Conan for building C++ Tango devices

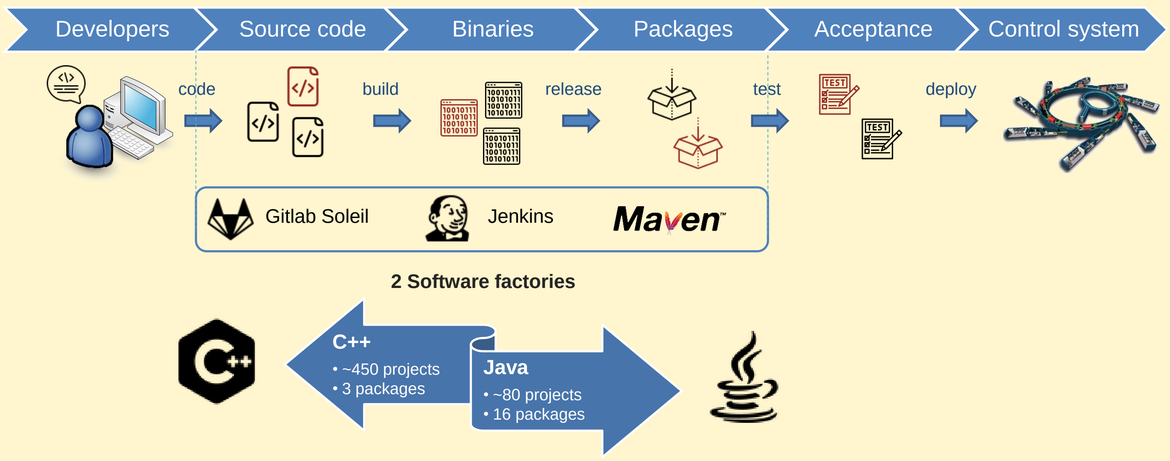
P. Madela, Y.M. Abiven, G. Abeillé, X. Elattaoui, J. Pham, F. Potier, Synchrotron SOLEIL, Gif-sur-Yvette, France

Context

Control system CI/CD

- ✓ Continuous integration to build software artifacts
- ✓ Semi-automatic continuous delivery to create packages
- ✓ Deployment of packages at each technical shutdown

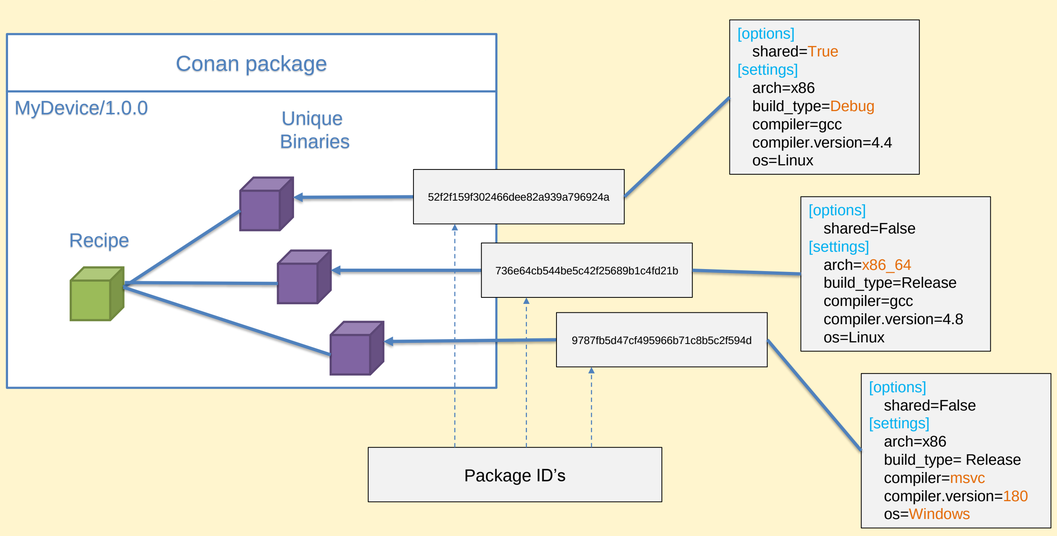
- × Components of our C++ factory are severely outdated
- × No support for future platforms
- × No support for latest C++ standards
- × No support for building third-party code
- × Maven is not common for C++ development



Strategy

Conan as a Maven alternative for C/C++

- Conan = Package and dependency manager for C/C++
- A repository system for multi-platforms and multi-binaries packages
- An abstract build system for any other build system
- Packages can be used from any build system: CMake, Make, MSBuild...
- A public central repository for the most popular open-source C/C++ libraries
- An ideal solution for C/C++ continuous integration workflows



Our forward-looking approach

- We have chosen CMake as our primary build system, which is widely adopted and versatile for both Linux and Windows.
- Conan's recipes help us include dependencies and compilation options, ensuring efficient development and packaging of our Tango device servers.
- We adopted a forward-looking strategy to address component obsolescence and future needs, creating a new infrastructure for both future and legacy 32-bit environments.

CentOS 6 32-bit
CentOS 6 64-bit
Windows 10 64-bit
Windows 7 32-bit
CentOS 7 64-bit
???

