

Who publishes attestations on PyPI and npm — and who stopped?

EasyxLab · EasyByte Hub S. Coop. Mad. · Study S5

Working draft Published 2026-10-03

EasyxLab · study S5 · snapshot of 2026-10-02 · working draft, not peer-reviewed

Abstract

PyPI has accepted PEP 740 digital attestations since October 2024. They let anyone check that a file was built by a named repository and workflow. Most of PyPI has never used them: in the 15,000 most-downloaded projects (hugovk's list of 2026-10-01), 11,336 of the 14,995 we could analyse (75.6%) have never uploaded an attested file.

Adoption. The latest version is attested for 3,479 of 14,995 projects (23.2%), and for 31.1% of those whose latest version was first uploaded after 2024-10-01. Adoption falls from 59.0% in the top 100 to 20.3% in ranks 10,001–15,000. The share of projects whose latest version is attested rose from 4.0% at the end of November 2024 to 23.2% at the end of September 2026.

Projects that stopped. 148 projects uploaded an attested file and later made a newer release without one. For 138, the line `pip` installs by default had attestations and its latest version has none: 3.8% of the 3,659 projects that ever attested. After rule-based classification and a one-by-one review by the study agent:

CAUSE	PROJECTS
tool or workflow change	96
one isolated unattested release	15
attestations restored after the latest release	1
unknown	26

Among the 96, 53 publishing workflows now run `uv publish` with no attestation step. The other 10 projects have unattested pre-releases or maintenance releases, while the version `pip` installs is still attested or was never attested.

Publisher changes. At least 307 attested projects (382 changes) changed publisher between attested versions. Of the 188 changes that move to another repository or publisher kind, 160 are confirmed by public data and 16 are consistent with the project's own metadata.

npm. In a 1,500-package npm sample, 30.8% of the top 1,000 publish provenance; 4 of 459 packages that ever did (0.9%) stopped.

Not attesting is the norm, and stopping is a change of release tooling, not a sign of compromise.

1. Why this matters

PEP 740 adds a `provenance` key to each file in PyPI's JSON simple API. It points to a provenance object that binds the file's attestations to "the identity that produced them" (PyPI Integrity API documentation): a GitHub or GitLab workflow, or a Google service account. PyPI "will reject non-Trusted Publisher attestations at upload time". With the official PyPA action, attestations "are generated and uploaded automatically by default, with no additional configuration necessary"; this default arrived in v1.11.0 (2024-10-30).

Attestations support two checks. One is point in time: this file came from that repository. The other is continuity: the next release comes from the same place. The continuity check is only meaningful if projects keep attesting, so we ask:

1. how much of the popular ecosystem attests, and since when;
 2. how many projects stopped, and why;
 3. how often the attesting identity changes, and whether public data explains it;
 4. how npm compares.
-

2. Prior work

We searched on 2026-10-02 and 2026-10-03. The queries and raw results are in `data/sources/prior_work_search.json`.

- **arXiv API** (`export.arxiv.org/api/query`): `all:"PEP 740"` , `all:attestations AND all:PyPI` , `all:"trusted publishing" AND all:PyPI` , `all:provenance AND all:npm AND all:sigstore` , `all:sigstore AND all:adoption` , `all:"software signing" AND all:registries` , `all:"package registries" AND all:signing` , `all:"build provenance" AND all:npm` . Nothing matched PEP 740 or PyPI Trusted Publishing.
- **Crossref** (`api.crossref.org/works?query.bibliographic=`): eight queries on PEP 740, Sigstore adoption, npm provenance, PyPI Trusted Publishing and registry signing. Only one relevant record came back (below).
- **Semantic Scholar** (`api.semanticscholar.org/graph/v1/paper/search`): every query was answered with HTTP 429, so this source returned nothing.
- **GitHub search** (`gh search repos`): "PEP 740 attestations adoption", "pypi attestations regression", "are-we-pep740-yet", "npm provenance adoption statistics". The only hit was `trailofbits/are-we-pep740-yet` .
- **Standards and vendor documentation**: PEP 740, 691 and 700, docs.pypi.org, the gh-action-pypi-publish README, the uv guide and npm's documentation (`data/sources/excerpts.json`).

