



ENHANCED C/C++ PACKAGE MANAGEMENT MADE EASY WITH CONAN 2.0

Christopher McArthur, Conan Developer Advocate
Diego Rodriguez-Losada Gonzalez, Conan Co-Founder



Introduction!



Christopher McArthur

Former Conan User and long time contributor

“I’ve been opening enterprise issues for Conan over the [past 3 years](#)”



Diego Rodriguez-Losada Gonzalez

Conan Co-Founder and Maintainer

“I have been resolving enterprise issues for Conan over the past 6 years”



CONAN

C/C++ Package Manager



CONAN
C/C++ Package Manager



What is Conan?



CONAN

C/C++ Package Manager

C and C++ Package Manager

What is the role of a package manager?

- Easily install dependencies
 - `conan install`
`--requires=spdlog/1.11.0`

How is Conan different?

- Enable you to build and distribute binaries

We offer **JFrog's ConanCenter** with 1500+ open-source projects with over 100 configuration (os, compiler, arch) being created and publish to help accelerate open-source.

- Open-Source (MIT license)
- Distributed (1 client many servers)
- Scalable and flexible
- Remotes + Server

Supports

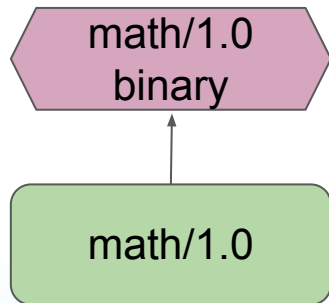
- CMake, Meson, Autotool, etc...
- Any Platforms
- More than just CMakeLists.txt



CONAN

C/C++ Package Manager

Conanfile: A package “recipe”



```
$ git clone ... math && cd math  
$ conan create .
```



math/conanfile.py

```
from conan import ConanFile  
  
class Math(ConanFile):  
    name = "math"  
    version = "1.0"  
  
    def source(self): ...  
    def build(self): ...  
    def package(self): ...
```



Introduction to the new Tutorial Section

<https://docs.conan.io/2/tutorial.html>

**CONAN 2.0**
C/C++ Package Manager

Search docs

[Introduction](#)
[What's new in Conan 2.0](#)
[Install](#)

☰ **Tutorial**

[Consuming packages](#)
[Creating packages](#)
[Working with Conan repositories](#)
[Developing packages locally](#)
[Versioning](#)
[Other important Conan features](#)

[Integrations](#)
[Examples](#)

[Docs](#) » [Tutorial](#)

[Edit on GitHub](#)

Tutorial

The purpose of this section is to guide you through the most important Conan features with practical examples. From using libraries already packaged by Conan, to how to package your libraries and store them in a remote server alongside all the precompiled binaries.

- [Consuming packages](#)
 - [Build a simple CMake project using Conan](#)
 - [Using build tools as Conan packages](#)
 - [Building for multiple configurations: Release, Debug, Static and Shared](#)
 - [Understanding the flexibility of using conanfile.py vs conanfile.txt](#)
 - [How to cross-compile your applications using Conan: host and build contexts](#)
 - [Introduction to versioning](#)
- [Creating packages](#)
 - [Create your first Conan package](#)
 - [Handle sources in packages](#)
 - [Add dependencies to packages](#)



CONAN

C/C++ Package Manager

Why Conan?

- Key difference between Conan and other C/C++ package manager is the focus on binaries. Being able to create packages that can be **re-used** across multiple teams throughout an organization. Enables flexibility and scalability. Framework for doing DevOps and Package Management in a very enterprise ready manner.

You can model **platform configurations** and the **linkage between libraries**. To deterministically know what to build but more importantly what you already have to save time and money!