Realization

Tools and infrastructure

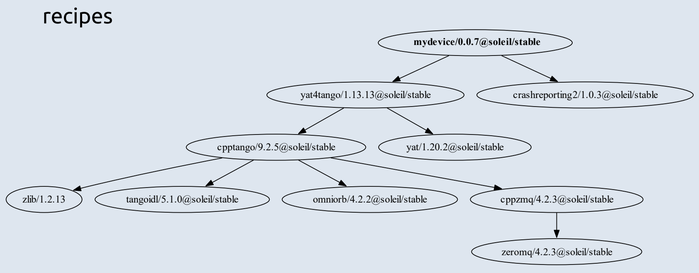
Significant effort was needed to establish essential build tools and infrastructure.

- Updated build tools for legacy Linux and Windows environments
- Created Docker images for Linux, seamlessly integrating build tools.
- Deployed new software infrastructure with key components:
  - Artifactory as a Conan repository.
  - Jenkins for automating build processes.
  - Docker-based Jenkins agents tailored for Linux.
  - Jenkins agents for Windows with essential build tools.

GitLab · JFrog ARTIFACTORY · docker · CONAN C/C++ Package Manager · CMake · git · Jenkins

Conan recipe

- Conan recipe is a python script that describes how to build and package an application, a library, a tools …
- It specifies the dependencies, build instructions, and other metadata
- It provides information about a library to consumers.
- Conan provides a collection of tools to help with building within recipes



conanfile.py

```
from conan import ConanFile
from conan.tools.cmake import CMake

class mydeviceRecipe(ConanFile):
    name = "mydevice"
    version = "0.0.7"
    license = "GPL-3.0-or-later"
    url = "https://gitlab.synchrotron-soleil.fr/mydevice"
    description = "A fantastic device"

    settings = "os", "compiler", "build_type", "arch"

    generators = "CMakeToolchain", "CMakeDeps"

    exports_sources = "CMakeLists.txt", "src/*"

    requires = "yat4tango/1.13.13@soleil/stable",
"crashreporting2/1.0.3@soleil/stable"

    def build(self):
        cmake = CMake(self)
        cmake.configure()
        cmake.build()

    def package(self):
        cmake = CMake(self)
        cmake.install()
```

Transition to Conan

Key steps:

1. **Learning Conan:**
   - Begin by learning how to use Conan.
2. **Building necessary libraries:**
   - Build the required libraries for Tango device servers.
3. **Proof of concept (POC):**
   - Validate Conan's effectiveness and compatibility.
   - Notable POCs:
     - ❑ Successfully built the Lima package on all target platforms: CentOS 6, Centos 7, Windows (32-bit/64-bit).
     - ❑ Created a Tango device with ChimeraTK, challenging on CentOS platforms.
4. **Automation:**
   - Automate parts of the package-building process.
   - Essential for managing diverse dependencies and platforms efficiently.
5. **Transition to production:**
   - Shift all Tango device server projects to Conan and CMake, replacing Maven.
   - Non-regression tests on control systems.
   - Training and transfer of knowledge to developers
6. **Next, more automation and quality:**
   - Automatic build of dependencies tree .
   - Add more unit tests.
   - Implement continuous inspection of code quality.

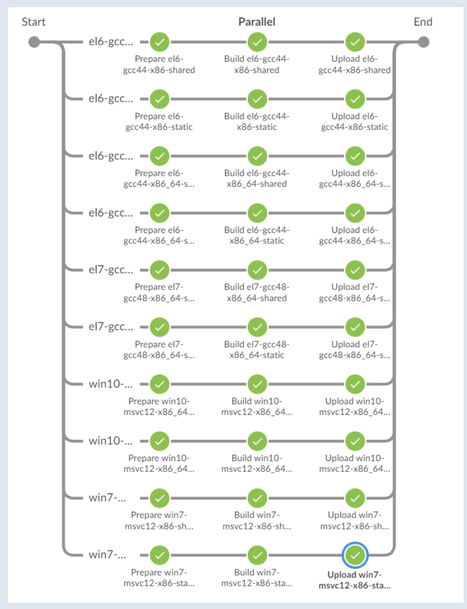


Table 1: Number of C++ projects

Type	Linux	Windows	Both
Application	1	0	0
Libraries	29	6	8
Devices	330	53	23

Benefits of adopting Conan and CMake

- Conan and Cmake as a Maven alternative for our build system.
- Simultaneously builds devices for legacy and future platforms
- Facilitates collaborations while maintaining OS independence
- Simplifies upgrading third-party libraries
- Centralized Jenkins pipelines templates
- Shared profiles/configurations between CI/CD and developer's environments
- Prepares for upcoming challenges:
  - Migration to 64-bit and newer compilers/standards.
  - Updating deployment processes.
  - Expanding CI/CD capabilities for other domains.

Well-equipped for developing future Tango device servers for the SOLEIL II upgrade with Conan and CMake.

SOLEIL, is a research center located near Paris, France. It is a particle (electron) accelerator that produces the synchrotron radiation, an extremely powerful light that permits exploration of inert or living matter. SOLEIL covers fundamental research needs in physics, chemistry, material sciences, life sciences (notably in the crystallography of biological macromolecules), earth sciences, and atmospheric sciences.
It offers the use of a wide range of spectroscopic methods from infrared to X-rays, and structural methods such as X-ray diffraction and diffusion with 29 beamlines.
It delivers 6500 hours of beam time included 5000 hours for 2000 users per year since 2008.
https://www.synchrotron-soleil.fr