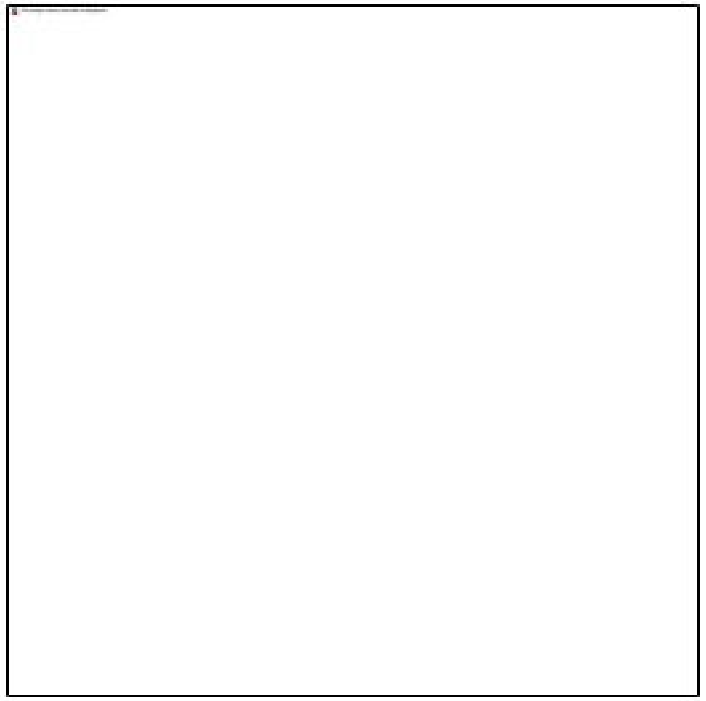
PACKAGING OPEN-SOURCE LIBRARIES WITH CONAN.IO

Konstantin Ivlev

About me



Konstantin Ivlev

Maintainer of Bincrafters community

Consultant in JFrog, conan developer

Tomsk, Russian Federation

corehard.by

WHAT IS CONAN?

- FOSS (MIT License)
- Integration with Visual Studio, CMake, XCode, etc.
- Distributed
- Pre-built packages or build from source
- Cross-platform
- Supports Cross-Compilation
- 100+ contributors, 2K+⭐ (GitHub)
- Used in production by hundreds of companies

USING CONAN

- Let’s write some GUI application using wxWidgets
- Declare dependencies
- Fetch packages
- Build
- Profit

```
#include "wx/wx.h"
class HelloWorldApp : public wxApp {
 public:
     bool OnInit() {
                 wxFrame *frame = new wxFrame((wxFrame*) NULL, -1,
          _T("use wxWidgets from Conan.io"));
                 frame->CreateStatusBar();
                 frame->SetStatusText(_T("Hello World"));
                 frame->Show(true);
                 SetTopWindow(frame);
                 return true;
          }
};

DECLARE_APP(HelloWorldApp)
IMPLEMENT_APP(HelloWorldApp)
```

CONANFILE.TXT

```
[requires]
wxwidgets/3.1.1@bincrafters/stable

[generators]
cmake
```

CONAN IN ACTION

$ conan install .

```
wxwidgets/3.1.1@bincrafters/stable: Not found in local cache, looking in remotes...
wxwidgets/3.1.1@bincrafters/stable: Trying with 'conan-center'...
wxwidgets/3.1.1@bincrafters/stable: Trying with 'bincrafters'...
Downloading conanmanifest.txt
[==================================================] 166B/166B
Downloading conanfile.py
[==================================================] 16.8KB/16.8KB
Downloading conan_export.tgz
[==================================================] 760B/760B
Decompressing conan_export.tgz: 100%|██████████|
libpng/1.6.34@bincrafters/stable: Not found in local cache, looking in remotes...
```

