Trusted Publishing & Digital Attestations in the OSS Ecosystem

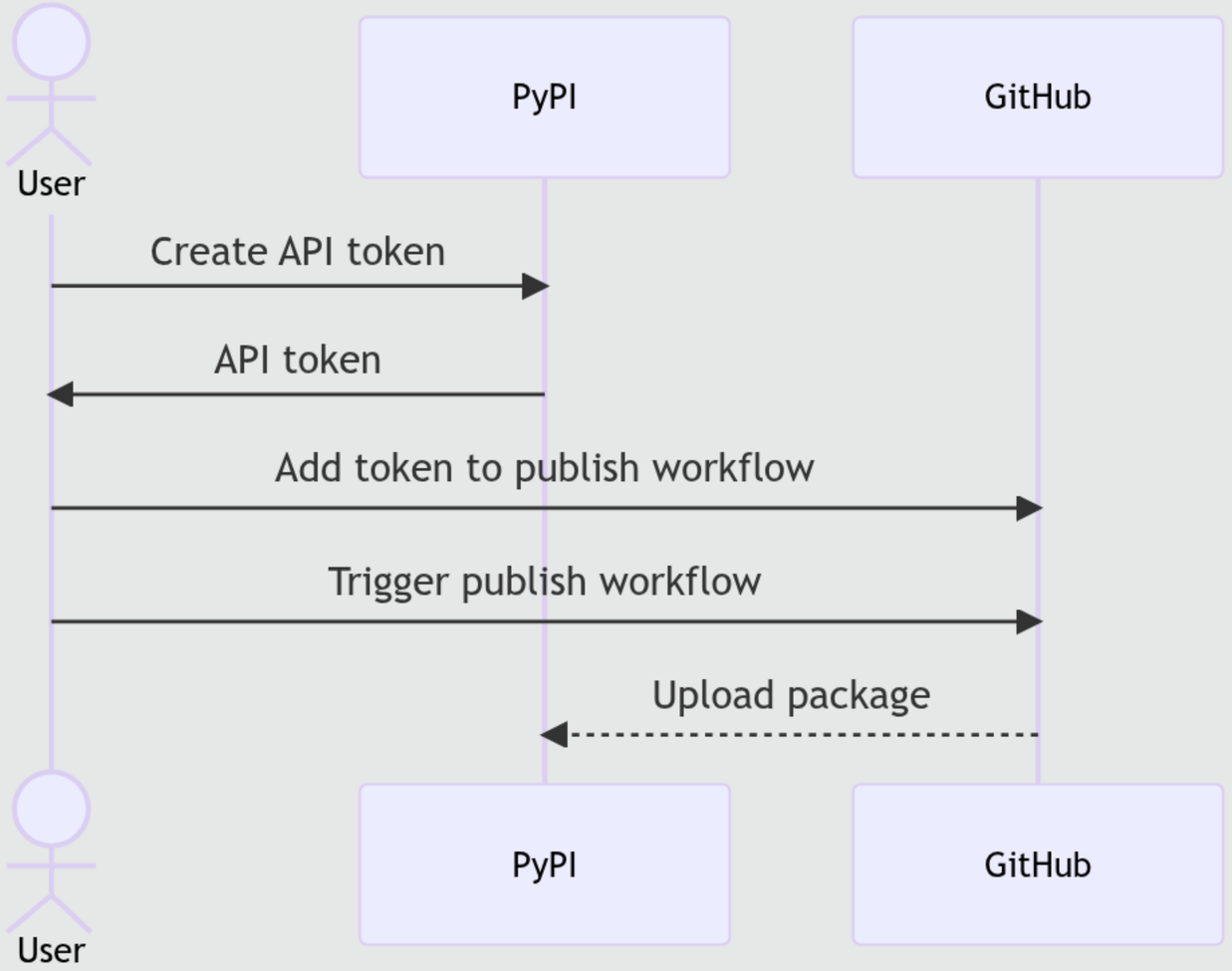
$ whoami

Jordan Duabe

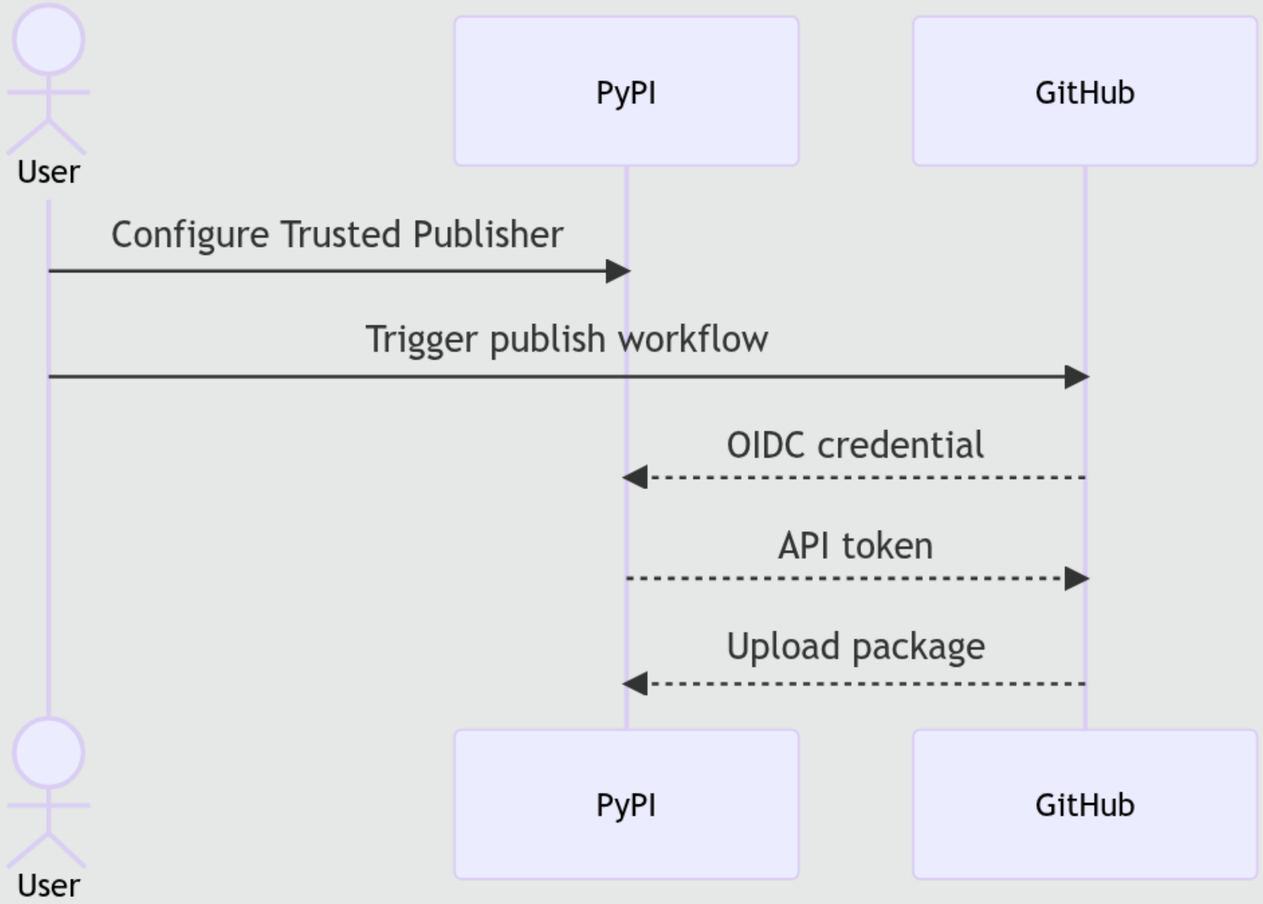
- Software Engineer
- Mechanical keyboard enthusiast
- *jordan@jduabe.dev*
- *@j4ckofalltrades@infosec.exchange*

