COPIED, FETCHED, AND LEFT WITHOUT EVIDENCE: PERMISSIVE WASHING IN AI-AGENT SUPPLY CHAINS

Anonymous authors
Paper under double-blind review

ABSTRACT

Permissive washing occurs when an artifact declares MIT, Apache-2.0, or BSD-3-Clause without the matching license text and required copyright holders that form its compliance payload. Documented in datasets and models, it now affects an AI-agent ecosystem built on shared Skills, Model Context Protocol (MCP) servers, and Plugins. These components are reused and distributed like software. Authors copy folders and files into repositories, and installers and registries deliver selected directories and packages. Their permissive licenses condition such copying and redistribution on retaining notices, so losing the compliance payload can have legal consequences. We trace the compliance payload across publication, reuse, composition, and distribution. Among 345,371 assessed Skill occurrences, 59.4% lack a complete compliance payload anywhere from folder to repository root, and only 9.5% carry one inside the Skill folder. Even when the reference was complete before reuse, 41.1% of Skill file copies and 58.5% of MCP server file copies (copies made without forking) lose it. Distribution loses it the same way. Claude Code leaves 173 of 229 repository-complete Skill deliveries without either part, Codex agrees on every shared outcome, and 22 of 121 registry archives for MCP servers omit the compliance payload. The compliance payload survives when the unit copied or delivered encloses it, not merely when the repository does. We release all data and code in a replication package.

1 INTRODUCTION

AI agents increasingly gain capabilities by reusing components that others build and share. Skills package reusable instructions, MCP servers expose reusable tools, and Plugins bundle such components for installation across agents. Adoption is already substantial. MCP SDKs have surpassed 25 million weekly downloads (Hasan et al., 2026), a single Skill on the skills.sh registry records 3.6 million installations (Vercel, 2026), and Skill instances in Plugin marketplaces grew 22-fold in six months (Hereiz et al., 2026). These components spread through direct reuse, as users copy Skill folders, fetch MCP server packages, and install Plugins through marketplaces (Destefanis et al., 2026; Anthropic, 2026e; Model Context Protocol, 2026a).

Permissive licenses enable this sharing by allowing others to copy, modify, and redistribute an artifact. MIT or Apache-2.0 already label 53.5% of Hugging Face datasets (Stalnaker et al., 2026). In our corpora, MIT, Apache-2.0, and BSD-3-Clause cover 55.4% of reachable MCP server repositories, 61.3% of Plugin repositories, and 428,093 Skill occurrences. These permissions are conditional. Redistributed copies must preserve the matching license text and required copyright holder, which Jewitt et al. (2026b) call the *compliance payload*. They call an artifact that carries a permissive label without its compliance payload a case of *permissive washing*. Recipients then receive an artifact labeled for reuse without what its license requires them to preserve when they redistribute it.

Agent-component reuse more closely resembles software reuse than model training. Whether training requires permission remains subject to fact-specific fair-use analysis (Bartz v. Anthropic PBC, 2025; Kadrey v. Meta Platforms, Inc., 2025). Agent components instead circulate through copying and redistribution, as software does, so a copy without its compliance payload raises the same licensing problem as software redistributed without its license terms. Courts have enforced such conditions in software (Jacobsen v. Katzer, 2008; Artifex Software, Inc. v. Hancom, Inc., 2017), although these cases establish enforceability rather than liability for any copy we observe (Appendix N).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Software components usually avoid this problem because the unit that reuse and packaging copy carries the license. A fork copies the whole repository, including its root license file (GitHub, 2026), and a published npm package always includes its root license file (npm, 2026). Agent ecosystems often copy components as smaller units. A copied Skill folder or a marketplace-selected Plugin directory can leave the files above it behind (Figure 1), and neither format requires the license text and copyright holder inside that directory (Agent Skills, 2025; Anthropic, 2026e; OpenAI, 2026b); package builds can likewise omit files elsewhere in the source tree. We trace the compliance payload through these copied units at four checkpoints: *license integrity* at publication (C1), *attribution integrity* after copying (C2), *composition integrity* after bundling (C3), and *distribution integrity* at delivery (C4).

We find that most permissively labeled Skills lack a complete compliance payload at publication, and that copies often lose one that their reference carried. Among 345,371 assessed Skill occurrences, 59.4% lack a complete compliance payload anywhere from their folder to the repository root, and only 9.5% carry one inside the Skill folder. Even when the reference carried a complete compliance payload before copying, 41.1% of Skill file copies and 38 of 65 MCP server file copies (copies made without forking) fail to preserve it, compared with 2 of 18 forks (§4.1–§4.2).

Composition can preserve the compliance payload without placing it inside the component that distribution later copies. Of 611 bundled Skill copies that preserve a complete compliance payload, 591 depend on files above the Skill folder. In a separate installation cohort, Claude Code delivers 173 of 229 repository-complete Skill deliveries without either part of the compliance payload, and Codex agrees on every shared outcome. Every applicable license and notice file already inside the selected directory survives byte-for-byte, locating the loss at directory selection. Registry archives follow the same pattern, with 22 of 121 lacking the compliance payload; in 19 of them, the matching source license text sits above the package directory (§4.3–§4.4).