CONANBUILDINFO.CMAKE

- Include directories
- Library directories
- Libraries
- Compile definitions

CONANBUILDINFO.CMAKE

$ **cat conanbuildinfo.cmake**

```
set(CONAN_WXWIDGETS_ROOT
"C:/Users/SSE4/.conan/data/wxwidgets/3.1.1/bincrafters/stable/package/c4c7f302ce")
set(CONAN_INCLUDE_DIRS_WXWIDGETS
"C:/Users/SSE4/.conan/data/wxwidgets/3.1.1/bincrafters/stable/package/c4c7f302ce/include"
"C:/Users/SSE4/.conan/data/wxwidgets/3.1.1/bincrafters/stable/package/c4c7f302ce/include/msvc")
set(CONAN_LIB_DIRS_WXWIDGETS
"C:/Users/SSE4/.conan/data/wxwidgets/3.1.1/bincrafters/stable/package/c4c7f302ce/lib")
set(CONAN_LIBS_WXWIDGETS wx_gtk2u_xrc-3.1 wx_gtk2u_webview-3.1)
set(CONAN_COMPILE_DEFINITIONS_WXWIDGETS "wxUSE_GUI=1")
```

CONAN_BASIC_SETUP

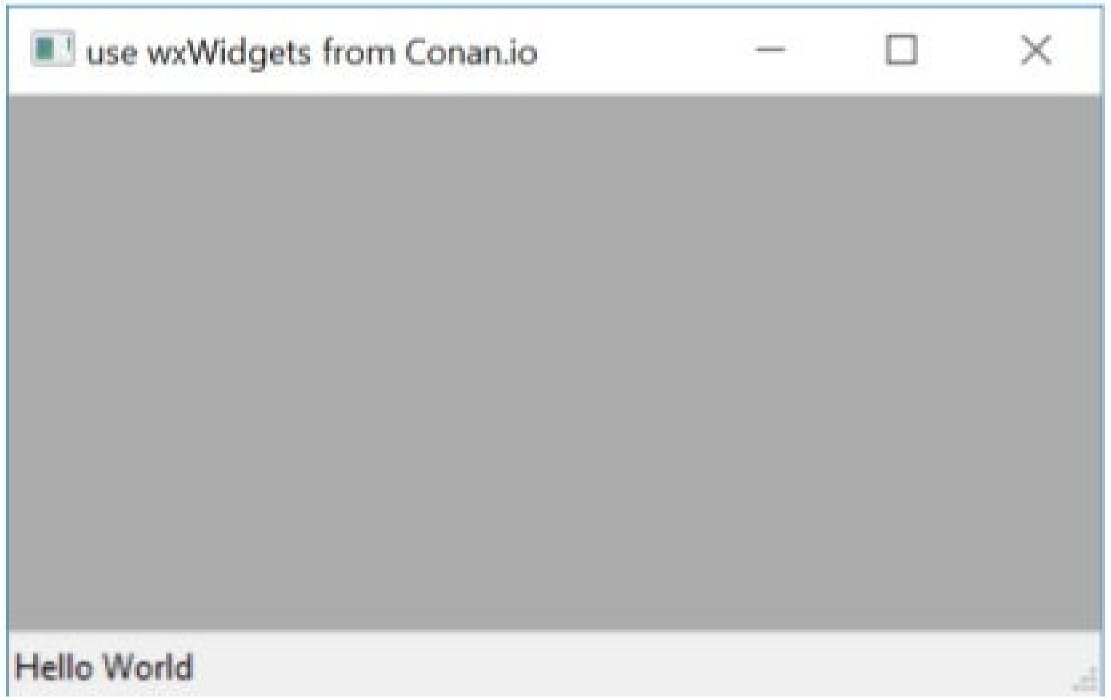
```
cmake_minimum_required(VERSION 3.1)
project(UseWxWidgetsFromConan CXX)

include(${CMAKE_CURRENT_BINARY_DIR}/conanbuildinfo.cmake)
conan_basic_setup(TARGETS)

add_executable(${PROJECT_NAME} WIN32 main.cpp)
target_link_libraries(${PROJECT_NAME} PRIVATE
                                       CONAN_PKG::wxwidgets)
```