Trusted Publishing

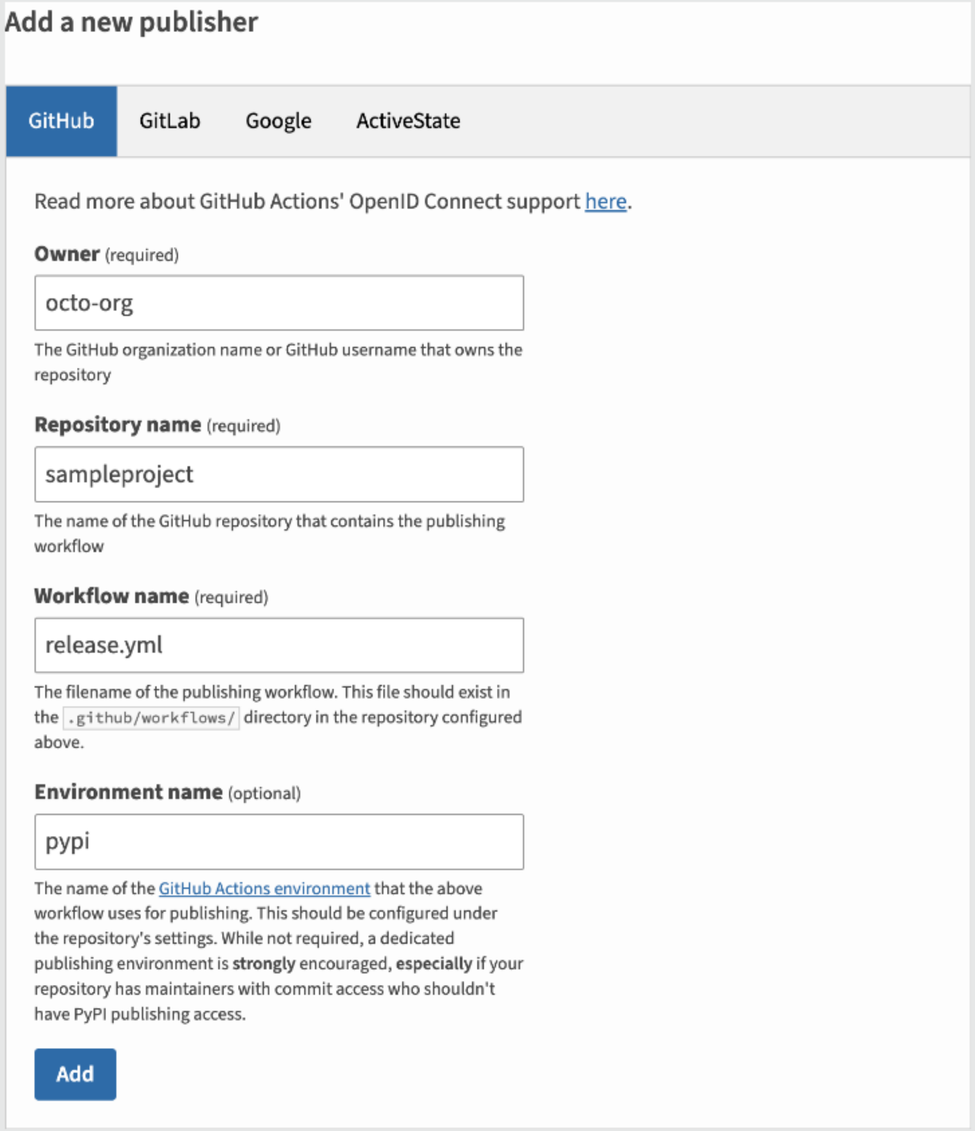
API Token publish flow



Trusted Publishing flow



Configure a Trusted Publisher in PyPI



Source: https://docs.pypi.org/trusted-publishers/adding-a-publisher/

Publishing with a Trusted Publisher

```
# https://github.com/pypa/gh-action-pypi-publish
jobs:
  pypi-publish:
    name: upload release to PyPI
    runs-on: ubuntu-latest
+    # Specifying a GitHub environment is optional, but strongly encouraged
+    environment: pypi
+    permissions:
+      # IMPORTANT: this permission is mandatory for Trusted Publishing
+      id-token: write
    steps:
      # retrieve your distributions here

      - name: Publish package distributions to PyPI
        uses: pypa/gh-action-pypi-publish@v1.11.0
-       with:
-         username: __token__
-         password: ${{ secrets.PYPI_TOKEN }}
```

Source: https://docs.pypi.org/trusted-publishers/using-a-publisher/

Benefits over long-lived API tokens

- Can be accidentally exposed in CI logs or configuration files
- Require manual rotation and management
- If compromised, they provide persistent access until revoked
- Oftentimes do not have granularity over permissions