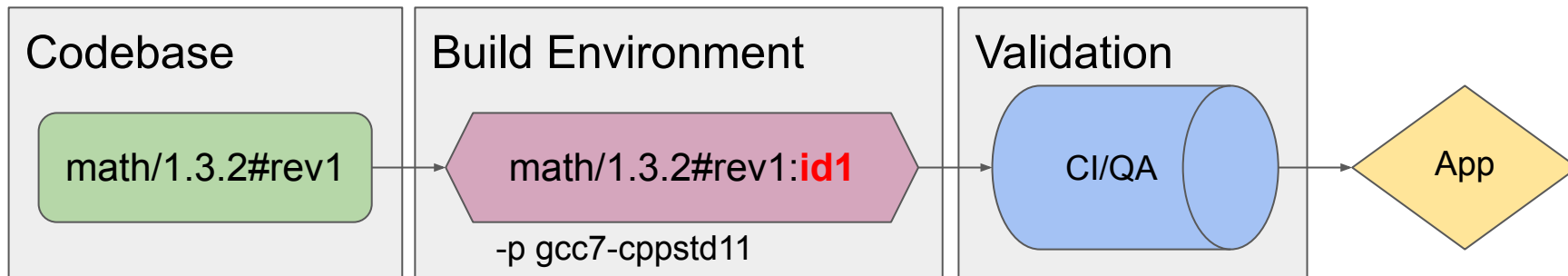
CONAN

C/C++ Package Manager

A common problem

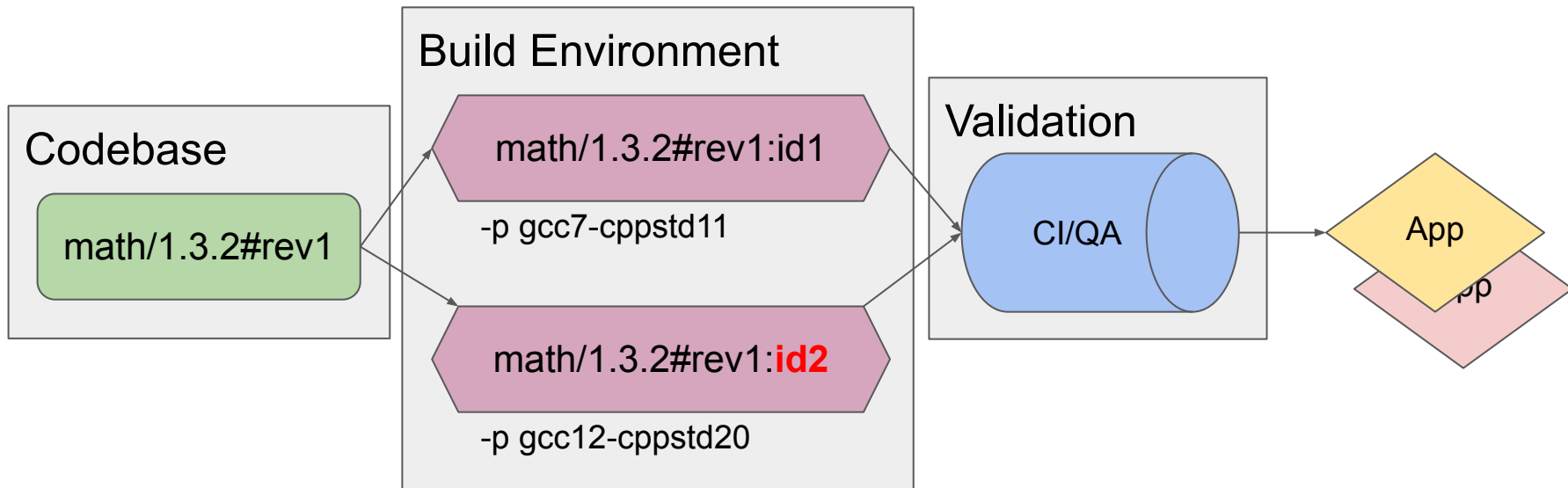
- Most C++ developers are still running C++11- but in a perfect world who doesn't want to update?

Today (build and shipped to customer)



A common solution

Future (Current development being validated along side LTS)



What's new in Conan 2.0



CONAN

C/C++ Package Manager

Everything is new!



