with all C++ versions. (backward compatibility)

PyBind11

Written to mimic BOOST.PYTHON, self contained support only C++ 11 and newer.

Packet managers and others deployment tools



CPP

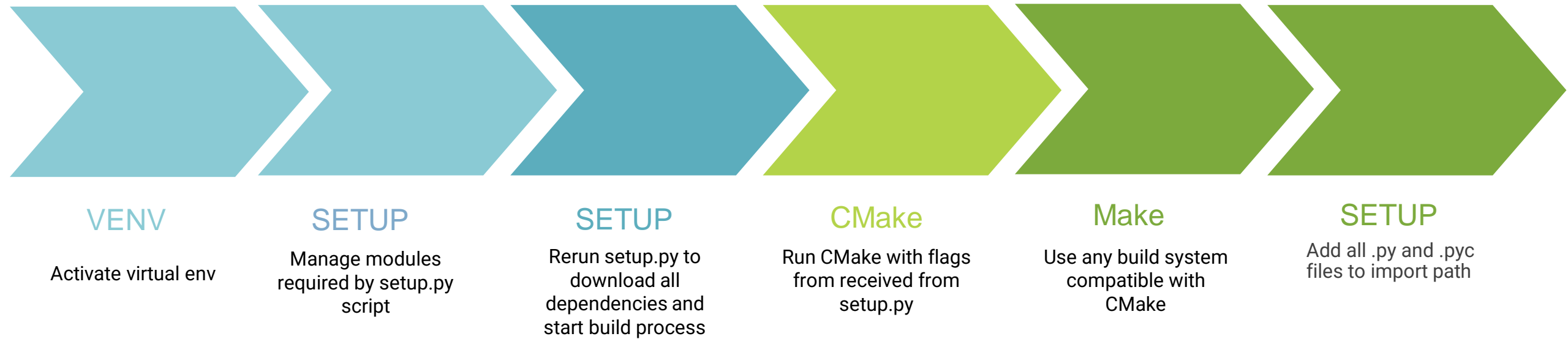
- Conan
- Vcpkg
- Conda
- and dozens more



PYTHON

- Pip
- Conda
- Venv

Build pipeline



Minimal project

<https://github.com/ZychuDev/pybind11-CMake-Setup-Project-Structure>