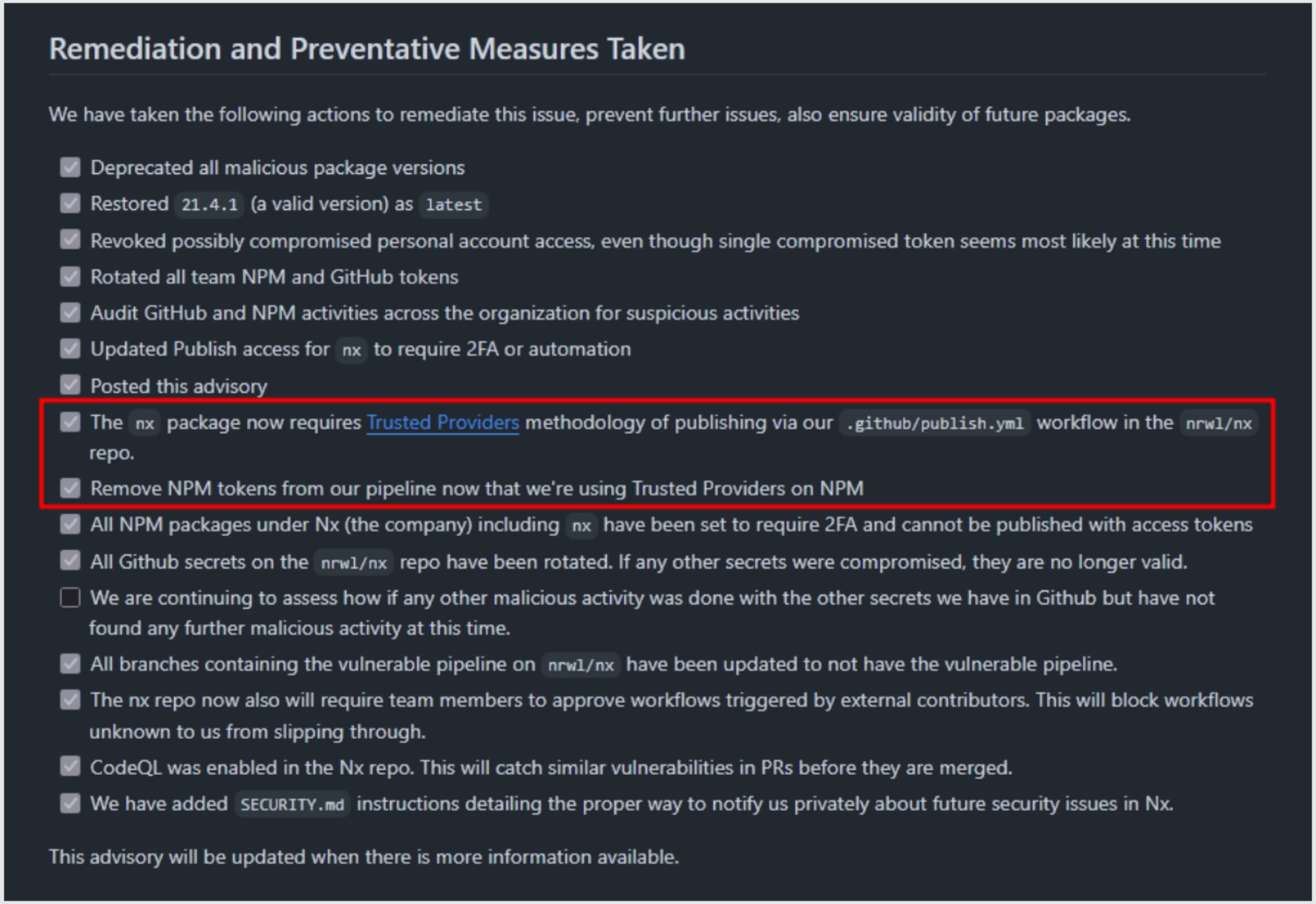
Benefits over long-lived API tokens

- Can be accidentally exposed in CI logs or configuration files
- Require manual rotation and management
- If compromised, they provide persistent access until revoked
- Oftentimes do not have granularity over permissions

Adoption in the OSS ecosystem

- Ruby's RubyGems
- Dart's Pub Package Registry
- Rust's Crates
- JavaScript's npm and JSR

Supply Chain Attack on Nx build system package



Remediation and Preventative Measures Taken

We have taken the following actions to remediate this issue, prevent further issues, also ensure validity of future packages.

- [x] Deprecated all malicious package versions
- [x] Restored `21.4.1` (a valid version) as `latest`
- [x] Revoked possibly compromised personal account access, even though single compromised token seems most likely at this time
- [x] Rotated all team NPM and GitHub tokens
- [x] Audit GitHub and NPM activities across the organization for suspicious activities
- [x] Updated Publish access for `nx` to require 2FA or automation
- [x] Posted this advisory
- [x] The `nx` package now requires Trusted Providers methodology of publishing via our `.github/publish.yml` workflow in the `nrwl/nx` repo.
- [x] Remove NPM tokens from our pipeline now that we're using Trusted Providers on NPM
- [x] All NPM packages under Nx (the company) including `nx` have been set to require 2FA and cannot be published with access tokens
- [x] All Github secrets on the `nrwl/nx` repo have been rotated. If any other secrets were compromised, they are no longer valid.
- [ ] We are continuing to assess how if any other malicious activity was done with the other secrets we have in Github but have not found any further malicious activity at this time.
- [x] All branches containing the vulnerable pipeline on `nrwl/nx` have been updated to not have the vulnerable pipeline.
- [x] The nx repo now also will require team members to approve workflows triggered by external contributors. This will block workflows unknown to us from slipping through.
- [x] CodeQL was enabled in the Nx repo. This will catch similar vulnerabilities in PRs before they are merged.
- [x] We have added `SECURITY.md` instructions detailing the proper way to notify us privately about future security issues in Nx.

This advisory will be updated when there is more information available.

