

RETHINKING PYTHON PACKAGING

A thought experiment

@pradyunsg

pradyunsg.me/talks

WHO?

Pradyun Gedam

@pradyunsg

pradyunsg.me

Member of
Python Packaging Authority

Maintainer of
pip, virtualenv, packaging and more

PSF Fellow

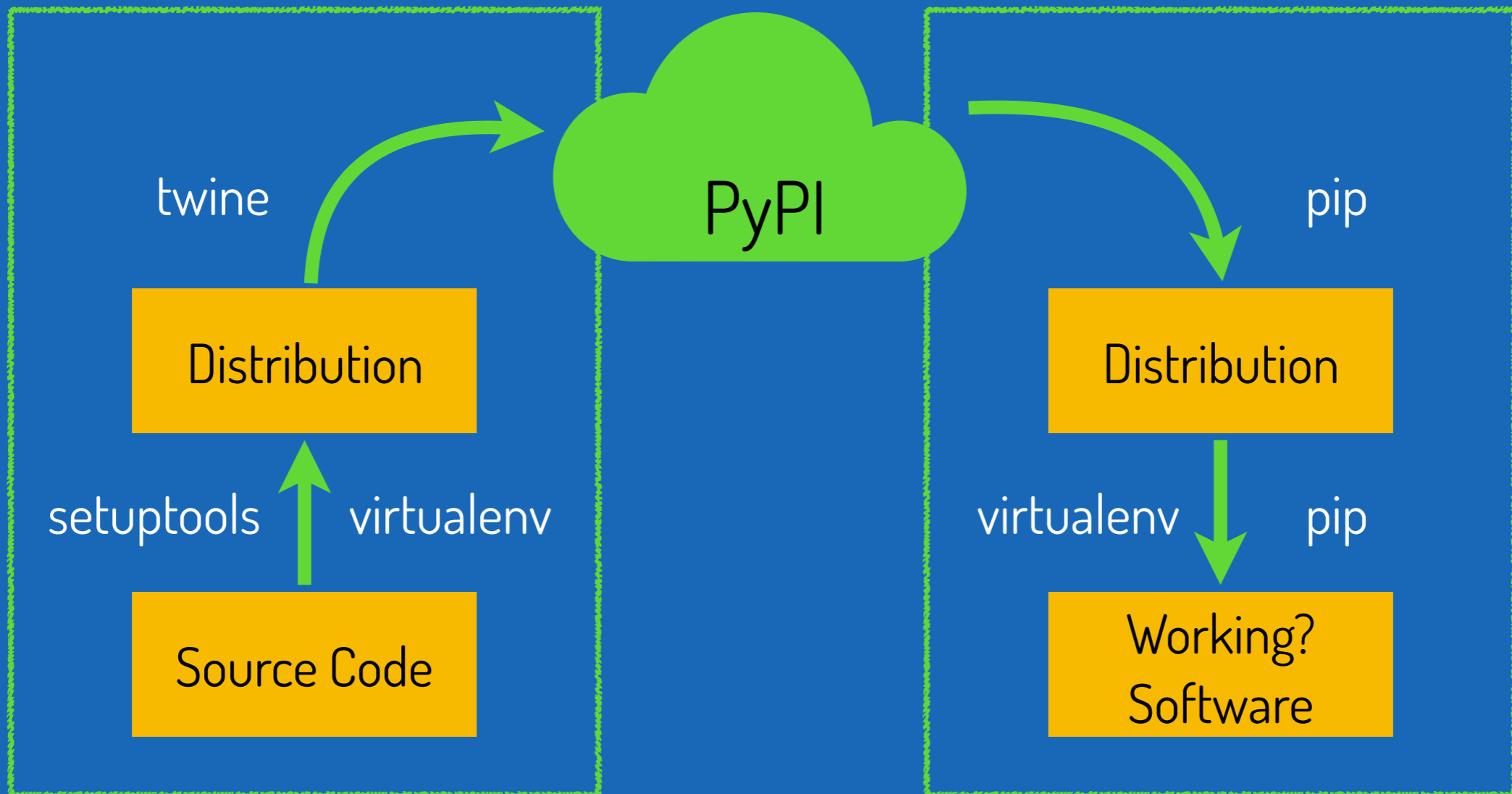
College student!