NOW BUILD AND RUN

```
$ cmake . -G "Visual Studio 15 2017 Win64"
$ cmake --build . --config Release
```

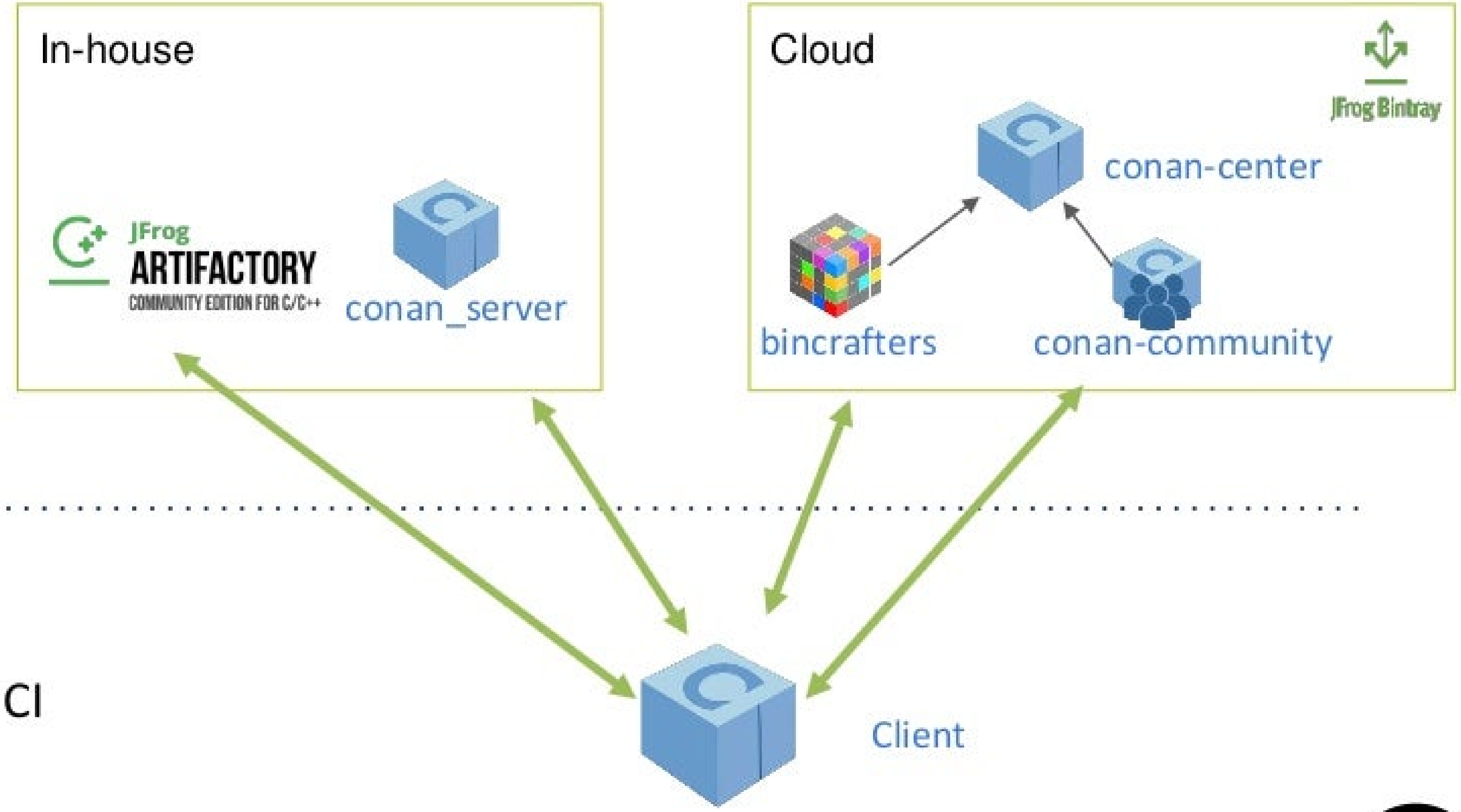


Hello World

CONAN IS DECENTRALIZED

Servers
(artifact storage)