Source: https://github.com/nrwl/nx/security/advisories/GHSA-cxm3-wv7p-598c

Digital Attestations

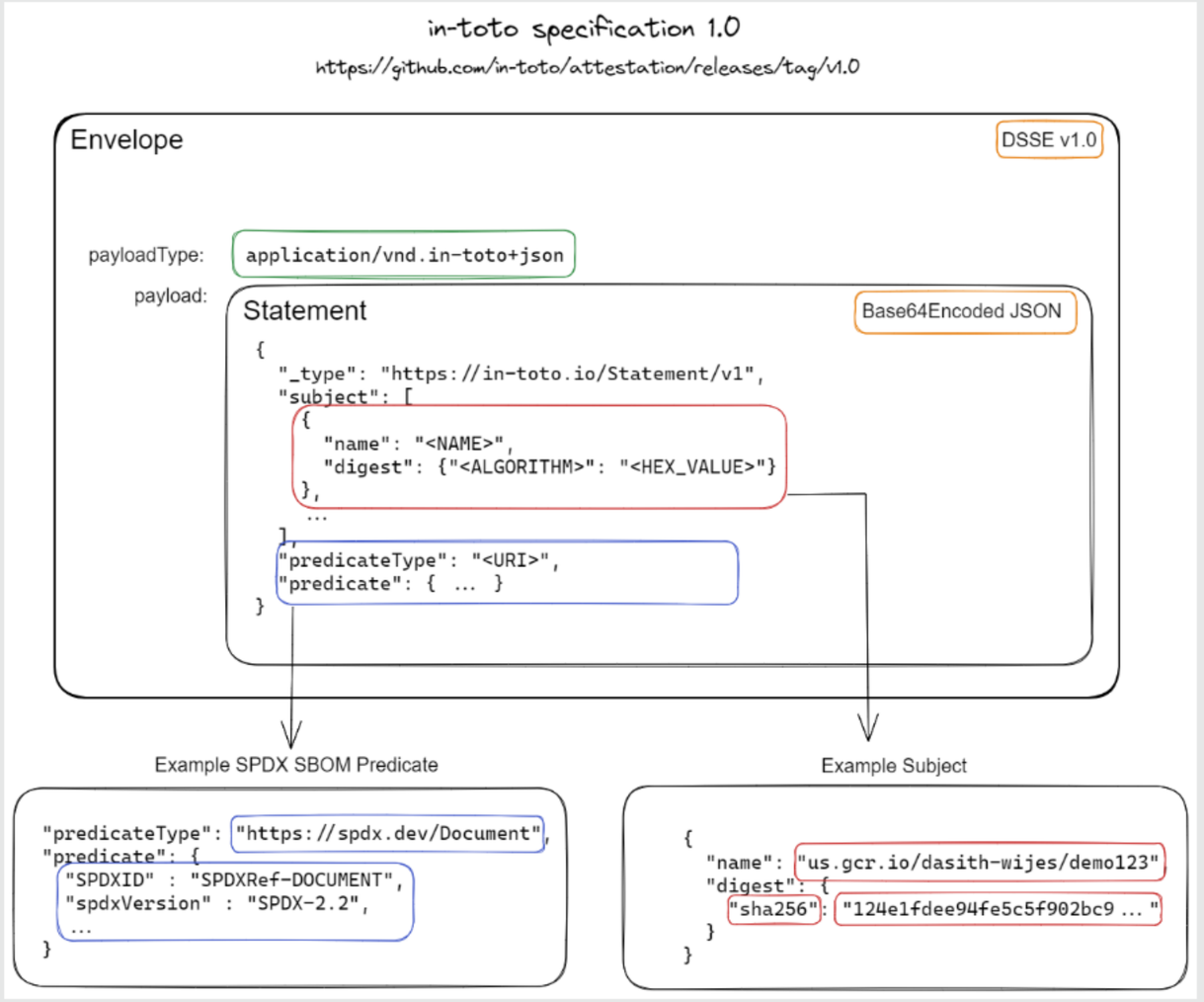
PEP 740 – Index support for digital attestations