Beyond these cohorts, the same setup is common. Across the 4,671 Plugins with a complete root compliance payload, 93.3% of the 26,343 Skills with a complete compliance payload keep it only above the Skill folder, so any delivery that selects that folder would leave it behind. This paper makes four contributions: (1) an audit of permissive washing at four checkpoints of the AI-agent supply chain, from publication through reuse and composition to delivery; (2) evidence that the compliance payload survives when the copied or delivered unit encloses it, not merely when the repository does; (3) component-level checks for publishers, marketplaces, installers, and registries; and (4) a replication package with all scripts, data, ledgers, and audit artifacts.

2 Reuse and distribution in AI-agent supply chains

We study Skills, MCP servers, and the Plugins that bundle them. A Skill packages task instructions and supporting files, and an MCP server exposes tools through the Model Context Protocol; authors publish and reuse both as standalone artifacts under open specifications that several hosts support (Agent Skills, 2025; Model Context Protocol, 2026b), and Plugins are where components are assembled and installed (Anthropic, 2026e). Skill folders and Plugin marketplace entries are *agent-native formats*, which agent ecosystems add on top of software infrastructure, whereas MCP servers are published as ordinary software repositories and packages. Other component kinds follow host-specific formats or configure a single project, so they fall outside the audit (Appendix B).

Components change hands at two points, and permissive licenses attach conditions to both (Figure 1). In *reuse*, an author copies a component into their own repository, for example by copying a Skill folder, forking an MCP server, or bundling components into a Plugin; in *distribution*, a user receives a component by installing a Plugin or fetching a package that a Plugin references. The two differ in who chooses the files. During reuse, the author decides what to copy, so each copy keeps or omits the license text and copyright holder by that choice. During distribution, an installer or package build copies a fixed unit, so a compliance payload stored outside that unit is lost for every recipient.

2.1 Reuse

All three licenses require the license text and copyright holder to stay with every copy (Open Source Initiative, 2026b;a; Apache Software Foundation, 2004), so a component published without them gives reusers nothing to keep, and its author forgoes the copyright statement that defeats an innocent-

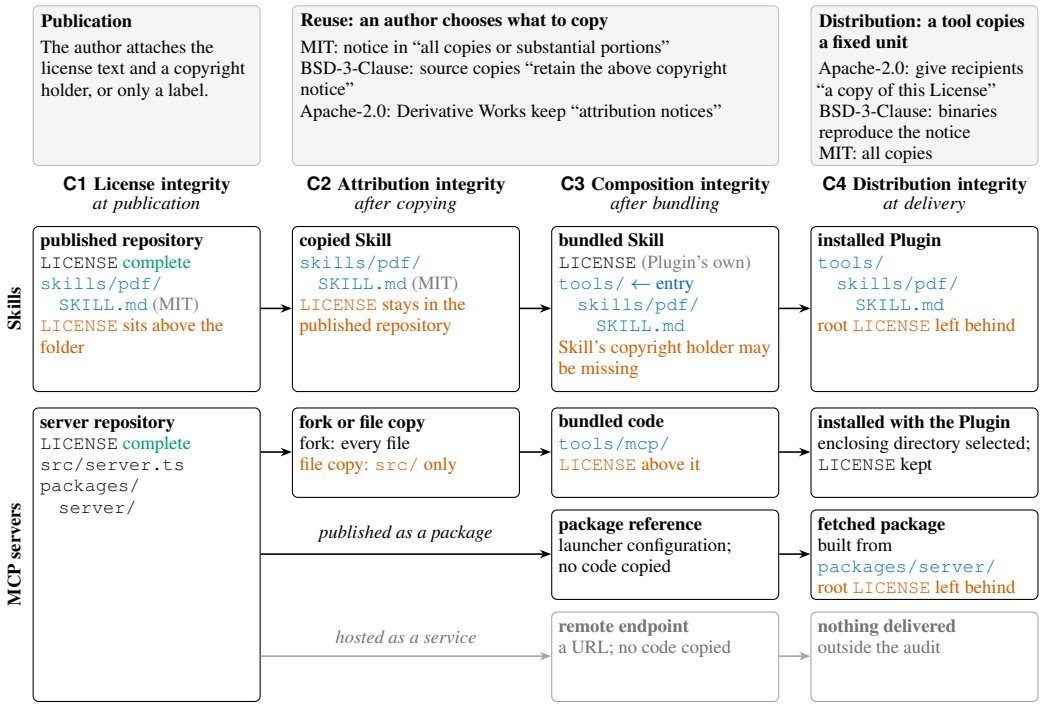


Figure 1: How Skills and MCP servers reach users, and where their compliance payload can be left behind. In reuse, an author chooses what to copy; in distribution, an installer copies the directory a marketplace entry selects, or a package build decides what a registry archive contains. The bands quote each license's requirement (Open Source Initiative, 2026b;a; Apache Software Foundation, 2004).

infringement defense (U.S. Copyright Office, 2021) and the warranty disclaimer that the license text carries. Hosts treat Skills as directory-scoped artifacts (Anthropic, 2026c;b; OpenAI, 2026c;a), and authors copy the folder between repositories (Destefanis et al., 2026), so a compliance payload stored above a Skill folder stays behind. An MCP server is copied by a fork, which copies every tracked file including the license (GitHub, 2026), or as what we call a *file copy*, in which the author selects the files that enter the new repository; a copied Skill folder is a file copy, as is MCP server code copied without forking. A Plugin adds a composition step. Its publisher copies Skills and MCP server code into a repository that carries the Plugin's own license, so a root-level license check can pass while a bundled component's license text or copyright holders are missing. Courts have enforced these conditions for reuse and composition. In *Jacobsen v. Katzer*, reused files omitted their copyright holders and license references, the Federal Circuit held those terms to be conditions of the copyright grant, and the litigation settled for $100,000 and a permanent injunction (Jacobsen v. Katzer, 2008; 2010). In *Entr'Ouvert v. Orange*, a GPL-licensed library built into a government portal without its copyright and license statements drew a finding of infringement and €800,000 in damages (Entr'Ouvert v. Orange, 2024).[1]

