PYTHON PACKAGING

Where we are
Where we’re headed

@pradyunsg

gg.gg/pycon-in-2019
WHO?

Pradyun Gedam
@pradyunsg pradyunsg.me

Member of
Python Packaging Authority

Maintainer of
pip, virtualenv, packaging and more

PSF Fellow

College student!
WHERE WE ARE

An Overview of Tooling in the Python Packaging ecosystem
NOT HOW WE GOT HERE

https://www.youtube.com/watch?v=AQsZsgJ30AE
WHAT'S PACKAGING?
Medium for Moving Packages

Package

Thing

Package

Thing
SOFTWARE DISTRIBUTION
"Moving" Software

Distribution

Source Code

Distribution

Working Software
SOFTWARE DISTRIBUTION

AN UNSOLVED PROBLEM

Distribution

Source Code

"Moving" Software

Distribution

Working Software
Distribution
Source Code

"Moving" Software

Distribution
Working Software
Publisher

Distribution

"build"

Source Code
Publisher

- Build a Distribution from Source Code
- Build Environment
- Build Mechanisms
Publisher

Source Code

"build"

Distribution

"Moving" Software
Publisher

"upload"

Distribution

"build"

Source Code

"Moving" Software
Publisher

"upload"

Distribution

"build"

Source Code

"Moving" Software

- Upload the Distribution
- Upload Mechanisms
"Moving" Software

User
"Moving" Software

Distribution

User
Download the Distribution
Download Mechanisms
Choosing what to download
"Moving" Software

User

"download"

Distribution

Working Software
"Moving" Software

User

"download"

Distribution

"install"

Working Software
“Moving” Software Distribution

- Install from the Distribution
- Install Environment
- Install Mechanisms
Publisher

"upload"

Distribution

"build"

Source Code

"Moving" Software

User

"download"

Distribution

"install"

Working Software
Distribution

"Moving" Software

Publisher

"upload"

User

"download"

Distribution
DEALING WITH ENVIRONMENTS

- The Packaging Gradient
  https://sedimental.org/the_packaging_gradient.html
  https://www.youtube.com/watch?v=iLVNWfPWAC8

- Python Packaging Overview
  https://packaging.python.org/overview/
PyPA Tooling
Tools built by volunteers
Built on standards

(mostly)

https://packaging.python.org/specifications/
Package Distribution Metadata

Publisher

Distribution

"build"

Source Code

User

Distribution

"install"

Working Software
We went too far once.

- PEP 426 - Metadata 2.0
  - Was trying to solve all the problems with Python’s metadata. Never became a reality.

- Incremental Improvements
  - Our current, softer approach — workable approach with volunteers.
Publisher

- **Distribution**
- **Source Code**
  - build
  - backend

- Usually setuptools
- PEP 517 and PEP 518
  - "modern" source distributions
  - pyproject.toml
It’s not that straightforward.
**NON-PyPA Tooling**

Tools built by companies or different volunteers.
Environments are difficult.
Publisher

- Simplifies the "build" experience
- Metadata in pyproject.toml
"... all of this stuff is really hard and there’s a lot of knowledge scattered around in people’s heads that could really do with being captured somewhere"

— Paul Moore
WHERE WE’RE HEADED

Future Improvements in Python Packaging
I don’t have a time machine.
Core Functionality Improvements

ongoing

- securing PyPI — The Update Framework (TUF)
- newer manylinux standards — manylinux2014
- yanking packages from PyPI
- better licensing metadata
Core Functionality Improvements soon?

- structured lockfiles in pip
- package preview functionality from PyPI
- security notifications for vulnerable packages
- pip dependency resolver
Better User Experience

- better defaults in pip
- feature flags on PyPI
- interoperability testing
- professional UX review
Reducing Technical Debt

- virtualenv rewrite
- making the transition from distutils -> setuptools
- pip build logic refactor
- pip "working schemes" cleanup
- change the right environments
MOAR STANDARDISATION

• editable installations

• better, more powerful, extras

• less ambiguous licensing

• interoperability with other package managers
How do we get there?

- Volunteer!
- Tell us how you deal with "interesting" issues.
- Fund one of the targeted funded projects.