Client
Developer machine / CI



REMOTES: CONAN-CENTER

https://bintray.com/conan/conan-center

- Default remote
- Official, carefully-moderated repository
- Packages from library authors
- Currently ~120 packages available
- Goal is to double amount by the end of 2018

REMOTES: CONAN-COMMUNITY

https://github.com/conan-community/community

- Supported by conan developers
- Incubator for conan-center inclusion
- + 30 additional packages
- Foundational packages (zlib, OpenSSL, boost, etc.)

REMOTES: BINCRAFTERS

https://github.com/bincrafters/community

- Supported by OSS community
- Incubator for conan-center inclusion
- +200 additional packages
- Accept your packages for support
- Complex libraries (Qt, wxWidgets, Wt++, ffmpeg, ImageMagick, SDL2, CppRestSDK, ZeroMQ, etc.)

ARTIFACTORY CE FOR C++

https://conan.io/downloads

- Free (CE=Community Edition), also for commercial purposes
- Run your own package server, in-house
- Easy to install
- Scalability
- Web UI
- Permissions management
- Concurrency

SEARCHING FOR SOMETHING

```
$ conan search Qt* -r all

Existing package recipes:

Remote 'bincrafters':
Qt/5.11.0@bincrafters/stable
Qt/5.11.1@bincrafters/stable
Qt/5.11.2@bincrafters/stable
```

CREATING A PACKAGE

- How to get my own library packaged?
- How to write recipe?
- How to build it for all configurations?
- How to check packages are good?

CONANFILE.PY

Attributes:

- name
- version
- settings
- options
- default_options

Methods:

- source( )
- build( )
- package( )
- package_info( )
- package_id( )

EXAMPLE OF CONANFILE.PY (GSL)

```
class GslMicrosoftConan(ConanFile):
    name = "gsl_microsoft"
    version = "2.0.0"
    description = "Functions use by the C++ Core Guideline"
    url = "https://github.com/bincrafters/conan-gsl_microsoft"
    license = "MIT"
    exports = ["LICENSE.md"]
    no_copy_source = True
    _source_subfolder = "source_subfolder"
```

EXAMPLE OF CONANFILE.PY (GSL)