2.2 Distribution

Distribution shifts the same question to the recipient's copy. No license fixes one location for the license text and copyright holder, and BSD-3-Clause and Apache-2.0 accept accompanying documentation or files (Open Source Initiative, 2026a; Apache Software Foundation, 2004), so we inspect the files the recipient receives. For components bundled in a Plugin, the boundary is the directory that the marketplace entry selects (Anthropic, 2026a; OpenAI, 2026b). Claude Code copies nothing outside it (Anthropic, 2026d), and Codex loads an installed copy of it from a local

[1]These cases show that open-source license conditions can be enforced; they do not determine liability for the components we audit (Appendix N).

cache (OpenAI, 2026b), so a compliance payload above that directory stays behind unless the entry selects an enclosing directory. For registry-referenced MCP servers, the Plugin copies no code and the package publisher's build decides what the user fetches (Model Context Protocol, 2026a), so a compliance payload outside the build's inputs, or excluded by its packaging rules, never reaches the user. Distribution failures have drawn enforcement as well. In *Welte v. D-Link*, firmware distributed without the GPL text, a warranty disclaimer, or the source code led the Frankfurt court to uphold the license and order reimbursement of the enforcer's costs (Welte v. D-Link Deutschland GmbH, 2006).

3 Data and method

We measure the compliance payload at the four checkpoints of Section 2 on the same corpora with the same detection rules. Each checkpoint changes only the unit we assess and what its compliance payload is compared against, and each has its own population rather than a sequence that every component follows. Gates define the assessed sets rather than samples, so we report exact counts (Appendix F).

Data collection. For Skills, we use GitSkills (Destefanis et al., 2026), which collected 3,797,117 `SKILL.md` files from 282,200 public repositories in July 2026. For MCP servers, we combine 13,875 servers from the MCP.so registry (Lin et al., 2025) with 2,281 collected from GitHub (Toeppe et al., 2026); 12,157 of the 14,740 deduplicated repository names are reachable. For Plugins, we use the marketplace dataset of Hereiz et al. (2026) without its ten-star filter and clone 9,743 of the 10,432 repositories we attempt (Appendix C).

Filtering. We retain components that declare exactly MIT, Apache-2.0, or BSD-3-Clause, in front matter for Skills and in a root manifest or license file for MCP servers and Plugins; compound expressions, exceptions, and conflicting declarations remain unresolved (Appendix D). This leaves 428,093 Skill occurrences, 6,737 MCP server repositories, and 5,976 Plugin repositories. We keep a Skill occurrence only if its recorded Git blob is still present in the repository tree we retrieve, which leaves 345,371 occurrences covering 144,615 distinct contents in 28,126 repositories.

Detecting the compliance payload. We use ScanCode Toolkit 32.5.0 (AboutCode, 2026). Following Jewitt et al. (2026b), license text counts only when a text-like rule covers at least 90% of the rule, and copyright holders come only from detected copyright statements (Appendix D). For a Skill, the *governing chain* is its folder and every ancestor up to the repository root, and we record the nearest directory that holds both parts. We assess MCP servers and Plugins at the repository root because their source corpora catalog repositories.

C1: License integrity at publication. We assess 345,371 Skill occurrences along their governing chains and 6,737 MCP server and 5,976 Plugin repositories at their root. Because BSD-3-Clause and Apache-2.0 accept notices in accompanying documentation, a sensitivity analysis also scans README files under the same rules (Appendix E).

C2: Attribution integrity after reuse. We remove each Skill's front matter, which can change while the body stays the same (Destefanis et al., 2026), hash the body, and search it across the full GitSkills corpus. Its earliest dated occurrence is the earlier published Skill that later occurrences match, and we call it their *reference*; a later occurrence in another repository is a *copy*. We retain a copy only when its reference passes C1, belongs to a different repository, and strictly predates it, which leaves 2,317 Skill copies; none of the 185 repositories that hold them is a fork, so every Skill copy is a file copy. A copy *preserves* the compliance payload when its governing chain contains matching license text and every copyright holder that its reference requires, and otherwise *loses* it. MCP server forks come from GitHub parent metadata. We detect MCP server file copies by pairing repositories that share at least 12 distinctive files (Git blobs present in 2 to 20 repositories) covering at least 55% of the smaller repository's files; a clone check on 431 sampled pairs puts this rule's precision at 83.4% (Appendix G). Ordering each pair by the date both repositories held the shared files, excluding ties and missing histories, and requiring a reference that passes C1 leaves 23 explicit forks and 65 file copies.

C3: Composition integrity inside Plugins. We assess bundled components inside the 4,671 Plugins with a complete root compliance payload. The C2 rules leave 705 bundled Skill copies across 44 Plugins. For bundled MCP servers, we resolve local entrypoints, search GitHub for corroborating implementation code, and retain a copy only when its component is associated with an external repository that passes C1, which leaves 22 bundled MCP copies in 14 Plugins (Appendix F).

