PYTHON PYPROJECT.TOML + VERSIONING + PUBLISHING

AGENDA

- (Part 0: Migrating to `pyproject.toml`)
- Part 1: Dealing with Python package versions
  - ❌ We have multiple places for the version!
- Part 2: Automating the busywork
  - Publishing the package on GitHub
  - Publishing the package on PyPI

💡 Consistency 💡 Less manual work

0: MIGRATING TO PYPROJECT.TOML

- Mostly straight-forward
    - e.g. just filling in the fields of the file
- setuptools' `where`
- ❌ Dealing with the package version in `ocrd-tool.json` → Motivation for setuptools-ocrd

1: DEALING WITH VERSIONS – THE CHALLENGE

We have multiple sources for the program version:

- Python package (`pyproject.toml`)
- `ocrd-tool.json`

  → Part 1A
- git tag

  → Part 1B

1A: SOURCING THE VERSION FROM `ocrd-tool.json`

- Before `pyproject.toml`, we programatically read the version in `setup.py`
- ❌ Can't do that anymore with `pyproject.toml`
- ✅ We now have setuptools-ocrd
    - setuptools plugin
    - reads version from `ocrd-tool.json`
    - makes sure `ocrd-tool.json` is in the sdist

HOW TO USE SETUPTOOLS-OCRD

- Include as part of the `build-system` in `pyproject.toml`:

```
[build-system]
requires = ["setuptools>=61.0.0", "wheel", "setuptools-ocrd"]

[project]
...
#version = "1.2.3" ← Remove this line
dynamic = ["version", ...]  # Make it dynamic
```

- Building the Python package (e.g. `python -m build`) should now produce a package (and sdist) with the correct version!

1B: PYTHON PACKAGE VERSION VS GIT TAG

- ❌ Can't source the package version from the git tag, because we need it in `ocrd-tool.json`
- 💡 But we can: Check git tag on tag push in CI

GitHub Action workflow `release.yml` (shortened!):

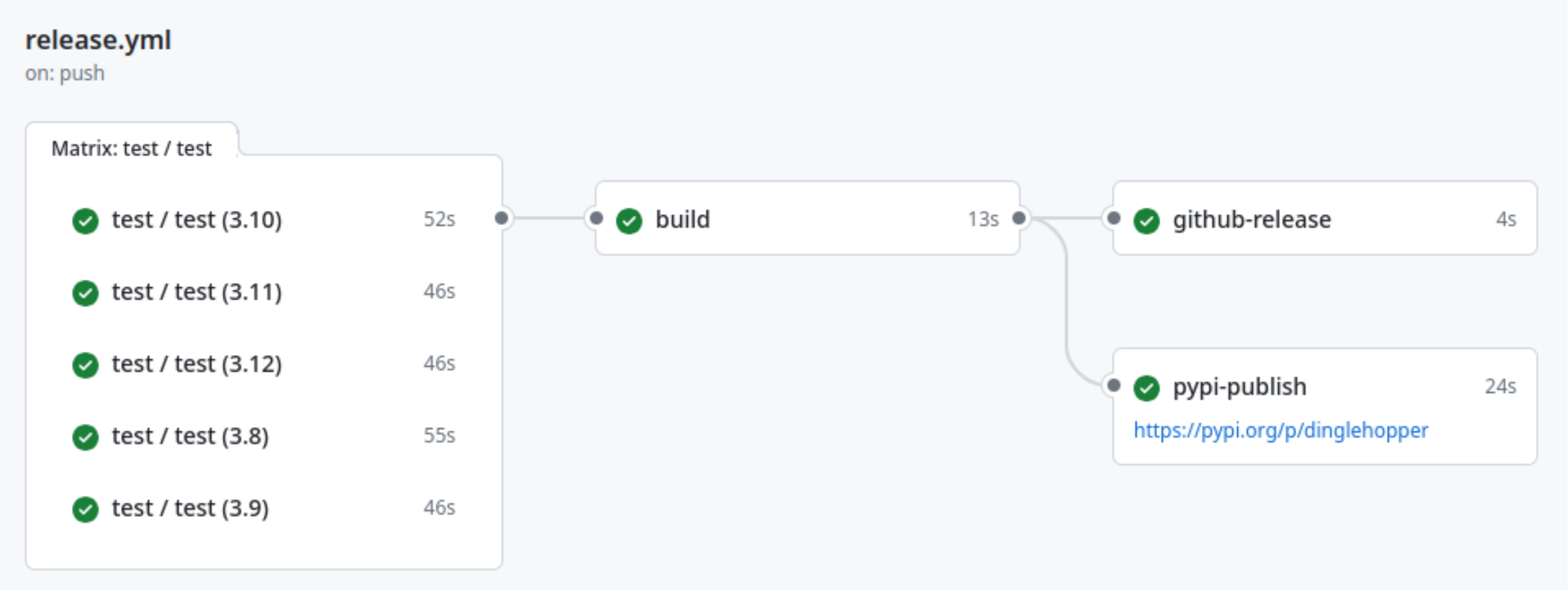
```yaml
on:
  push:
    tags:
      - "v*.*.*"

jobs:
  # [...]
  build:
    needs: test
    runs-on: ubuntu-latest
    steps:
      # [...]
      - name: Check git tag vs package version
        run: .github/workflows/release-check-version-tag
```

2: AUTOMATING THE BUSYWORK

- Goal: Have a consistent upload of
  - git tag
  - GitHub release
  - PyPI release
- 💡 Trigger GitHub + PyPI releases by git tag

GITHUB ACTIONS WORKFLOW



release.yml
on: push

Matrix: test / test

✓ test / test (3.10) 52s

✓ test / test (3.11) 46s

✓ test / test (3.12) 46s

✓ test / test (3.8) 55s

✓ test / test (3.9) 46s

✓ build 13s

✓ github-release 4s

✓ pypi-publish 24s
https://pypi.org/p/dinglehopper

- The following YAML snippets are shortened!
- Full example in the dinglehopper project
- This should be possible to do with CircleCI, too

TRIGGER ON GIT TAG PUSH

```
name: release

on:
  push:
    tags:
      - "v*.*.*"

# [continued]
```

BUILD PYTHON PACKAGE

```yaml
jobs:
  build:
      # [... After check from part 1 ...]

      - name: Build package
        run: |
          python3 -m pip install --upgrade build
          python3 -m build

      - name: Upload dist
        uses: actions/upload-artifact@v4
        with:
          name: dist
          path: dist/
```

CREATE A GITHUB RELEASE (INCL. FILES)

```yaml
github-release:

  steps:

    - name: Download dist
      uses: actions/download-artifact@v4
      with: { name: dist, path: dist/ }

    - name: Create release on GitHub
      uses: softprops/action-gh-release@v1
      with:
        files: dist/*
```

(Uses GitHub's implicit credentials.)

CREATE A PYPI RELEASE

```
pypi-publish:
  environment:
    name: pypi
    url: ${{ env.PYPI_URL }}
  permissions:
    id-token: write

  steps:
    - name: Download dist
      uses: actions/download-artifact@v4
      with: { name: dist, path: dist/ }
    - name: Publish package distributions to PyPI
      uses: pypa/gh-action-pypi-publish@release/v1
```

(Uses PyPI's trusted publishing.)

FUTURE WORK?

- Unfortunate that `ocrd-tool.json` requires a version
  - no single-sourcing from git!
- `.github/workflows/release-check-version-tag` could be a reusable GitHub Action

Probably not:

- It's good that the above release workflow is composed of different steps
  - Don't combine into a GitHub Action to retain flexibility
  - Copying the YAML is good enough
- CircleCI

QUESTIONS?