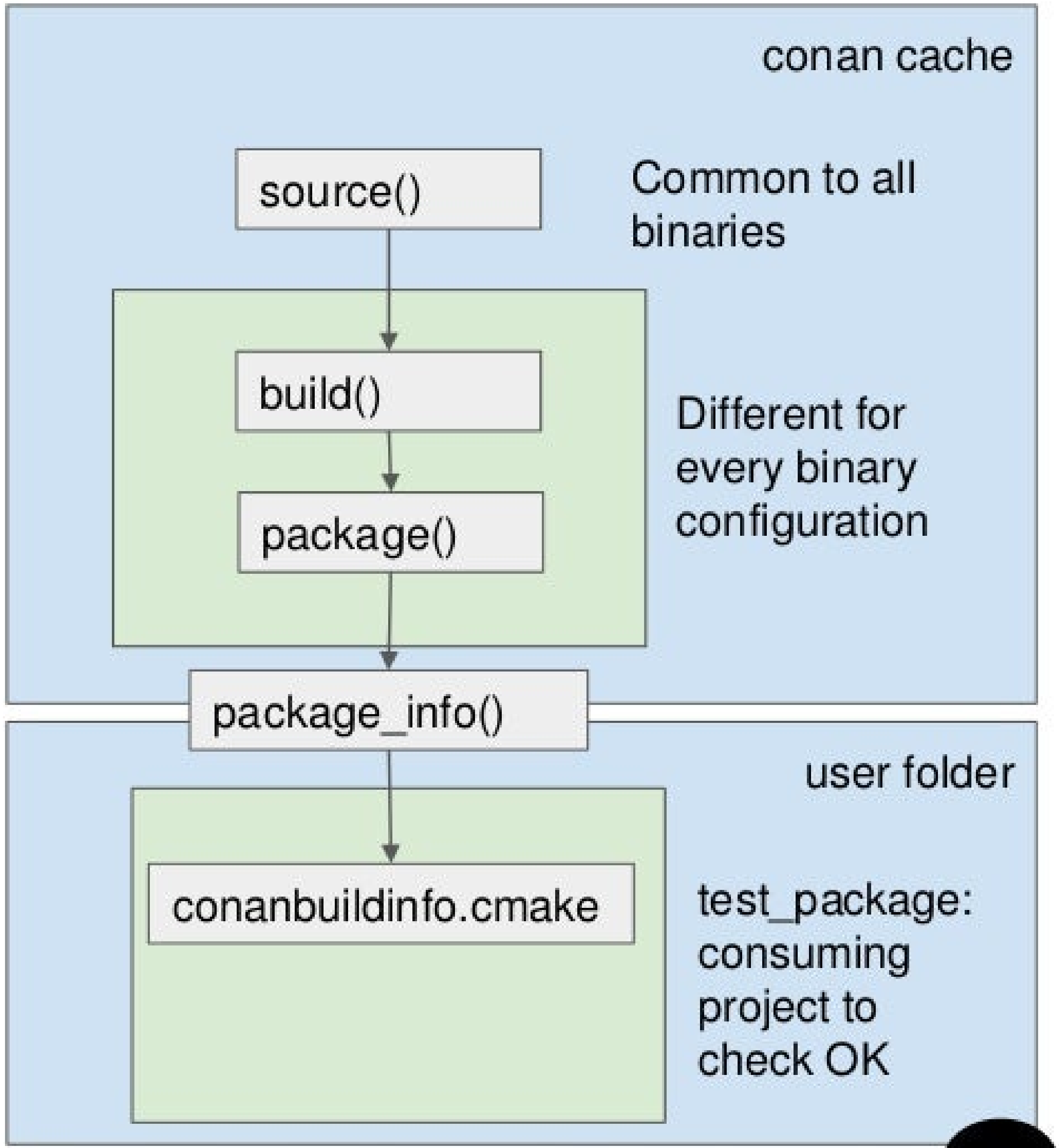
```
def source(self):
    tools.get("https://github.com/Microsoft/GSL/archive/v2.0.0.zip")
    extracted_dir = "GSL-" + self.version
    os.rename(extracted_dir, self._source_subfolder)
```

EXAMPLE OF CONANFILE.PY (GSL)

```
def package(self):
    include_folder = os.path.join(self._source_subfolder, "include")
    self.copy("LICENSE", dst="licenses", src=self._source_subfolder)
    self.copy(pattern="*", dst="include", src=include_folder)
```