Timing check. A copy can preserve only a compliance payload that existed when it was made. For all 3,132 copies at C2 and C3, we date each reference's compliance payload by the earliest commit at which its governing chain holds matching license text and every required holder together, and keep a copy, regardless of its outcome, only if that date is no later than its copy date. The check excludes 594 copies, leaving 1,736 Skill copies, 18 forks, and 65 MCP server file copies at C2 and 700 bundled Skill and 19 bundled MCP copies at C3 (Appendix H).

C4: Distribution integrity for what users receive. We install a fixed cohort of 195 Plugins with a complete root compliance payload that bundle a Skill or MCP server with an earlier external reference, mapping their 245 marketplace entries to pinned snapshots and running Claude Code 2.1.280 and Codex 0.156.0 in network-disabled sandboxes (Appendix J). One bundled component installed through one marketplace entry is a *delivery*, and for each we compare the compliance payload in the full repository, in the directory the entry selects, and in the installed cache, whose bytes must match the snapshot. For registry-referenced MCP servers, we inspect the published archive for 349 registry requests covering 121 archives whose source repository passes C1 (Appendix L). An exposure census applies the C1 placement rule to the 28,399 permissively labeled Skills in the 4,671 Plugins with a complete root compliance payload and checks each selected-directory/license combination for the Plugin's own license text and holder (Appendix K).

4 RESULTS

4.1 C1: MOST PERMISSIVELY LABELED SKILLS LACK A COMPLETE COMPLIANCE PAYLOAD

Most Skills declare a permissive license without carrying both its license text and its copyright holder. Of 345,371 assessed Skill occurrences, 205,030 (59.4%) lack a complete compliance payload anywhere from their folder to the repository root, against 28.3% of MCP server repositories and 21.8% of Plugin repositories (Figure 2, C1). This gap is conservative. A Skill's governing chain includes the repository root, and the root is the only level at which we assess MCP servers and Plugins. MCP server and Plugin repositories carry a complete compliance payload at rates (71.7% and 78.2%) close to the 74.2% that Jewitt et al. (2026b) report for permissively labeled GitHub applications under comparable text-and-holder requirements. Large repositories hold many of the incomplete Skills but do not inflate the headline rate. The largest 1% by assessed Skill count contain 49.4% of Skills without a complete compliance payload, while weighting each of the 28,126 repositories equally raises the rate from 59.4% to 66.4%. Every assessed Skill lacks a complete compliance payload in 64.3% of these repositories (Appendix D). Reviewed README notices complete the compliance payload for only 886 more Skill occurrences and raise no population's rate by more than 0.3 percentage points (Appendix E).

Skills usually lack the license text itself, whereas MCP server and Plugin repositories that fall short often lack only the copyright holder. Matching license text is absent from 54.9% of assessed Skills, against 13.2% of MCP server and 13.1% of Plugin repositories, and from 189,453 of the 205,030 Skills without a complete compliance payload (92.4%). Among Skills that carry matching license text, 90.0% also name a copyright holder, so the typical incomplete Skill declares a permissive license without carrying its text. Among incomplete repositories, by contrast, 1,022 of 1,909 MCP servers and 523 of 1,305 Plugins carry the license text but lack a copyright holder.

Even Skills with a complete compliance payload usually keep it outside the folder that reuse copies. Of the 140,339 Skills with a complete compliance payload, 107,361 (76.5%) depend on an ancestor directory for completeness, including 83,763 that depend on the repository root. Only 32,978 of all assessed Skills (9.5%) carry a complete compliance payload inside their own folder. Because reuse typically copies the Skill folder (Destefanis et al., 2026; Anthropic, 2026b; OpenAI, 2026c), a Skill can be copied while its compliance payload stays behind.

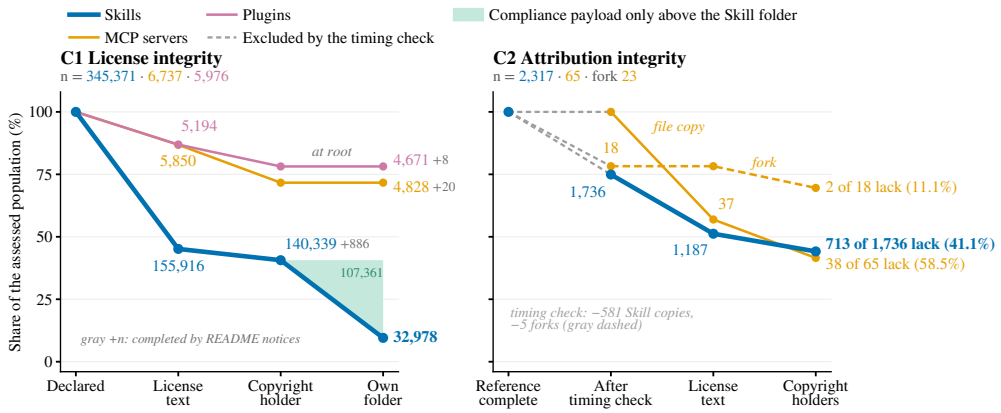


Figure 2: Compliance payload at publication (C1) and after reuse (C2). Lines count components still complete at each step as a share of the assessed population (n under each title). C2 line ends give the share of time-eligible copies that lack the compliance payload; gray dashes mark copies removed by the timing check. Shaded: compliance payload only above the Skill folder.

