

PyShip

Easily Ship Python Desktop Apps to End Users

James Abel

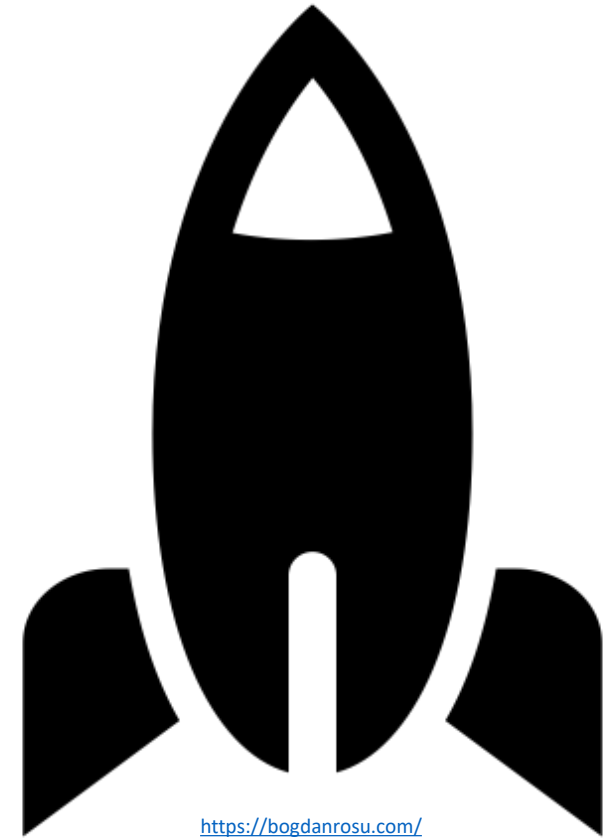
PyCon US 2023

www.pyship.com

www.abel.co

@jamesabel

<https://github.com/jamesabel>



Requirement:

Deliver Python Desktop Applications to End Users

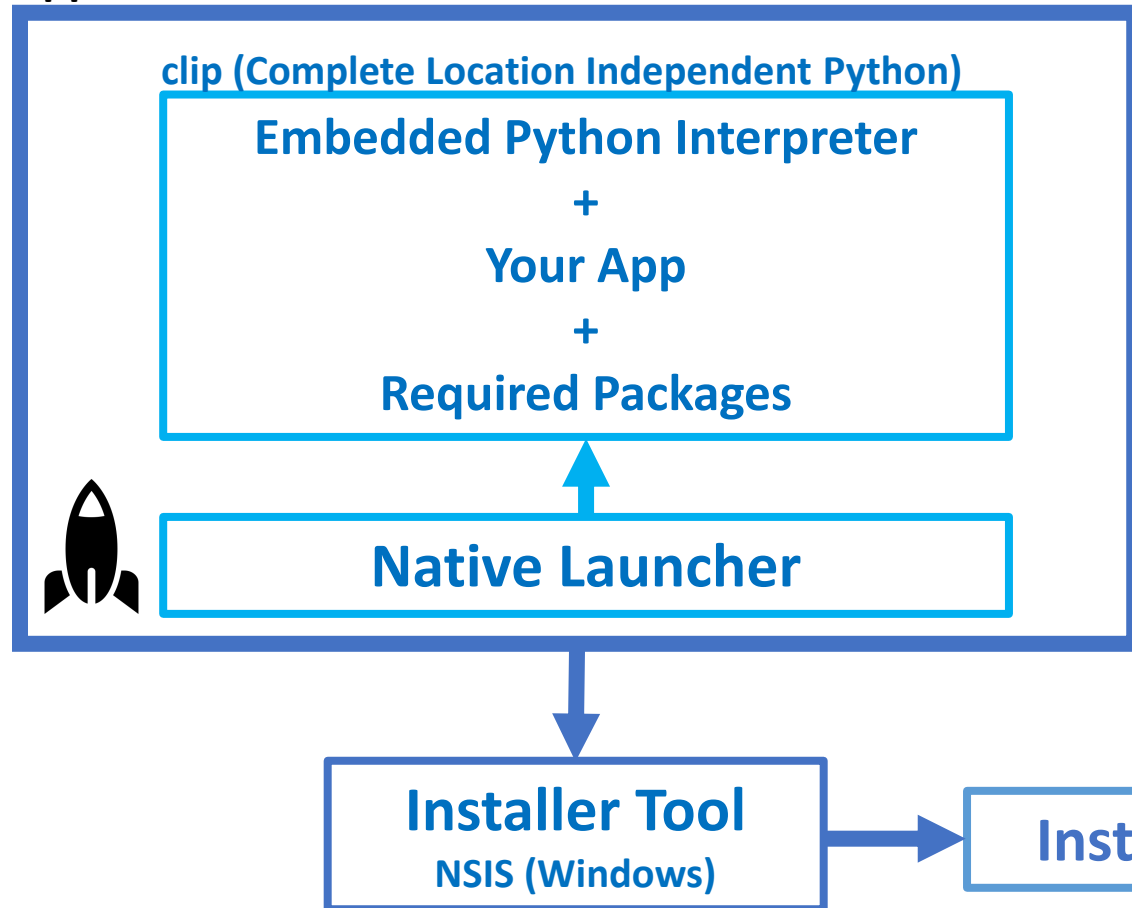
- Freeze practically any Python application
 - Most often GUI applications, but also works for CLI
- Create an Installer to give to end users
- "Run on Startup" option
- Native executable (e.g. .exe for Windows that user can click on)
- Application can auto-update (no end user intervention required)
- Windows
- Futures
 - App store aware/signed
 - MacOS, popular Linux distros (e.g., Ubuntu, Fedora)

How to use pyship

1. `pip install pyship`
2. Create your "main module" application
 - Runnable via `python -m <app>`
3. Package your app
 - Installable with `pip`
4. Put `pyship` specific options in `pyproject.toml`
 - `is_gui = true` (default is false)
 - `run_on_startup = true` (default is false)
5. `python -m pyship`
 - Creates application launcher (e.g., .exe for Windows)
 - Creates `clip` (Complete Location-Independent Python in a folder)
 - Creates installer (using NSIS) for end users

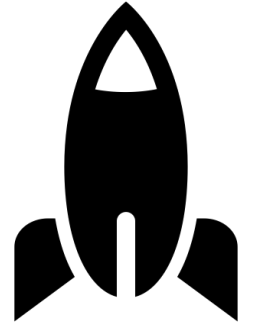
PyShip High Level Architecture

Application



1. Create your Python application runnable as a “main module” (e.g. `python -m <app>`)
2. Package your application as a distribution (e.g., wheel)
3. Add a few fields to `pyproject.toml`
4. Run `pyship` to create installer and clip
5. Release installer to users
6. Update as desired

Thank You



- Thomas Kluyver ([pynsist](#) , [flit](#))
- Glyph (presentations such as [Shipping Software To Users With Python](#) , discussions)
- Steve Dower (Microsoft, embedded Python)

BACKUP

- Related Tools

- [cx freeze](#)
- [py2exe](#)
- [py2app](#)
- [briefcase](#)
- [pyinstaller](#)
- [pynsist](#)