PyPI uses the in-toto Attestation Framework for the attestations it accepts.

- SLSA Provenance (*build provenance*)
- PyPI Publish (*publish provenance*)

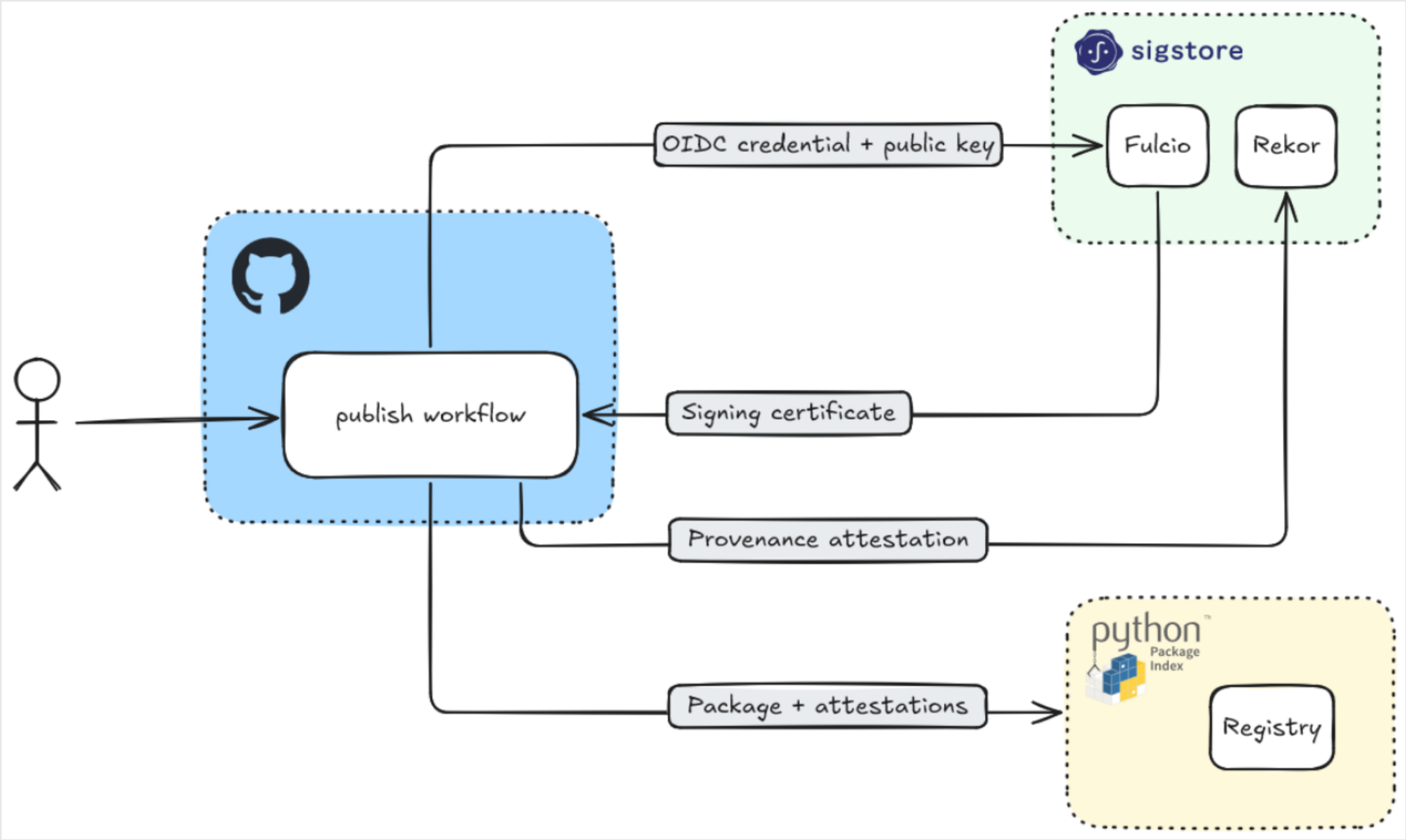
Source: https://trailofbits.github.io/are-we-pep740-yet/