Together, these results show that a permissive label says little about whether a Skill is published with a complete compliance payload in the unit that reuse copies.

4.2 C2: File copies often fail to preserve the compliance payload

File copies fail to preserve the compliance payload far more often than forks, which copy the whole repository. Of 1,736 Skill file copies made after their reference carried a complete compliance payload, 713 (41.1%) lack the matching license text or a required copyright holder (Figure 2, C2). Among MCP servers, 38 of 65 file copies (58.5%) lack a complete compliance payload, compared with 2 of 18 explicit forks (11.1%). Strict provenance, direction, and timing requirements leave only 83 eligible MCP server pairs from 203 candidates, so this contrast describes these pairs rather than MCP servers in general. GitHub's fork mechanism copies the surrounding repository, including tracked root license files (GitHub, 2026), whereas an author making a file copy can take only the component files and omit a compliance payload stored above them. Five large references also lower the aggregate Skill failure rate. Excluding them, 556 of 963 Skill copies (57.7%) lack a complete compliance payload (Appendix H).

When a copy fails to preserve the compliance payload, the required copyright holder is always missing. All 713 failed Skill copies and all 38 failed MCP server file copies miss at least one copyright holder required by their reference. For 549 of these Skill copies (77.0%) and 28 of these MCP server copies (73.7%), the matching license text is also absent. Even the 2 of 18 forks that fail retain the license text but miss a required holder. A license-text-only check would therefore pass 164 Skill copies, 10 MCP server file copies, and both failed forks despite their missing copyright holder.

Adding license text and a copyright holder after the initial publication of a component does not update copies already made. We find 581 Skill copies made before their reference carried a complete compliance payload, including 573 from one reference. Those Skills already declared MIT when copied, but the reference added the matching license text and required copyright holder 26 days later. When we later observe the copies, none carries that completed compliance payload. Instead, 578 carry licensing material naming a different copyright holder, and three have no recognized license or notice file in our data. Five MCP server forks show the same ordering, with three retaining the license file present at the fork point (Appendix H). Because the reference's complete compliance payload did not exist when these copies were made, we exclude them from the failure rate; counting them as failures would raise the measured Skill failure rate from 41.1% to 55.8%. The publication gap observed in C1 can therefore persist in downstream copies after the reference later completes its compliance payload.

Together with C1, these results show that reuse can preserve only the compliance payload available when a component is copied. Copies of the 59.4% of Skills published without a complete compliance payload cannot inherit one, and later additions need not reach copies already made. Even when the

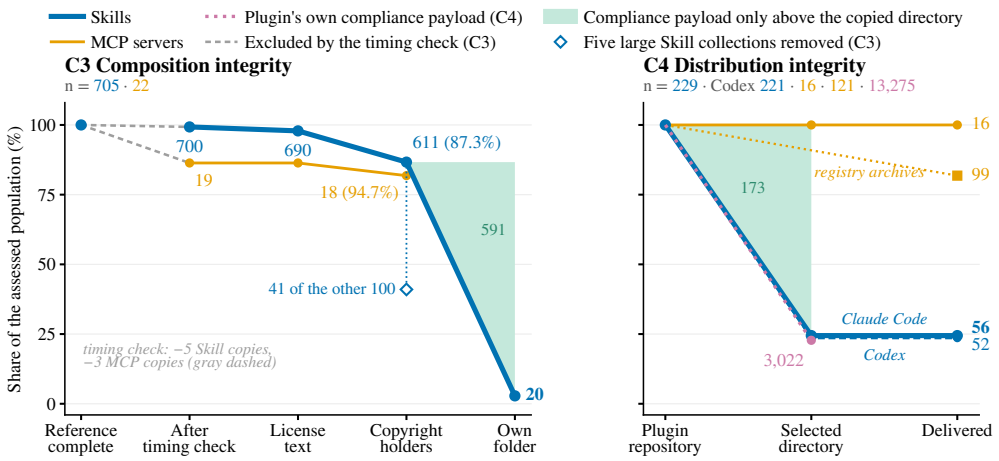


Figure 3: Compliance payload in composition (C3) and distribution (C4), drawn as in Figure 2. C3 line ends give the preserved share, and the diamond removes the five large Skill collections. C4 shows Claude Code (solid) and Codex (dashed) deliveries, registry archives, and the Plugin's own compliance payload. Shaded: compliance payload only above the copied directory.

reference is complete, 41.1% of Skill file copies fail to preserve its compliance payload, consistent with 76.5% of Skills with a complete compliance payload storing it above the copied folder.

4.3 C3: Plugin composition preserves the compliance payload, but mostly outside the component

Most bundled Skills preserve the compliance payload of the Skill they copy, but five large Skill collections account for most of them. Of 700 bundled Skill copies made after their reference had a complete compliance payload, 611 (87.3%) preserve the matching license text and every required copyright holder (Figure 3, C3). So do 18 of 19 bundled MCP servers. The 700 Skill copies come from only 41 of the 4,671 Plugins with a complete root compliance payload, and 600 of them copy one of five large Skill collections, which account for 570 of the 611 preserved copies. The other 100 copies preserve a complete compliance payload in only 41 cases, similar to the 42.3% of standalone Skill file copies that do so after removing their five largest references (Appendix H). The high aggregate rate therefore reflects the five collections that supply 600 of the 700 copies.