The closest results:

- **Schorlemmer et al., "Signing in Four Public Software Package Registries: Quantity, Quality, and Influencing Factors"** (IEEE S&P 2024, doi:10.1109/SP54263.2024.00215, arXiv 2401.14635). It measures signing in Maven, PyPI (PGP era), Docker Hub and Hugging Face. It reports that "getting

- started is the hard part -- once a maintainer begins to sign, they tend to continue doing so", and that "signing adoption is primarily affected by registry policy". It predates PEP 740.
- **arXiv 2503.00271, "Why Johnny Adopts Identity-Based Software Signing: A Usability Case Study of Sigstore."** A qualitative study ("interviews with 17 industry experts") with no registry measurements.
 - **Trail of Bits, *Are we PEP 740 yet?*** It tracks "the top 360 most-downloaded packages on PyPI" daily, by latest release. On 2026-10-02 it showed 158 green of 360. This is the closest existing measurement, and for its 360 packages it already answers our first question.

What this study adds:

- latest-release status across 15,000 projects instead of 360;
 - a monthly history since October 2024;
 - a defined, reproducible count of projects that stopped attesting, with causes backed by commits;
 - an audit of publisher changes through the Integrity API for every attested project;
 - an npm comparison on the same definitions.
- We found no prior count of stops, or of publisher changes, under PEP 740 or npm provenance. The search above is not exhaustive.

3. Data

SOURCE	WHAT WE COLLECTED	REQUESTS
hugovk's <i>top-pypi-packages</i> (<code>last_update</code> 2026-10-01 12:40:51; 30-day downloads)	the population, fetched at a pinned commit and not redistributed	–
PEP 691 JSON simple index, one per project	every file with its PEP 700 <code>upload-time</code> and PEP 740 <code>provenance</code>	15,000
PyPI Integrity API	the publisher of attested files	8,384
<code>npm-high-impact</code> 1.13.0 (2026-06-08)	the top 1,000 plus 500 random packages from ranks 1,001–15,916; <code>dist.attestations</code> for every version	939 + 1
GitHub	for each stop, the workflow named by the last attestation at <code>HEAD</code> (following reusable workflows, local actions, <code>make</code> targets and called scripts) and the commits touching it since the last attested release	–

Coverage. 14,995 projects are analysable. The other five are `aaaaaaaa` (HTTP 404), `vnai` and `vnstock` (no files), `pikobs` (files only after the list date) and `types-pkg-resources` (all files yanked). Files uploaded after 2026-10-01 12:40:51 UTC are ignored.

Conduct. Requests were limited to 7 per second per host with an identifying User-Agent. We collected no field identifying who uploaded a file.

4. Method

`METHOD.md` has the full definitions and rules.

Definitions.

- A version is **attested** if a non-yanked file has provenance. Attestation is dated by that file's own upload time.
- The **latest version** is the highest stable PEP 440 version.
- A project **stopped** if its newest version (by first upload) is unattested while an attested file was uploaded before it (148 projects).
- It stopped on the **pip-default line** if its latest version is unattested while an earlier stable, non-backport version was attested (138 projects). Projects that only ever attested pre-releases or backports are therefore not counted.

Causes. Rules applied in order:

1. secondary pipeline;
2. attestations re-added after the latest release (restored);
3. tool or workflow change: the workflow file was deleted while its repository is readable, or it uploads with a non-attesting tool, sets `attestations: false`, authenticates with an API token, or pins the action below v1.11;
4. isolated release: one unattested release while the workflow still attests;
5. otherwise **unknown**, including repositories that are not publicly readable.

Publisher changes are located by bisection over attested versions. Each change counts as confirmed if it stays in the same repository, keeps the same owner, or if the old and new repositories resolve to the same repository through GitHub redirects. A change whose new repository is declared only in the project's own metadata counts as consistent with it, not confirmed.