CONAN CREATE

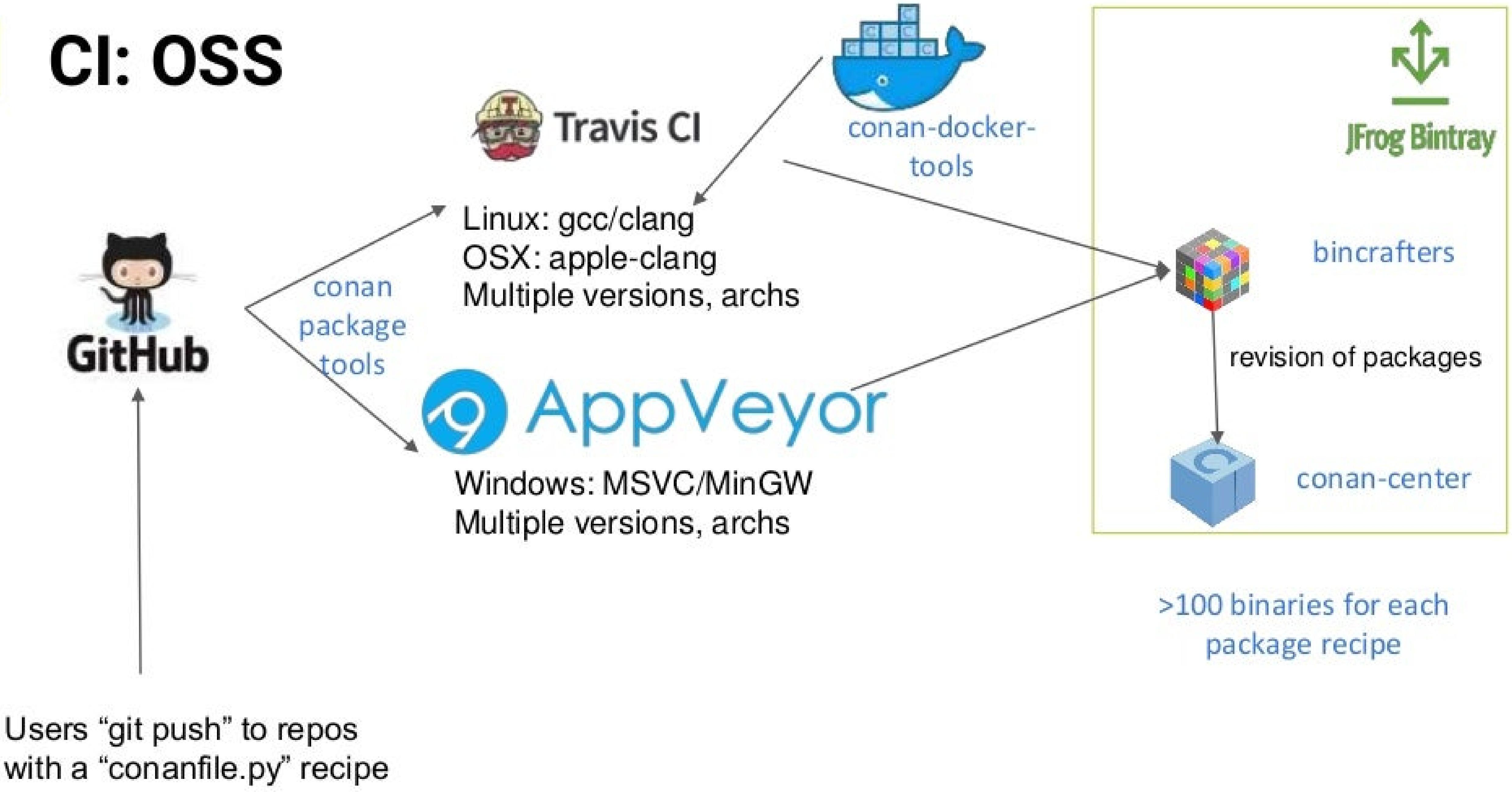
- source
- build
- package
- package_info
- test package