When a bundled Skill lacks a complete compliance payload, the missing part is almost always a required copyright holder, not the license text. All 89 incomplete Skill copies miss at least one required holder, but only 10 also lack the matching license text. The other 79 still have matching license text in their folder or a directory above it. That text can come from the surrounding Plugin, while the license and notice files in scope still miss a copyright holder required by the copied Skill.

Even where composition preserves the compliance payload, the files that carry it usually sit outside the bundled component. Of the 611 Skill copies with a complete compliance payload, only 20 carry it inside the Skill folder. The other 591 (96.7%) depend on files above it, including 370 at the Plugin repository root. None of the 18 bundled MCP servers with a complete compliance payload carries it inside the component folder.

Composition integrity therefore often depends on the surrounding Plugin rather than the bundled component itself. A Plugin can be complete at repository scope while the component remains dependent on files above its own folder. Whether that compliance payload reaches users depends on which directories distribution includes, which we test in C4.

4.4 C4: Distribution often leaves behind compliance payloads stored above the delivered directory

Plugin installation leaves behind most compliance payloads stored above the selected directory. Before installation, 229 of 257 Skill deliveries have a complete compliance payload in their Plugin

repository; after Claude Code installs the marketplace-selected directory, only 56 remain complete and 173 (75.5%) lose both matching license text and required copyright holders (Figure 3, C4). Every applicable license and notice file already inside the selected directory survives byte-for-byte, locating the loss at directory selection rather than at copying.

Claude Code and Codex both leave behind license and notice files above the selected directory, while bundled MCP servers stay complete. Of 221 repository-complete Skill deliveries, 52 remain complete after Codex installation and 169 (76.5%) lose both the license text and the required copyright holders. Claude Code and Codex agree on the compliance payload outcome of all 572 bundled-component installations they both complete. All repository-complete bundled MCP servers remain complete, 16 of 16 on Claude Code and 12 of 12 on Codex, because their selected directories contain the required files.

The observed losses are concentrated, but many more Skills and Plugins keep their compliance payload above the directory an installer may copy. Two Skill collections account for 168 of Claude Code's 173 losses, so the measured rate is not ecosystem-wide. Yet 24,579 of 26,343 permissively labeled Skills with a complete compliance payload in the 4,671 Plugins with a complete root compliance payload (93.3%) depend on files above the Skill folder. The Plugins' own compliance payloads are stored the same way. Of 13,275 distinct selected-directory/license combinations, 10,253 (77.2%) rely on license text or a copyright holder above the selected directory. Whether these compliance payloads are lost therefore depends on which directory each marketplace entry distributes (Appendix K).

Registry packages for MCP servers also omit license text stored above the package directory. Of 349 registry references mapped to a source repository with a complete compliance payload, 75 (21.5%) deliver an archive without it, covering 22 of 121 distinct archives (18.2%). None of those 22 archives contains its own recognized license or notice file. For 19, the matching source license text sits only above the package directory, while for three it sits inside the source package but is still omitted from the archive. Separately, four published npm archives point to a `LICENSE` file that they do not contain (Appendix L).

Together with C3, these results show that repository-level composition integrity does not guarantee distribution integrity. A complete compliance payload reaches users only when the delivered directory or package includes the files that carry it.

5 Discussion

A compliant repository does not make a compliant copy. Unlike a fork or a registry package, a Skill folder or a marketplace-selected directory is smaller than the repository that licenses it, and neither the Agent Skills specification nor Plugin manifests require the license text and holder inside it (Agent Skills, 2025; Anthropic, 2026e; OpenAI, 2026b). Ordinary reuse can therefore separate an unchanged component from notices that remain intact in its repository, and code file copies lose the compliance payload at least as often as Skill copies. Rule and context files copied between repositories (Jiang & Nam, 2026; Chatlatanagulchai et al., 2026) are likely exposed in the same way. The repository can pass an audit while the recipient holds a labeled copy without what the label requires them to pass on. Nor does a later repair help existing copies, because Skill reuse is largely a one-time copy, and 53% of reused Skills are never modified after adoption (Gao et al., 2026).

The notice condition attaches to every copy of an agent component. Skills extend a pattern in which license declarations outpace the files behind them. Dataset audits find omitted and miscategorized licenses (Longpre et al., 2024; 2025), and permissively labeled datasets and models carry the license text and holder in only 2.3% and 3.2% of cases (Jewitt et al., 2026b). Unlike a dataset or model, an agent component is copied and redistributed, the acts at which courts have enforced notice conditions (Jacobsen v. Katzer, 2008; Artifex Software, Inc. v. Hancom, Inc., 2017). Prevention therefore has to begin at publication. A downstream check cannot infer a copyright holder from a license label, so the 59.4% of Skills published without a complete compliance payload need it from their publishers before the first copy. Requiring the bundled license file that the specification already permits (Agent Skills, 2025), with the notices the declared license requires, would close this gap without a new licensing mechanism.