4.1 Automation and review

AI agents carried out this study: collection, rules, classification, review and all of this text. An independent AI review agent then checked it and recomputed every headline figure (`private/` , not published).

The study agent reviewed 57 projects one by one on 2026-10-02 (`data/pypi_review_sample.csv`):

- all 24 in the top 1,000 at that time;
- from the rest, 11 tool or workflow changes, 5 indeterminate, 8 isolated uploads, 8 pre-release pipelines and 1 artefact. The draw was not scripted, so this composition is a record of what was reviewed, not a stratified design.

The independent reviewer found one of the agent's overrules wrong (fastmcp). After the rules were corrected, 55 of the 57 are still in the set and the agent's verdicts agree with the rules in 47. Each of the 8 overrides cites a commit (`data/pypi_regressions_reviewed.csv`). The 92 unreviewed projects keep their rule class.

5. Results

5.1 Adoption by popularity

RANK BAND	LATEST ATTESTED	AMONG ELIGIBLE ¹	EVER ATTESTED
1–100	59.0%	60.8%	62.0%
101–500	37.8%	42.4%	41.0%
501–1,000	32.6%	39.4%	35.2%
1,001–2,500	26.6%	32.4%	28.6%
2,501–5,000	25.3%	33.9%	26.8%
5,001–10,000	21.3%	29.2%	22.1%
10,001–15,000	20.3%	28.3%	21.1%
all	23.2% (3,479)	31.1% (3,478 of 11,176)	24.4% (3,659)

¹ Latest version first uploaded on or after 2024-10-01. The first attested file in the data dates from 2024-10-03.

- **Partial attestation.** In 13 projects only part of the latest version's files are attested.
- **Publisher kinds.** The last attested publisher is GitHub for 3,507 projects, Google for 147 and GitLab for 5.

5.2 Adoption over time

MONTH END	VERSIONS	VERSIONS ATTESTED	PROJECTS RELEASING THAT ATTESTED	PROJECTS WHOSE LATEST IS ATTESTED
2024-11	12,502	11.6%	14.5%	4.0%
2025-06	14,111	14.6%	17.7%	11.5%
2025-12	14,643	19.2%	23.4%	15.1%

MONTH END	VERSIONS	VERSIONS ATTESTED	PROJECTS RELEASING THAT ATTESTED	PROJECTS WHOSE LATEST IS ATTESTED
2026-06	20,359	30.7%	34.8%	21.0%
2026-09	28,539	39.6%	38.0%	23.2%

The monthly series is in `data/pypi_monthly.csv` and `data/tables.md`. After the November 2024 jump, when the PyPA default changed, growth is gradual: about one percentage point a month in the last column. The version-level share is noisy, because a few projects publish hundreds of versions. All series use today's top list (§7).

5.3 Projects that stopped attesting

Among the 138 pip-default stops:

- **Tool or workflow change (96):**
 - 53 workflows at `HEAD` run `uv publish` with no attestation step;
 - 19 attesting workflows were deleted (their repositories are readable);
 - 12 set `attestations: false`;
 - 2 authenticate with an API token;
 - 2 use other uploaders;
 - 8 are agent overrides backed by a commit, for example a move into a monorepo or a workflow disabled by a named commit.
- **One isolated unattested release (15).**
- **Restored (1).**
- **Unknown (26):** 7 have publishing repositories that are not publicly readable, and 4 have upload steps our patterns do not recognise.

21 of the 138 had only one attested stable release. A median of 8 unattested releases followed the last attested one.

About uv. The uv guide states that "uv publish does not currently generate attestations; attestations must be created separately before publishing". uv has uploaded attestations found next to the distributions since 0.9.12 (2025-11-25), and its GitHub guide has shown a separate `astral-sh/attest-action` step since 0.12.3 (2026-08-07). The finding is therefore narrow: these projects moved to `uv publish` **without adding an attestation step**. Some did so knowingly. zizmor's workflow, for one, says "# TODO: Re-enable attestations once reusable workflows work."