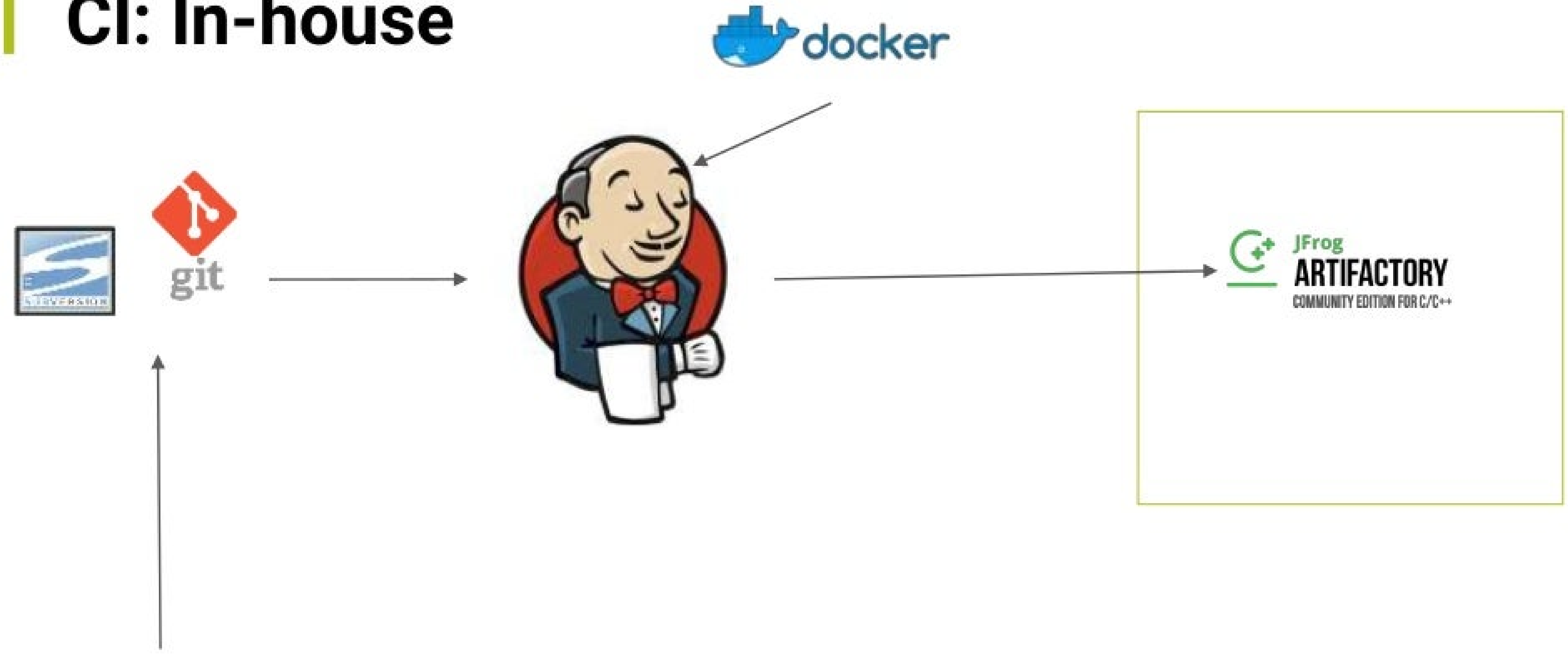
CHALLENGE: VARIOUS CONFIGURATIONS

- OS (Windows/Linux/macOS)
- Architecture (x86, x86_64)
- Compilers (MSVC, GCC, Clang)
- Release and Debug
- Shared and Static
- Hard to build everything on single machine
- Need to automate somehow

CI: OSS


Travis CI
conan-docker-tools
JFrog Bintray
Linux: gcc/clang
OSX: apple-clang
Multiple versions, archs
conan package tools
GitHub
bincrafters
revision of packages
AppVeyor
Windows: MSVC/MinGW
Multiple versions, archs
conan-center
>100 binaries for each package recipe
Users "git push" to repos with a "conanfile.py" recipe

CI: In-house



Users “git push” to repos
with a “conanfile.py” recipe

CI: TRAVIS AND APPVEYOR

- Register via GitHub account
- Add **.travis.yml** and **appveyor.yml**
- Consider **conan new** command
- Or copy from **bincrafters-templates**

CPT: CONAN PACKAGE TOOLS

https://github.com/conan-io/conan-package-tools

```
$ pip install conan-package-tools
$ pip install bincrafters-package-tools
```

- build packages for all possible configurations
- run build, test and upload
- integration with various CI systems
- python module