in-toto Attestation Framework Spec



Source: https://github.com/in-toto/attestation/blob/main/spec/README.md

Trusted Publishing with Sigstore



Generating and uploading attestations

```yaml
# https://github.com/pypa/gh-action-pypi-publish
# Attestations enabled by default as of v.1.11.0 and newer
jobs:
  pypi-publish:
    name: upload release to PyPI
    runs-on: ubuntu-latest
    environment: pypi
    permissions:
      id-token: write
    steps:
      # retrieve your distributions here

      - name: Publish package distributions to PyPI
        uses: pypa/gh-action-pypi-publish@v1.11.0
```

Source: https://docs.pypi.org/trusted-publishers/using-a-publisher/

Provenance in PyPI

Provenance

The following attestation bundles were made for `powerline_k8s-1.5.4.tar.gz`:

Publisher: `pypi-publish.yml` on j4ckofalltrades/powerline-k8s

Attestations:

Values shown here reflect the state when the release was signed and may no longer be current.

- Statement:
  - Statement type: `https://in-toto.io/Statement/v1`
  - Predicate type: `https://docs.pypi.org/attestations/publish/v1`
  - Subject name: `powerline_k8s-1.5.4.tar.gz`
  - Subject digest: `4c491b88f3cb872720e38f5890565f6f1c93f228a51f52174ee486a30417807d`
  - Sigstore transparency entry: 245631303
  - Sigstore integration time: Jun 21, 2025, 7:22:04 PM

Source repository:

- Permalink: `j4ckofalltrades/powerline-k8s@8ee093f25d3d6515db6d28d41d8e4a935f2845c1`
- Branch / Tag: `refs/tags/v1.5.4`
- Owner: https://github.com/j4ckofalltrades
- Access: `public`

Publication detail:

- Token Issuer: `https://token.actions.githubusercontent.com`
- Runner Environment: `github-hosted`
- Publication workflow: `pypi-publish.yml@8ee093f25d3d6515db6d28d41d8e4a935f2845c1`
- Trigger Event: `push`

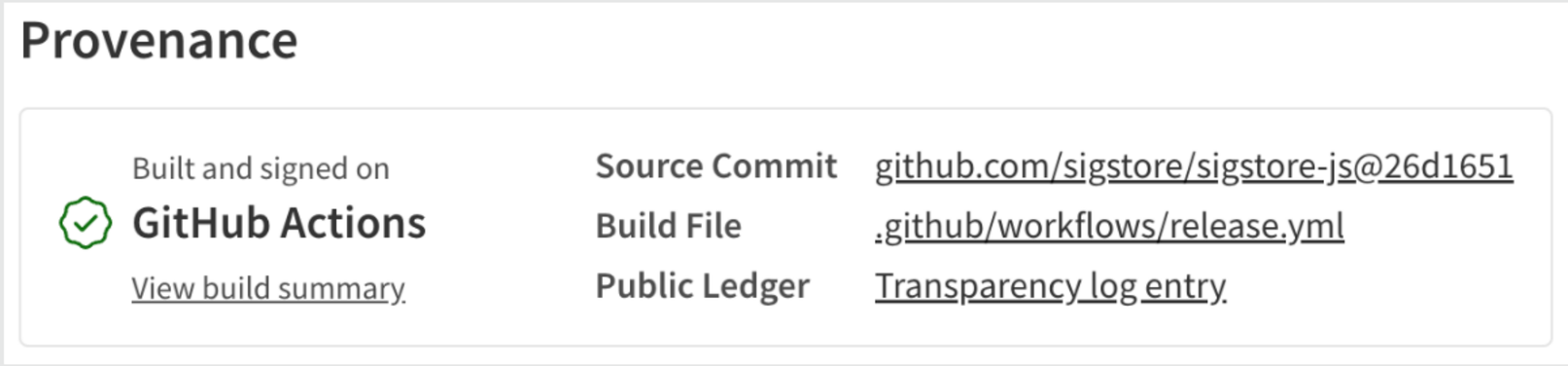
Source: https://pypi.org/project/powerline-k8s/#powerline_k8s-1.5.4.tar.gz