1.0

5 years, without breaking

60% new code, 20%
backports

1.X \Leftrightarrow 2.0 compatible syntax
subset



2.0

What to expect?

- Limited installer support - more to come
- Default home is now `~/.conan2`
- Cache layout changes
 - Temporary build folder
- Changes to `settings.yml`
- Opt-in default profile
- Always using two profiles, statically provided
- Unified command reference syntax
 - Lots of command line changes
- Generators
- Environments and scopes
- Structured outputs, serializations, custom formatters
- Private and hidden requirements by default
- Handling conflicts resolution
- **New graph resolution**
- Requirement traits
- Package types
- Binary Model and package IDs
- **Extensions**
 - Hooks
 - Profiles Checker
 - Command Wrapper
 - Package Signing
 - Compatibility
 - Deployers
 - Custom Commands
- Python API
- **Lockfiles**
- User settings

Everything is on docs.conan.io



CONAN

C/C++ Package Manager

We have been listening to you.




CONAN

C/C++ Package Manager

Usage and Feedback

- ~800K downloads/month from PyPI
- Designated as PyPI critical project (1% of most downloaded in whole PyPI)

 **Analytics**

Overview Channels **Members**

Data as of 02/16/2023, last updated 3 days ago

192 channels Export CSV Edit columns Last 30 Days Filter by channel name

Name	Created	Total membership	Messages posted	Members who posted	Members who viewed
# general	2016-08-16	22,292	3,097	107	739
# conan	2017-02-05	2,399	2,850	85	223
# boost	2016-09-02	2,874	5,087	54	178
# cmake	2017-06-21	3,751	1,018	49	230
# learn	2016-10-21	6,116	1,684	42	231
# off-topic	2017-11-17	919	2,707	31	75

PyPI Stats

[Search](#)

[All packages](#)

[Top packages](#)

[Track packages](#)

conan

[PyPI page](#)

[Home page](#)