CDT: CONAN DOCKER TOOLS

https://github.com/conan-io/conan-docker-tools
https://hub.docker.com/r/conanio/

- various compilers
- conan pre-installed
- x86, x64 and ARM images
- Recently Windows images were added

CHALLENGE: VARIOUS BUILD SYSTEMS

- CMake
- GNU autotools
- Plain makefiles
- Visual Studio projects
- Something else specially invented

BUILD HELPERS

- CMake
- AutoToolsBuildEnvironment
- MSBuild

BUILD HELPERS: CMake

```
def _configure_cmake(self):
    cmake = CMake(self)
    cmake.definition['ENABLE_LIBASTRAL'] = self.options.libastral
    cmake.configure(build_dir=self._build_subfolder)
    return cmake

def build(self):
    cmake = self._configure_cmake()
    cmake.build()

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

BUILD HELPERS: AutoToolsBuildEnvironment

```
def build(self):
    configure_args = []
    if self.options.shared:
        configure_args.extend(['–disable-static', '--enable-shared'])
    else:
        configure_args.extend(['–disable-shared', '--enable-static'])
    env_build = AutoToolsBuildEnvironment(self)
    env_build.configure(args=configure_args)
    env_build.make()
    env_build.install()
```

BUILD HELPERS: MSBuild

```
def build(self):
    sln = 'libtheora_dynamic.sln' if self.options.shared else \
                    'libtheora_static.sln'
    msbuild = MSBuild(self)
    platforms = {'x86': 'Win32', 'x86_64': 'x64'}
        msbuild.build(sln, upgrade_project=True, platforms=platforms)
```

CHALLENGE: BUILD TOOLS

build_requires:

- ninja_installer
- nasm_installer
- yasm_installer
- msys2_installer
- cygwin_installer
- mingw_installer
- premake_installer
- gyp_installer

PROFILES

- Define set of conan settings and options
- Allow to specify build requirements (like nasm)
- Allow to specify environment variables (e.g. CC, CXX)
- Extremely useful for cross-compilation

Examples:

- Clang-CL
- MinGW
- Raspberry Pi
- iOS
- Android
- Windows Phone
- Emscripten

PROFILE EXAMPLE: CLANG-CL + NINJA

```
[settings]
compiler=clang
compiler.version=6.0
compiler.libcxx=libstdc++
[env]
CC=clang-cl.exe
CXX=clang-cl.exe
CFLAGS= -fms-compatibility-version=1800
CXXFLAGS= -fms-compatibility-version=1800
PATH=[C:\Program Files\LLVM\bin]
[build_requires]
ninja_installer/1.8.2@bincrafters/stable
```

CHALLENGE: CHECKING CORRECTNESS

Use conan hooks (former plug-ins):

- Conan-center guidelines check
- Update BinTray & GitHub metadata
- Binary linter
- Signing
- License check

RESULTS

- >300 libraries packaged, with hundreds of binaries for each one. Largest repository of C++ binaries.
- Modular Boost
- >1M downloads per month
- Library authors are adopting conan:
    - Range V3
    - FlatBuffers
    - Poco

ONGOING WORK

- Need to add new compilers to all libraries
- How to track updates for libraries?
- Finish review for most-wanted complex libraries and their dependencies (Qt, wxWidgets, ffmpeg, ImageMagick, CppRestSDK, Folly, Abseil, etc.)
- Start work on more challenging libraries (LLVM, GStreamer, VTK, etc)

JOIN US

- Try conan
- Provide feedback (GitHub, Slack)
- Request missing libraries
- Try to write recipes for libraries
- Share your recipes
- Submit inclusion requests to conan-center

WISHLIST

https://croydon.github.io/conan_inquiry/#!wishlist

- Vote ⭐
- Add packages you need 🖤

GETTING HELP ON SLACK

https://cpplang-inviter.cppalliance.org/

Channels:

- **#CONAN**
- **#BINCRAFTERS**

Many thanks!

Questions welcome :)

Konstantin Ivlev tomskside@gmail.com +7 999 620 00 00