Delivery is where the loss is decided, so delivery is where to check. Claude Code, Codex, and package registries act as distribution intermediaries whose fixed delivery units decide the outcome for every recipient (Anthropic, 2026e; OpenAI, 2026b; Model Context Protocol, 2026a). The same Skill at the same commit arrives with or without its compliance payload solely because of which entry selects it, on both hosts (Appendix M). Hosts can therefore compare an entry's selected directory with the component's governing chain before installation (Anthropic, 2026a), and registries can inspect the archive they publish rather than the source tree, as PEP 639 does for declared license files (Ombredanne et al., 2024). The check must also be component-specific, because what disappears is often the copyright holder. Of the 89 incomplete bundled Skills, 79 still carry matching license text, which the surrounding Plugin's own license can supply, but that license cannot substitute for the constituent Skill's notices. Whatever license an ecosystem adopts, its declaration must remain bound to its compliance payload along every route that copies it.

6 Related work

License and provenance audits in AI. Audits of datasets and models find missing, inconsistent, or weakly documented licenses and provenance across AI supply chains (Longpre et al., 2024; 2025; Pepe et al., 2024; Stalnaker et al., 2026). Jewitt et al. (2026a) trace how license obligations change across dataset–model–application chains, while Jewitt et al. (2026b) ask whether permissive labels are backed by the license text and copyright holders they require. Both operate at artifact or repository scope. Agent studies instead show widespread Skill copying and adaptation (Destefanis et al., 2026; Gao et al., 2026; Zhu et al., 2026) and recover provenance across reused components (Chen et al., 2026). We connect these lines by following the compliance payload through the smaller units that agents copy, bundle, and deliver.

License compliance across copies and deliverables. Software research shows that copied code can lose required attribution (Baltes & Diehl, 2019) and that a deliverable's compliance state can differ from its source tree (van der Burg et al., 2014). REUSE and PEP 639 similarly place licensing information closer to files and packages (Free Software Foundation Europe, 2026; Ombredanne et al., 2024). We study the corresponding boundary in agent ecosystems, where notices can remain in the repository while falling outside the directory or package actually copied and delivered.

7 Conclusion

Permissive washing in agent supply chains is not only a publication problem. A component can begin with a complete compliance payload and lose it when reuse or distribution copies a smaller unit that leaves the required license text or copyright holder behind. Across Skills, MCP servers, and Plugins, losses during file copying, bundling, installation, and package publication arise the same way, because the copied unit excludes the files that hold the compliance payload. Repository-level checks are therefore insufficient. Publishers must place the compliance payload with the component before it is copied, and distribution systems must verify the unit they actually deliver. Appendix A sets out the limits of these measurements.

Reproducibility statement

Section 3 defines the four checkpoints and the evidence rules used in the analyses. Appendices C, D, and F document the public data sources, population definitions, and exclusions; the subsequent computational appendices report the sensitivity, timing, installation, exposure, and registry analyses. The supplementary replication package contains the frozen study ledgers and installation receipts, data provenance in `DATA.md`, and instructions and code to regenerate the empirical tables and figures and check reported values against the paper. The frozen artifacts preserve the historical repository and package observations, which may differ from current public sources.

AI USE STATEMENT

We used generative AI tools in four ways. An AI coding agent helped implement and revise analysis scripts and run queries over the retained ledgers under author direction, and reconciliation scripts in the supplementary material recompute every reported count. The authors verified all code, computations, and results. AI assistants gave feedback on methodology, framing, and interpretation of completed results. AI assistants helped discover and summarize related work and legal sources; the authors checked every cited source. AI assistants drafted and revised manuscript text, figures, tables, and references from author-supplied results and checked the manuscript against those results; manuscript preparation involved no new experiments, and the authors reviewed the text and verified every number and citation. No AI system generated research data or labeled validation cases. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

AboutCode. ScanCode toolkit. https://github.com/aboutcode-org/scancode-toolkit, 2026.

Agent Skills. Agent skills specification. https://agentskills.io/specification, 2025.

Anthropic. Create and distribute a plugin marketplace. Claude Code documentation, https://code.claude.com/docs/en/plugin-marketplaces, 2026a.

Anthropic. Agent skills. Claude Platform documentation, https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview, 2026b.

Anthropic. Use skills in Claude. Claude Help Center, https://support.claude.com/en/articles/12512180-use-skills-in-claude, 2026c.

Anthropic. Plugin loading reference. Claude Code documentation, https://code.claude.com/docs/en/plugins/loading, 2026d.

Anthropic. Claude code plugins reference. https://code.claude.com/docs/en/plugins-reference, 2026e.

Apache Software Foundation. Apache license, version 2.0. https://www.apache.org/licenses/LICENSE-2.0, 2004.

Artifex Software, Inc. v. Hancom, Inc. No. 16-cv-06982-JSC. U.S. District Court, N.D. Cal., order of 25 April 2017, 2017. URL https://cases.justia.com/federal/district-courts/california/candce/3%3A2016cv06982/305835/32/0.pdf.

Sebastian Baltes and Stephan Diehl. Usage and attribution of Stack Overflow code snippets in GitHub projects. *Empirical Software Engineering*, 24:1259–1295, 2019. doi: 10.1007/s10664-018-9650-5. URL https://doi.org/10.1007/s10664-018-9650-5.

Bartz v. Anthropic PBC. No. 3:24-cv-05417-WHA. U.S. District Court, N.D. Cal., order of 23 June 2025, 2025. URL https://www.leagle.com/decision/infdco20250625c57.