Author: JFrog LTD

License: MIT

Summary: Conan C/C++ package manager

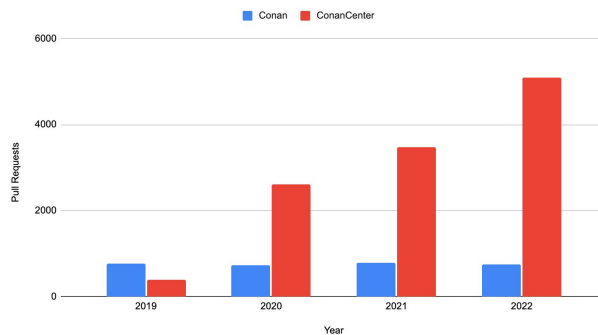
Latest version: 2.0.0

Downloads last day: 15,503

Downloads last week: 217,579

Downloads last month: 840,656

Conan and ConanCenter Pull Requests

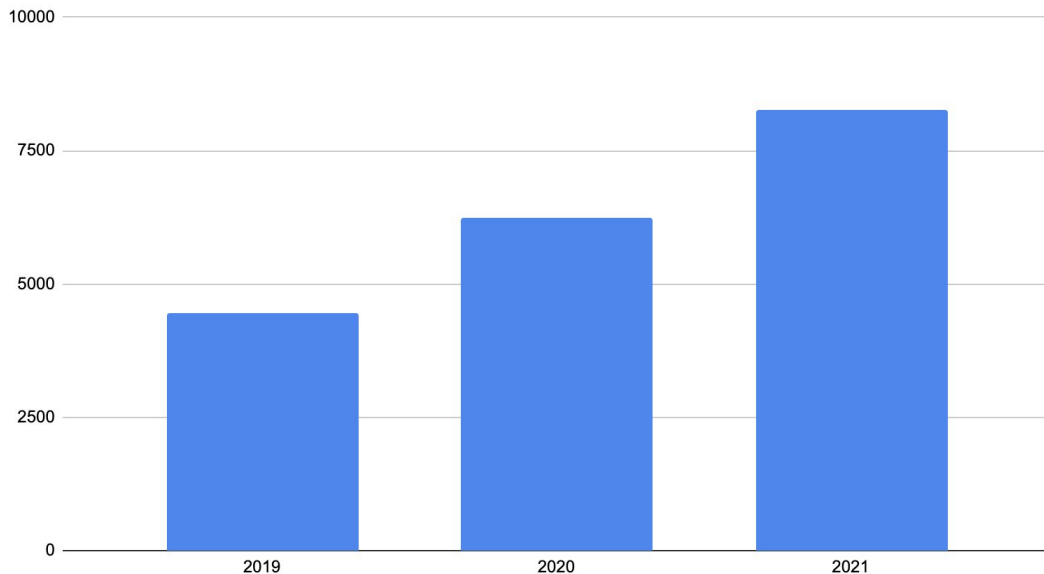


Support

+2000 Github issues / year

100 hr/year user video calls

Direct support (slack, almost daily)



Artifactory servers running Conan in production
and telemetry enabled (no firewalls)



CONAN

C/C++ Package Manager

Tribe 2.0 (conan.io/tribe.html)

Bose

TomTom

Continental

Nasa

Apple

Ansys

Bloomberg

Rohde & Schwarz

Bosch

ASAP

Rti

Zeiss

Nasdaq

Plex

Keysight

Datalogics

VMWare

... 50 more



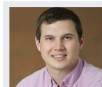
Alban Lefebvre

Bloomberg

Software Engineer at Bloomberg in Lugano, Switzerland. One of my focus is SDLC and in particular improving our Windows Build infrastructure.



[View Profile](#)



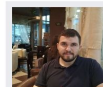
Alex Brinkman

NASA JPL

Robotics software developer at NASA JPL leveraging Conan to develop C++ manipulation applications.



[View Profile](#)



Alexander Krutikov

ConTech

SRE at ConTech. Over 10 years of C++ development experience. I design C/C++, embed code guidelines, analyze software architecture.



[View Profile](#)



Alexandr Timofeev

COO HTR WLCZ

C++ developer with experience in touch-screen OHS applications for testing of avionics and related systems.



[View Profile](#)



Andreas Hader-Kregl

ENGEL Austria GmbH

I am a software developer and architect at ENGEL Austria. I have a master's degree in software engineering and been working as a software developer since 2011. My main...



[View Profile](#)



Andreas Kleber

ESI Group

Starting as C++ developer 12 years ago. I moved more and more to DevOps topics since about 7 years ago and I am now a DevOps Engineer since about 2.5 years. My tag focus...



[View Profile](#)



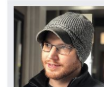
Ayaz Salikhov

AIM Tech

I create a low-latency high-frequency trading platform. I'm in love with C++ and Conan so far. I do believe that the C++ world should be better and try to help that.



[View Profile](#)



Chris Robinson

ANSYS

ANSYS employee. Use Conan in our software builds. Support use of Conan throughout the company. Streamline Conan deployment within the organization.



[View Profile](#)



Claudio Bantaloukas

CCDC

I'm covering a DevOps and Software Engineer role at CCDC, with a focus on build maintenance. I break down complex interdependencies, using Conan to deal with complexity.



[View Profile](#)



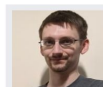
Cuong Trinh

Emviiv

I'm principal software engineer and also develop engineer. I'm responsible to build and maintain the CCDC system of the company. I have experience on mobile platforms...



[View Profile](#)



Daniel Roberts

Bose

Software Engineer with a wealth of experience in embedded software development and a passion for good design and system architecture.



[View Profile](#)



Eric Pedersen

Tradeweb Markets

I am a software developer working in finance. My focus over the last few years has been DevOps.



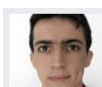
[View Profile](#)



Fabian Sturm

Rohde & Schwarz

I am a long time software developer and project lead at Rohde & Schwarz.