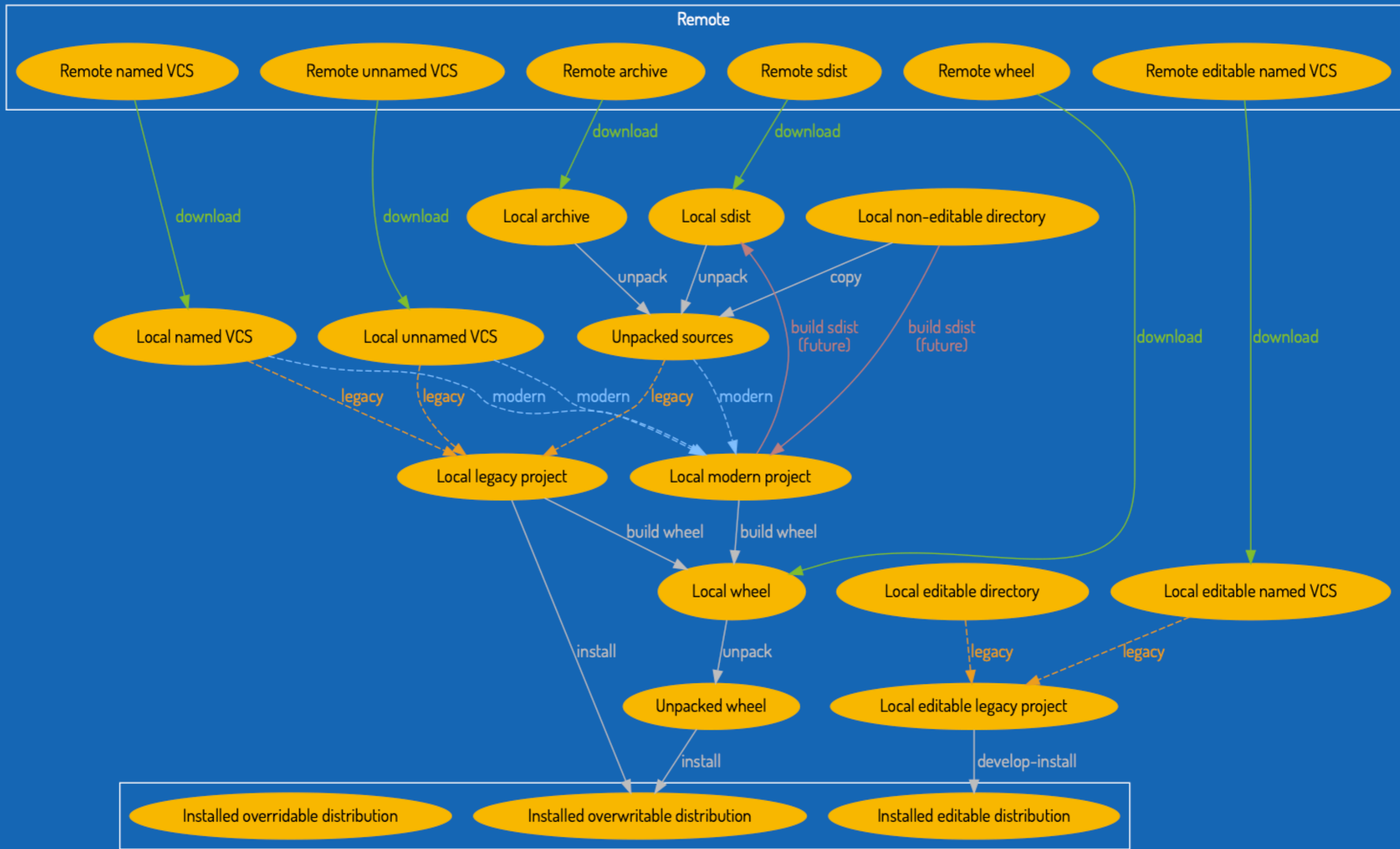


RETHINKING PYTHON PACKAGING

Publisher

User





RETHINKING PYTHON PACKAGING

Free to ignore existing tooling

Best practices as defaults

Try to not break everything

Easier to understand, use and
maintain

CONSTRAINTS

- No removal of functionality for:
 - infrastructure — PyPI
 - published packages
 - existing PyPA standards

DISCLAIMERS

complete speculation
(sort of)

other folks' ideas + my opinions

very little UX experience

"THINGS"