Named cases, with their evidence:

- **fastapi** attested 53 versions (0.115.5–0.128.0). Commit b4ba7f465223 ("Migrate to uv (#14676)", 2026-01-10) replaced the PyPA action with `uv publish`.
- **typer** made the same change in 39353afc8528 (#1472).

- **fastmcp** stopped after 0.2.0. Commit 67de515d1a80 (2024-11-30) switched to `uv publish`.
- **httpx2** and **httpcore2** stopped with 2b13fa92e418 ("Publish distributions with uv trusted publishing").
- **watchfiles** stopped with 5be19ff595e5.
- **supabase** publishes through `make publish`, whose target is `uv publish` (054f605f1d8d, "use uv instead of poetry"). Five sibling packages moved into the same monorepo, for example b8b0502405e8.
- **databricks-sdk**: 6b16a7830cdb ("Temporarily disable automated releases", three days after the last attested release).
- **ormsgpack** added `astral-sh/attest-action` in ae64f6849962 (2026-06-03), after its latest release. Its next release should attest.
- **langchain-core** is not a pip-default stop. Its only attested version, 0.1.53, is a backport; the main line opted out on 2024-10-30, the day the default changed (92024d0d7d1a).

Projects not named here. Projects whose publishing repository is not publicly readable are counted as unknown and not named. The full list is in `data/pypi_regressions.csv`.

Stops over time. By half-year of the last attested stable release, stops were 14, 26, 43 and 40. Measured against the projects with an attested stable release by the end of each half, that is 1.8%, 1.6%, 1.9% and 1.2%. Recent halves are right-censored: a stop only shows once an unattested release follows.

5.4 Publisher changes

The figures below are lower bounds, because bisection cannot see A→B→A sequences inside an interval. At least 307 of the 3,659 attested projects changed publisher between attested versions, 382 changes in all; 146 other projects changed only the deployment environment, which we do not count.

- **194 changes** use another workflow file in the same repository.
- **188 changes** move to another repository or publisher kind:
 - 160 confirmed: 67 keep the same owner and 93 resolve, through GitHub redirects, to the same repository as the old one;
 - 16 consistent with the project's own metadata;
 - 12, in 5 projects, not confirmed by public data. Their repositories are mostly gone, so this is not evidence of anything wrong.
- **Dedicated release repositories** are legitimate and growing. numpy and scipy have attested from `numpy/numpy-release` and `scipy/scipy-release` since their first attested release. scikit-learn and matplotlib moved to `<project>-release` repositories with the same owner.

5.5 npm

RANK BAND	NPM LATEST WITH PROVENANCE	PYPI LATEST ATTESTED
1–100	22.0%	59.0%
101–500	30.5%	37.8%

RANK BAND	NPM LATEST WITH PROVENANCE	PYPI LATEST ATTESTED
501–1,000	32.8%	32.6%
random 500 of 1,001–15,916	27.4%	–

- **Adoption.** Similar from rank 500 on, but the npm top 100 is far below PyPI's.
- **Stops.** 7 packages have an unattested `latest` above an attested version. Publish times confirm 4 of them (4 of 459 ever-attested packages, 0.9%, against 3.8% on PyPI):
 - `why-is-node-running`: an isolated release; its workflow still passes `--provenance` ;
 - three `@vercel` packages: cause unknown.
- **Not stops.** The other three started attesting on an older line after `latest` was published.
- **Why fewer stops (hypothesis).** npm's documentation says trusted publishing generates provenance automatically, "you don't need to add the `--provenance` flag", so changing npm tools keeps it by default. We did not test this.

5.6 Comparison with *Are we PEP 740 yet?*

We agree on 359 of its 360 packages. The exception is `sqlalchemy`: 64 of the 72 files of its latest version are attested (the `sdist` and seven `win_arm64` wheels are not). We count any attested file; the tracker counts the release as not attested. The two snapshots are 27.5 hours apart (our cut-off 2026-10-01 12:40 UTC, its update 2026-10-02 16:12 UTC), and 20 of the 360 released in between.