Fabien Laurent

ASAP

I am software engineer at ASAP GmbH focusing on C++ and DevOps.



Gayan Pathiraga

LSRG Technology

I'm a C++ and Python developer with about 14 years of experience in embedded software development.



Glenn Duffy

Bose

Software Engineer with 10 years of experience with embedded software development.

Overview

- Some of the Feature Highlights:
 - Binary and Graph
 - Extensions and Commands
 - Reproducibility with Lockfiles
- Conclusions



1. Learning to Fly



C and C++ are challenging

The first preview version of Conan was released in late 2016. Since then, Conan has experienced significant continued growth: empowering teams to bring their C++ projects into the DevOps era.

- **New documentation, written from scratch**

- [Hands-on tutorial](#) covering the most common use cases
- Comprehensive and better organized [API reference](#)
- Newly structured [examples section](#)
- Documented way to [deploy Conan without Python](#) in the developers computers

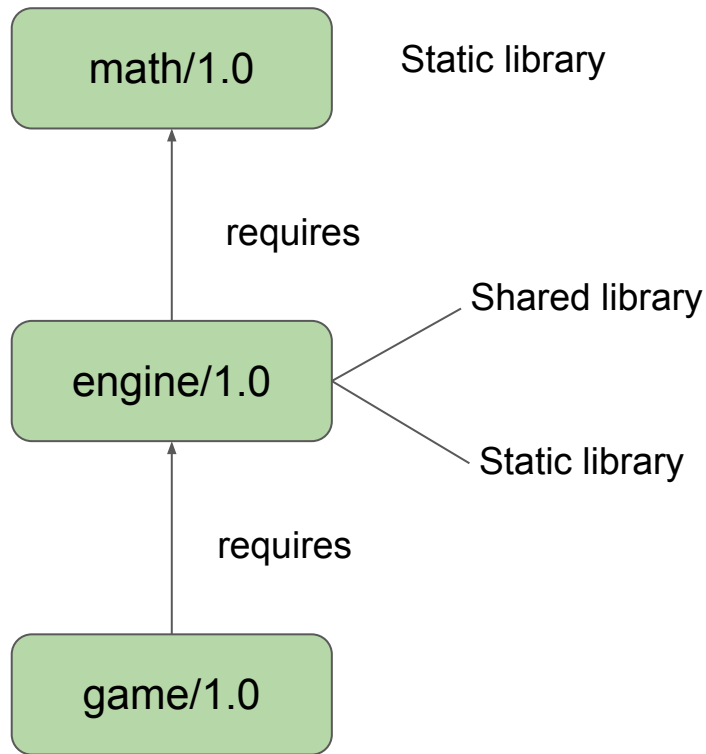


CONAN

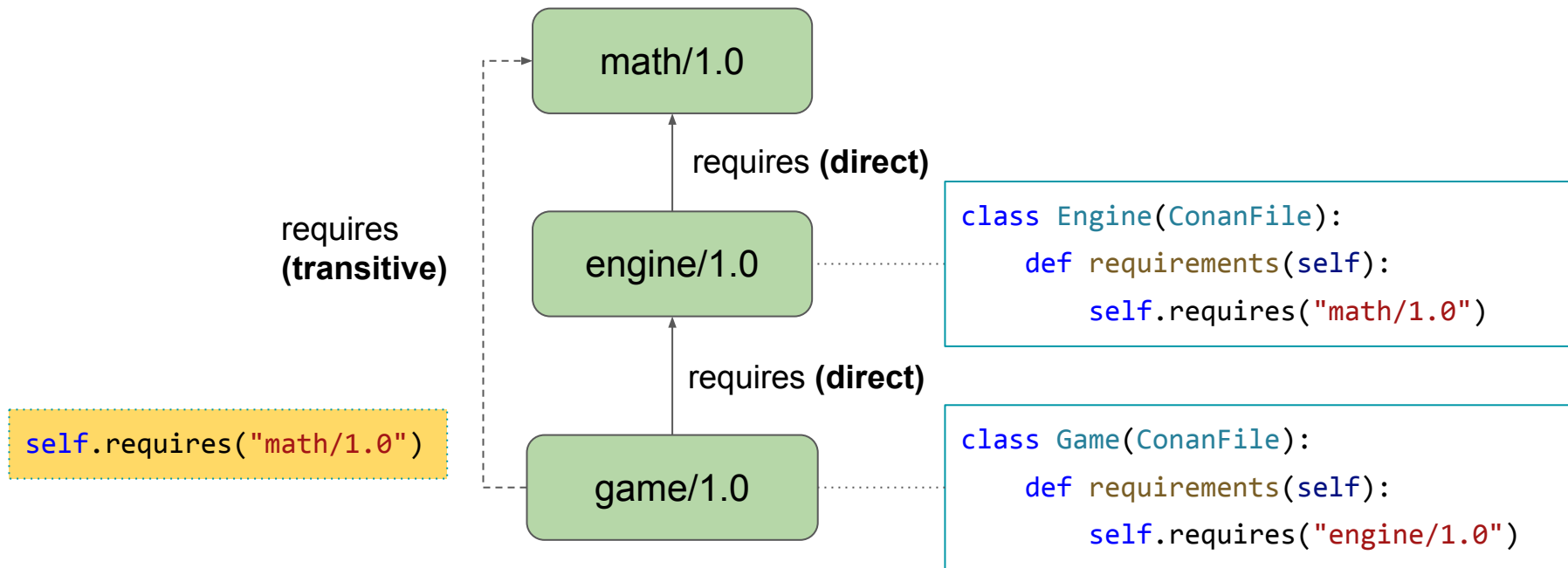
C/C++ Package Manager

Modeling of “build-time only” Dependencies

Inspired by how build systems model transitive usage requirements, we created a model in which “requires” can be qualified with several different “traits”, that define how packages are required



Support Direct vs. Transitive Dependencies



Dependency Graph

- Correct linkage requirements
- Correct header visibility
- Possible hidden/private dependencies
- and many more ([ACCU 2022](#))

Among different build systems!

Compatible “requires” syntax with 1.X



Conan 2.0 is way more efficient

Fetching only the binaries that it needs to build and run

- better `package_id` computation model)

Able to skip the download of large parts of the graphs binaries in many cases, accelerating install and build time.

- the new graph model
- the requirements traits and the new `package_type` recipe attribute

Will enable better modeling and management of dependencies.



CONAN

C/C++ Package Manager

2. Repeating yourself



C++ DevOps at Scale

So we are finally very excited to launch 2.0, with

- streamlined and more powerful lockfiles
- new improved CI oriented tools
 - Introducing extensions
 - Custom commands and Python API