Source Tree

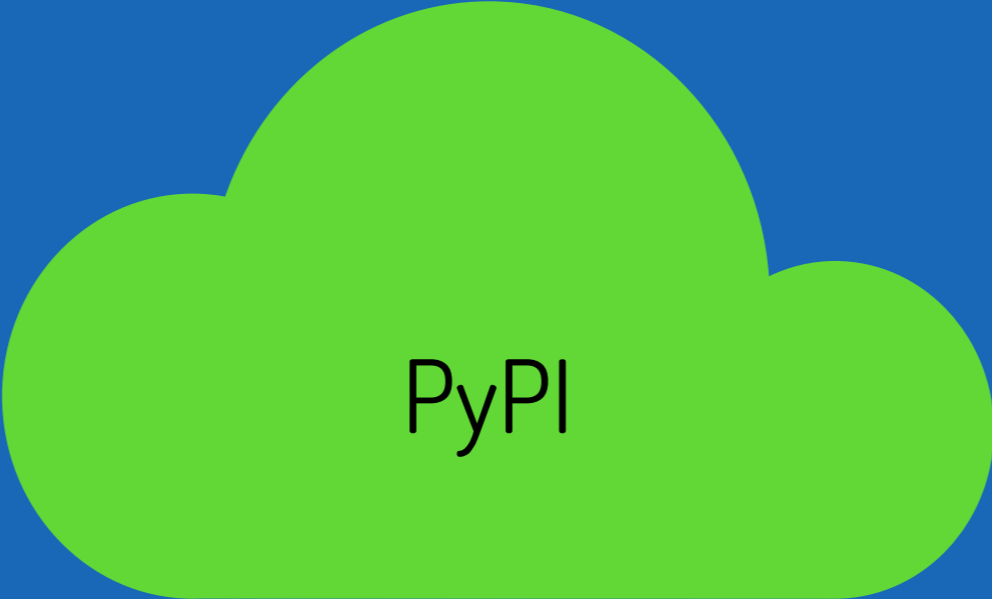
Source Distribution

Wheel Distribution

Environments

Installed Packages

Platform Details



PEP 517'S MODEL

PEP 517'S MODEL

PEP 517 -- A build-system independent format for source trees

MODERN PACKAGING TOOLING'S MODEL

MODERN PACKAGING TOOLING'S MODEL

- Build Backend — like setuptools, flit etc
 - Produce distributions from source trees
- Build Frontend — like pip
 - is user-facing — user workflows
 - manages environment
 - handles distributions
 - orchestrates build backends

BACKEND OPERATIONS

- Produce distributions from source trees
 - Handling of Python code
 - Handling of non-Python code / artefacts
 - Metadata generation
 - Properly place files into a distribution

METADATA

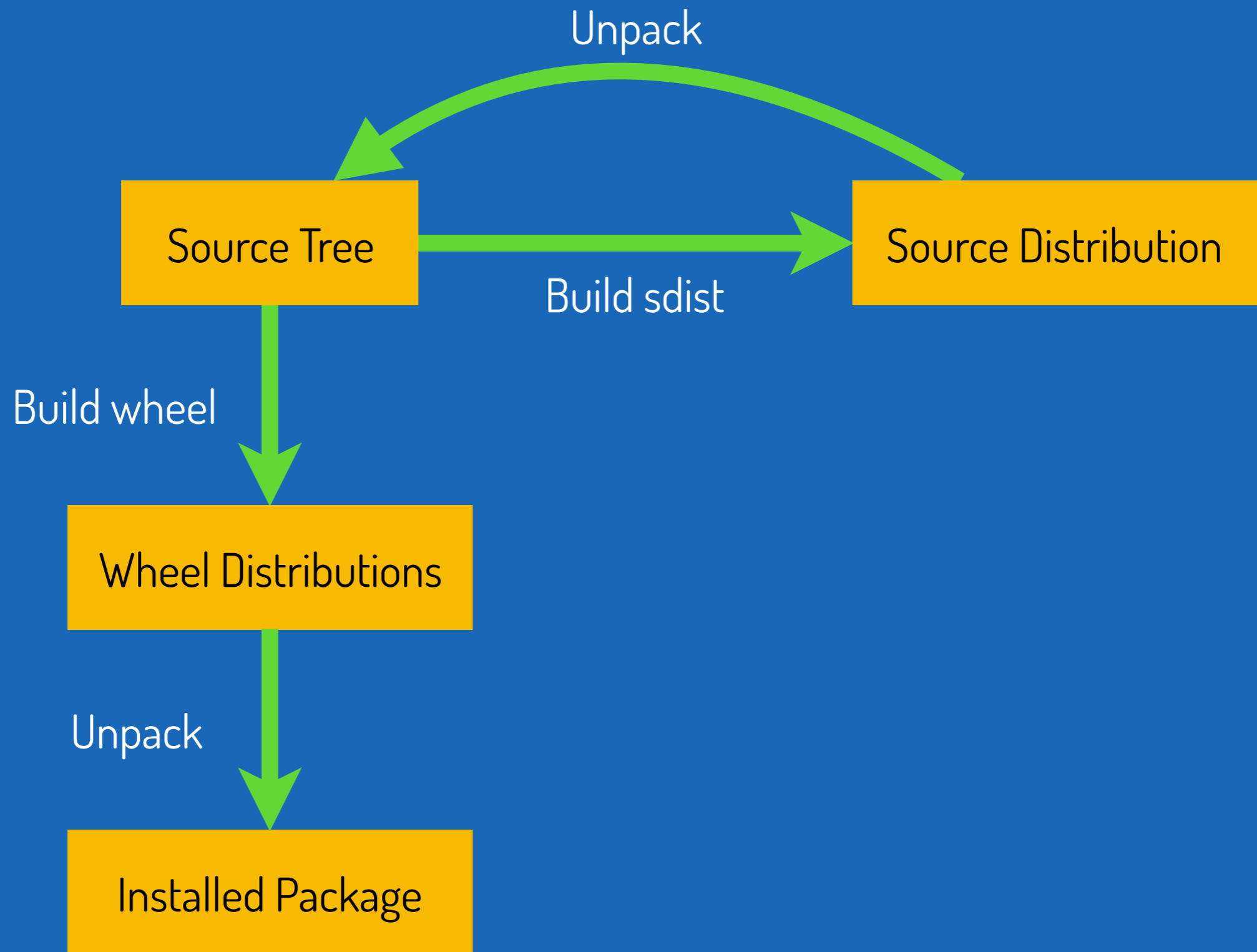
- setuptools allows for arbitrary logic, which can be used for generating metadata
- What we'd want
 - Static metadata, by default
 - Dynamic metadata, by opt-in
- Keep metadata in pyproject.toml
 - Allow specifying a script for enhancing it