6. Discussion

- **Most popular projects have never attested.** Among those that have, about 4% stopped on the line `pip` installs.
- **Most stops are tool changes.** Where commits show the cause, it is an ordinary tooling change, most often a move to `uv publish` without the separate attestation step `uv now documents`.
- **What users can conclude.** A stop removes a guarantee; it does not show that one was broken. A continuity check should name the probable cause.
- **Unconfirmed publisher changes are rare.** They are 5 projects in 3,659, so flagging them would not flood users with alerts.

7. Limitations

- **Survivorship.** The population is today's list (a 30-day download window), so projects that left the top 15,000 are missing.
- *****The version `pip` installs***** ignores Requires-Python and wheel availability.
- **Ordering.** Versions are ordered by upload time. Backports are handled only through the "mainline" test.
- **Publisher changes** are lower bounds, and one file per version is checked.

- **Workflow evidence** is read at today's `HEAD`. A deleted file or an unreadable repository leaves the cause unknown, and the Atom feed lists only recent commits.
 - **Review.** It is one AI agent's judgement, not blind to the rule class, on a sample that was not drawn by script.
 - **npm.** It is a sample: SemVer order, publish times only for candidates, and no time series.
 - **Moving target.** The registries change daily. The published compact data is the reference.
-

8. Data, code and licences

- **Data.** `data/compact/` holds the per-file PyPI data (files since 2024-09-01, with provenance and publisher where read), per-version aggregates for older files, the npm per-version data and the reduced evidence (commits, redirects, publishers).
 - **Reproduction.** `scripts/run.sh` reruns the analysis offline from those files. `FETCH=1` rebuilds them from the network with pinned inputs (`METHOD.md §8`).
 - **Licences.** Code is Apache-2.0; data and text are CC BY 4.0. The source lists are fetched, not redistributed.
-

Competing interests

EasyByte, the cooperative behind EasyxLab, develops attest-lint, an open-source tool that flags, in a project's lockfile, pinned dependencies whose attestations regressed: the phenomenon this study measures. attest-lint did not produce any figure here; the study uses its own scripts.

References

- PEP 740; PEP 691; PEP 700. <https://peps.python.org/>
- PyPI documentation: Attestations; Integrity API. <https://docs.pypi.org/>
- pypa/gh-action-pypi-publish, README and v1.11.0 (2024-10-30). <https://github.com/pypa/gh-action-pypi-publish>
- uv guide, "Building and publishing a package"; uv 0.9.12 and 0.12.3 release notes. <https://docs.astral.sh/uv/>
- npm Docs, "Trusted publishing", "Generating provenance statements". <https://docs.npmjs.com/>
- Trail of Bits, Are we PEP 740 yet? <https://trailofbits.github.io/are-we-pep740-yet/> (2026-10-02)
- T. R. Schorlemmer et al., Signing in Four Public Software Package Registries, IEEE S&P 2024. doi:10.1109/SP54263.2024.00215
- arXiv:2503.00271, Why Johnny Adopts Identity-Based Software Signing.
- H. van Kemenade, top-pypi-packages (commit e6b5b398c1fc). T. Wormer, npm-high-impact 1.13.0.

Cite as: EasyxLab (2026). Who publishes attestations on PyPI and npm — and who stopped? Study S5. EasyByte Hub S. Coop. Mad. <https://github.com/easybytehub/easyxlab>

Cite this study

CITATION

EasyxLab (2026). Who publishes attestations on PyPI and npm – and who stopped? Study S5. EasyByte Hub S. Coop. Mad.
<https://github.com/easybytehub/easyxlab/tree/main/studies/s5-pypi-attestations>

BIBTEX

```
@techreport{easyxlab_s5,  
  title      = {Who publishes attestations on PyPI and npm – and who stopped?},  
  author     = {{EasyxLab}},  
  institution = {EasyByte Hub S. Coop. Mad.},  
  number     = {S5},  
  year       = {2026},  
  url        = {https://github.com/easybytehub/easyxlab/tree/main/studies/s5-  
pypi-attestations}  
}
```