CONAN

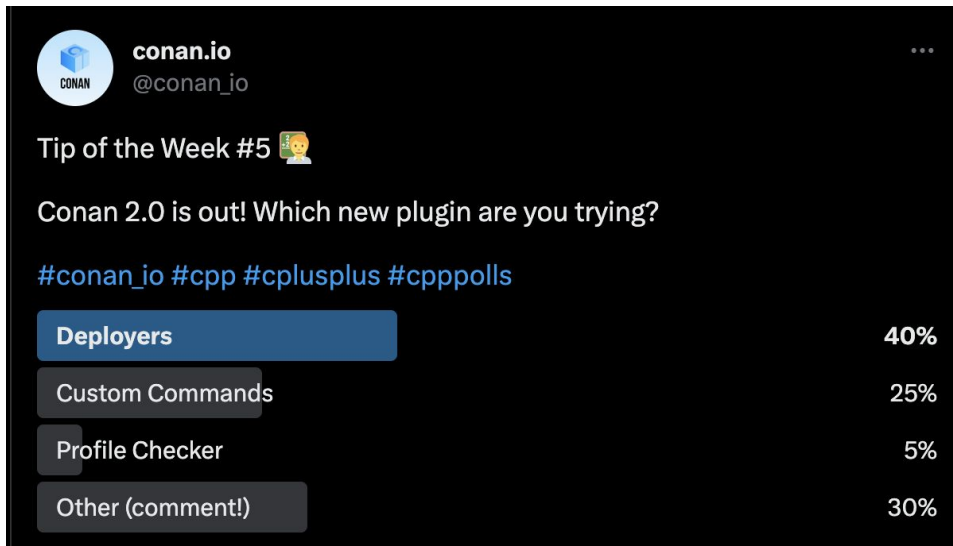
C/C++ Package Manager

Plugins

The solution - empower users to do it themselves!

Provide a framework for users to build solutions tailored to their needs with mechanisms that give them controlled management.

- Profile Checker
- Command Wrapper
- Package Signing



CONAN

C/C++ Package Manager

Deployers

- Flexible way to extract artifacts from cache
- Automate post-conan tasks
- Not in recipes, scale
- User customizable, “conan config install” installable

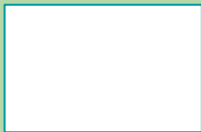


Compatibility Plugin

.conan2 (CONAN_HOME)

Built-in compatibility with
- Different cppstd

extensions/plugins/compatibility/



\$ conan config install
<url/git/path>

compatibility.py

```
def compatibility(conanfile):  
    ...
```

Python API and Custom Commands

- New Python API is released as “preview” no breaking changes
- Custom Commands built on top allow to tailor usage
- JSON output format to provide reliable stable output



CONAN

C/C++ Package Manager

Python API

Conan 1.X API

```
class API:
    def install(path, profile_path, ...)
    def create(path, profile, ...)
    def lock(path, profile, ...)
    def search( ...)
    def remove(...)
    def upload(patterns, remote)
```

Conan 2.0 API

```
class API:
    class RemoteAPI:
    class SearchAPI:
    class ProfileAPI:
        def get_profile(path, settings, options, conf)
    class GraphAPI:
        def graph(path, remotes, profiles, ...)
    class UploadAPI:
        def get_bundle(patterns, ..)
        def check_upstream(bundle, remote)
        def prepare(bundle)
        def check_integrity(bundle)
        def upload(bundle, remote)
```



CONAN

C/C++ Package Manager

Custom command

```
import json

from conan.api.output import ConanOutput
from conan.cli.command import conan_command

def output_json(msg):
    return json.dumps({"greet": msg})

@conan_command(group="My own commands", formatters={"json": output_json})
def hello(conan_api, parser, *args):
    """
    Simple command to print "Hello World!" line
    """
    msg = "Hello World!"
    ConanOutput().info(msg)
    return msg
```



CONAN

C/C++ Package Manager

3. Building a dam



C and C++ are trusted by enterprises worldwide

Thousands of teams are using Conan in their C and C++ workflows, ranging from startups to many large enterprises. We have made every effort to keep our promise of stability, continually delivering updates to the Conan 1.x series without breaking changes.