HANDLING OF NON-PYTHON CODE / ARTEFACTS

- setuptools invokes the compilers directly
 - complicated build process → complicated setup.py
- What we'd want
 - Generate build instructions for *other* build tools
 - Invoke the build tools

PROPERLY PLACE FILES INTO A DISTRIBUTION

- Idempotent source distributions
 - unpack sdist + build sdist → same as initial
- Build non-Python "stuff" only for wheels



BUILD SDIST

- Load information from `pyproject.toml`
- Generate metadata
- Package relevant files into a `.tar.gz` file

BUILD WHEEL

- Load information from `pyproject.toml`
- Build non-Python code into binaries
- Generate metadata
- Package relevant files into a `.whl` file

FRONTEND OPERATIONS

- Environment management
 - Dependency resolution (hah!)
 - Installation / Uninstallation / Upgrade
- Handling distributions
 - Discovering and fetching distributions
 - Unpacking distributions correctly
- Orchestrates build backends
 - Manage source tree → distribution

HANDLING DISTRIBUTIONS

- Discovering and fetching distributions
 - Index interaction
 - Networking — TLS, proxies, caching and more
- Unpacking distributions
 - Placing the contents of a wheel correctly

ENVIRONMENT MANAGEMENT

- Allow installation and uninstallation of projects into an environment.
 - Binary Compatibility — wheel compatibility tags
 - Reproducibility — lockfile
- Ensure all requirements are satisfied for all installed packages
 - Dependency Resolution

FRONT END USABILITY

- Produce a local directory of artifacts for offline usage
 - downloading from PyPI
 - producing wheels
- Introspect the current state of installed packages, and get more information about them

UPLOADING DISTRIBUTIONS

- Two Phase uploads — enable new workflows
 - <https://github.com/pypa/warehouse/issues/726>
 - Easier communication for deprecations
 - Stricter metadata requirements for new uploads
 - Better licensing information
- Yanking releases
 - Yank due to security bugs

**OK, GREAT.
NOW WHAT?**

WE CHART A COURSE TO GET THERE.

FURTHER READING

- <https://discuss.python.org/t/876>
- <https://discuss.python.org/t/2062>
- <https://discuss.python.org/t/2570>
- <https://discuss.python.org/t/2579>
- <https://discuss.python.org/t/2584>
- <https://discuss.python.org/t/2587>
- <https://discuss.python.org/t/2591>
- <https://www.curiousinefficiency.org/posts/2016/09/python-packaging-ecosystem.html>

THANKS!

@pradyunsg

pradyunsg.me/talks