Verifying provenance with PyPI's Integrity API

```
# GET /integrity/<project>/<version>/<filename>/provenance
$ http get https://pypi.org/integrity/powerline-k8s/1.5.4/powerline_k8s-1.5.4.tar.gz/provenance \
> | jq -r '.attestation_bundles[0].attestations[0].envelope.statement' \
> | base64 -d \
> | jq

{
  "_type": "https://in-toto.io/Statement/v1",
  "subject": [
    {
      "name": "powerline_k8s-1.5.4.tar.gz",
      "digest": {
        "sha256": "4c491b88f3cb872720e38f5890565f6f1c93f228a51f52174ee486a30417807d"
      }
    }
  ],
  "predicateType": "https://docs.pypi.org/attestations/publish/v1",
  "predicate": null
}
```

Adoption in the OSS ecosystem: npm



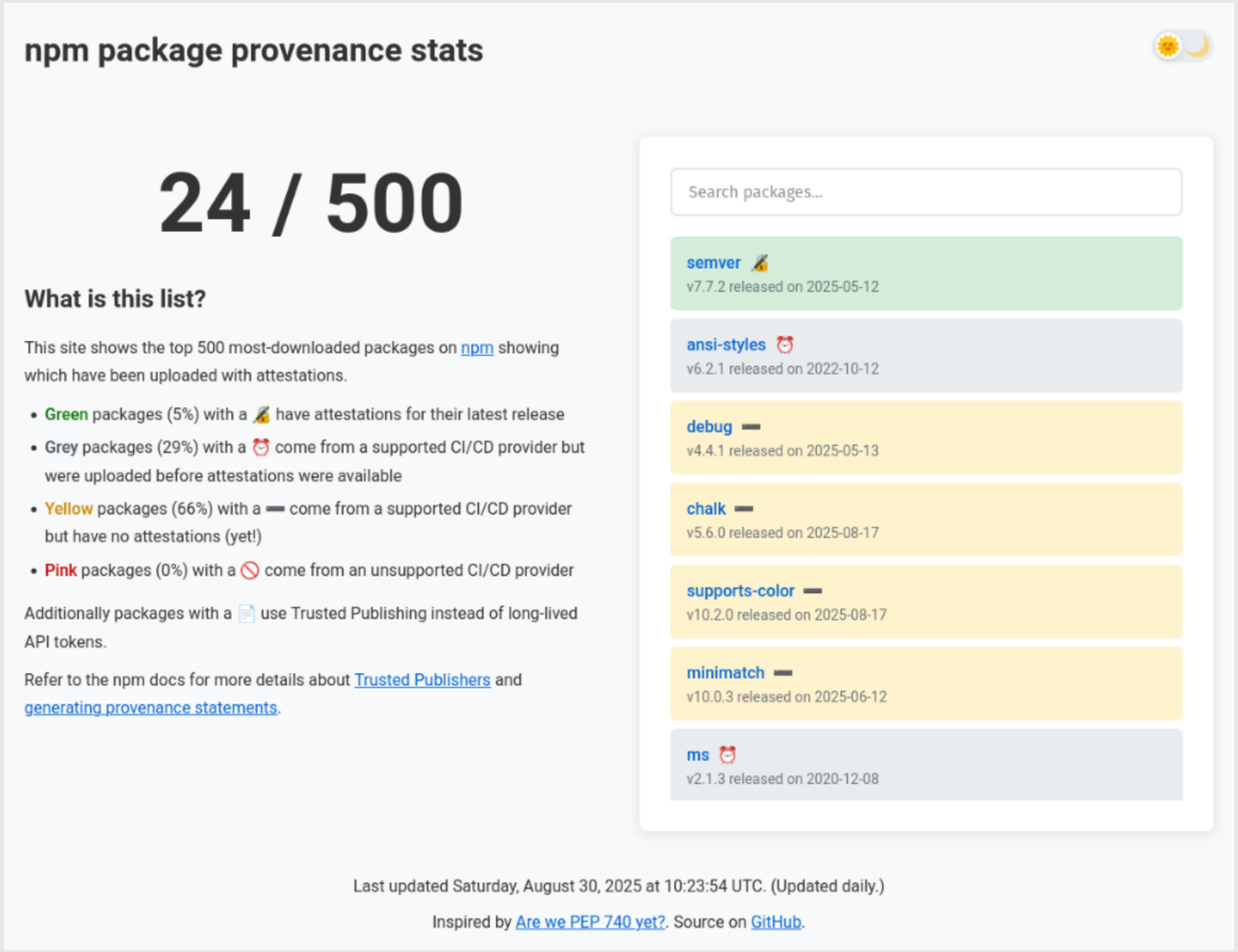

```
$ npm audit signatures
audited 507 packages in 7s

507 packages have verified registry signatures

10 packages have verified attestations
```