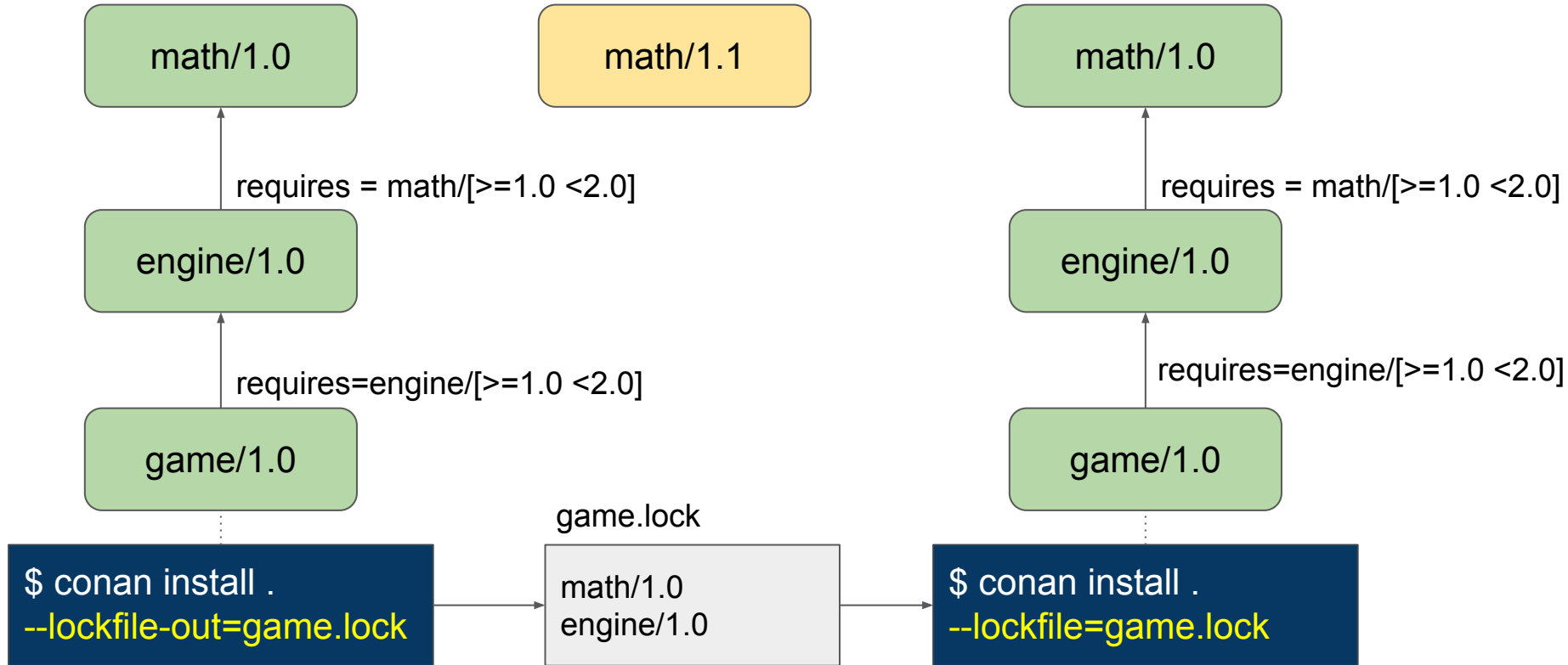
“Toss it over to diego” <3



CONAN

C/C++ Package Manager

Reproducible dependencies: Lockfiles



Lockfiles 2.0

- One lockfile for all configurations
- Easily mutable
- Easily understandable
- Fully strict and partial modes
- Easily mergeable
- Manual commands to modify (override)
- Possible to use multi-project
- Code in codebase 10x shorter
- **Game changer for CI at scale**



Welcome Enterprise DevOps for C and C++

- Enterprise scale can be high
- Enterprise/domain requirements can be challenging
- Continuous Integration at scale is critical
- Thinking beyond package and dependency management
 - Programming over time => SW engineering (T. Winters)
 - Dependency and Package management over time => DevOps



CONAN

C/C++ Package Manager

Conclusions



New graph

New plugin extensions

New deployers

New binary compatibility

Multi-revision cache

package_id

Lockfiles

New configuration and environment

Package immutability optimizations

... and many more

<https://docs.conan.io/2/whatsnew.html>

More Resources?

- ACCU Talk by Diego about the graph improvements:
<https://youtu.be/kKGglzm5ous>
- Introducing Conan 2.0 blog post:
<https://blog.conan.io/2023/02/22/Conan-2.0.html>
- New tutorial section <https://docs.conan.io/2/tutorial.html>
- Questions?
 - <https://github.com/conan-io/conan/issues>
 - <https://cppalliance.org/slack/> (community help)

Keep up to date! https://twitter.com/conan_io



CONAN

C/C++ Package Manager

Conclusion



`pip install conan`

<https://conan.io>



CONAN

C/C++ Package Manager