Source: https://docs.npmjs.com/generating-provenance-statements

npm package provenance stats



Source: https://jduabe.dev/npm-package-provenance-stats

Sigstore usage in the OSS ecosystem

Verifying Sigstore signatures in the OSS ecosystem

- PEP 761
- GitHub Artifact Attestations
- Homebrew provenance

PEP 761 – Deprecating PGP signatures for CPython artifacts

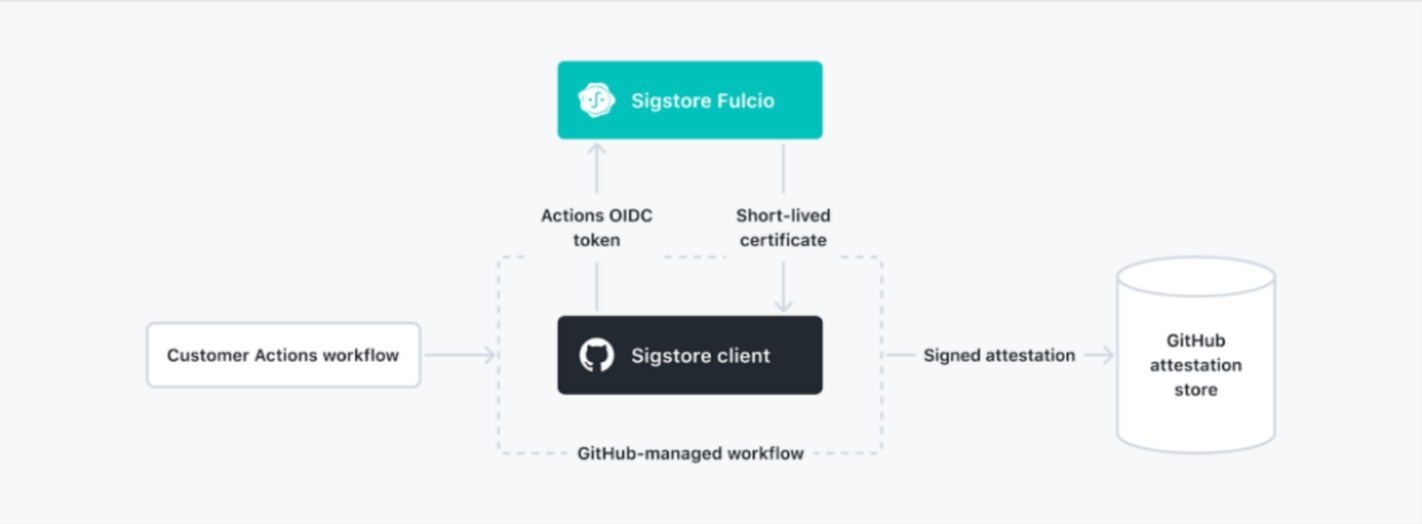
Release	PEP	Release manager	OIDC Issuer
3.7	PEP 537	nad@python.org	https://github.com/login/oauth
3.8	PEP 569	lukasz@langa.pl	https://github.com/login/oauth
3.9	PEP 596	lukasz@langa.pl	https://github.com/login/oauth
3.10	PEP 619	pablogsal@python.org	https://accounts.google.com
3.11	PEP 664	pablogsal@python.org	https://accounts.google.com
3.12	PEP 693	thomas@python.org	https://accounts.google.com
3.13	PEP 719	thomas@python.org	https://accounts.google.com
3.14	PEP 745	hugo@python.org	https://github.com/login/oauth

```
# download release artifact and verification bundle
$ wget https://www.python.org/ftp/python/3.14.0/Python-3.14.0rc1.tgz
$ wget https://www.python.org/ftp/python/3.14.0/Python-3.14.0rc1.tgz.sigstore

# verify using a sigstore client
$ uvx --prerelease=allow sigstore verify identity \
> --bundle Python-3.14.0rc1.tgz.sigstore \
> --cert-identity hugo@python.org \
> --cert-oidc-issuer https://github.com/login/oauth \
> Python-3.14.0rc1.tgz
$ OK: Python-3.14.0rc1.tgz
```

Source: https://www.python.org/downloads/metadata/sigstore/

GitHub attestations and brew provenance




```
# github artifact attestations can be verified using the gh cli
$ gh attestation verify my-artifact.tar.gz --owner my-organization

# homebrew provenance is still in beta
$ export HOMEBREW_VERIFY_ATTESTATIONS=1
$ brew install uv

# use verify manually (depends on gh cli)
$ brew verify uv
```

Source: https://github.blog/news-insights/product-news/introducing-artifact-attestations-now-in-public-beta/

Takeaways

- Trusted publishing improves security by removing the need for a preexisting shared secret
- Digital attestations provide verifiable provenance for Python package distributions
- Downstream verification -- integration with tooling like `pip` and `uv`
- Not a *panacea* -- one step that improves software supply chain security

Thank You