Worawalan Chatlatanagulchai, Hao Li, Yutaro Kashiwa, Brittany Reid, Kundjanasith Thonglek, Pattara Leelaprute, Arnon Rungsawang, Bundit Manaskasemsak, Bram Adams, Ahmed E. Hassan, and Hajimu Iida. Agent READMEs: An empirical study of context files for agentic coding. *ACM Transactions on Software Engineering and Methodology*, 2026. doi: 10.1145/3840295. URL https://doi.org/10.1145/3840295. Preprint available at https://arxiv.org/abs/2511.12884.

Jialuo Chen, Minghe Wang, Lingqi Jiang, Jianan Ma, Xinhao Deng, Xiaohu Du, Ruixiao Lin, Yunhao Feng, Linkang Du, and Jingyi Wang. SkillTrace: Multi-trace provenance auditing for LLM-agent skill reuse. arXiv:2608.05204, 2026. URL https://arxiv.org/abs/2608.05204.

Giuseppe Destefanis, Daniel Graziotin, Matteo Vaccargiu, and Marco Ortu. GitSkills: A dataset of agent skills on GitHub. arXiv preprint arXiv:2608.10906, 2026. URL https://arxiv.org/abs/2608.10906.

Entr'Ouvert v. Orange. Judgment of the cour d'appel de paris. Paris Court of Appeal, 14 February 2024, 2024. URL https://www.courdecassation.fr/decision/65cdbcdf2425a70008258563.

Free Software Foundation Europe. REUSE specification, version 3.3. https://reuse.software/spec-3.3/, 2026.

Haoyu Gao, Jai Lal Lulla, Hong Yi Lin, Sebastian Baltes, Christoph Treude, and Mansooreh Zahedi. From registry to repository: How AI agent skills are written, adapted, and maintained. *arXiv preprint arXiv:2607.00911*, 2026. URL https://arxiv.org/abs/2607.00911.

GitHub. About forks. GitHub Docs, https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/about-forks, 2026.

Mohammed Mehedi Hasan, Hao Li, Emad Fallahzadeh, Gopi Krishnan Rajbahadur, Bram Adams, and Ahmed E. Hassan. Model Context Protocol (MCP) at first glance: Studying the security and maintainability of MCP servers. *ACM Transactions on Software Engineering and Methodology*, 2026. doi: 10.1145/3814959. URL https://doi.org/10.1145/3814959. Preprint available at https://arxiv.org/abs/2506.13538.

Ahmed Hereiz, Yingzhe Lyu, Hao Li, Bram Adams, and Ahmed E. Hassan. On the maintenance and co-evolution of agent plugins: An empirical study of Claude Code plugin marketplaces. arXiv:2608.28497, 2026. URL https://arxiv.org/abs/2608.28497.

Jacobsen v. Katzer. 535 F.3d 1373. U.S. Court of Appeals for the Federal Circuit, 2008. URL https://openjurist.org/535/f3d/1373/jacobsen-v-katzer.

Jacobsen v. Katzer. Stipulated permanent injunction, no. 3:06-cv-01905-jsw. U.S. District Court, N.D. Cal., 22 February 2010, 2010. URL https://docs.justia.com/cases/federal/district-courts/california/candce/3%3A2006cv01905/179727/403.

James Jewitt, Hao Li, Gopi Krishnan Rajbahadur, Bram Adams, and Ahmed E. Hassan. Don't trust the label: License laundering in AI supply chains. *arXiv preprint arXiv:2607.20300*, 2026a. URL https://arxiv.org/abs/2607.20300.

James Jewitt, Gopi Krishnan Rajbahadur, Hao Li, Bram Adams, and Ahmed E. Hassan. Permissive-washing in the open AI supply chain: A large-scale audit of license integrity. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 2014–2025, 2026b. doi: 10.1145/3770855.3818130. URL https://doi.org/10.1145/3770855.3818130. Preprint available at https://arxiv.org/abs/2602.08816.

Shaokang Jiang and Daye Nam. Beyond the prompt: An empirical study of cursor rules. In *International Conference on Mining Software Repositories*, 2026. doi: 10.1145/3793302.3793367. URL https://doi.org/10.1145/3793302.3793367.

Kadrey v. Meta Platforms, Inc. No. 3:23-cv-03417-VC. U.S. District Court, N.D. Cal., order of 25 June 2025, 2025. URL https://zenodo.org/records/15847228.

Zhiwei Lin, Bonan Ruan, Jiahao Liu, and Weibo Zhao. A large-scale evolvable dataset for Model Context Protocol ecosystem and security analysis. In *IEEE/ACM International Conference on Automated Software Engineering, Tool Demonstrations*, pp. 3985–3988, 2025. doi: 10.1109/ase63991.2025.00356. URL https://doi.org/10.1109/ase63991.2025.00356. arXiv:2506.23474.

Shayne Longpre, Robert Mahari, Anthony Chen, et al. A large-scale audit of dataset licensing and attribution in AI. *Nature Machine Intelligence*, 6:975–987, 2024. doi: 10.1038/s42256-024-00878-8. URL https://doi.org/10.1038/s42256-024-00878-8.

Shayne Longpre, Nikhil Singh, Manuel Cherep, et al. Bridging the data provenance gap across text, speech, and video. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/97b983c974551153d20ddfabb62a5203-Abstract-Conference.html.

Model Context Protocol. About the MCP registry. https://modelcontextprotocol.io/registry/about, 2026a.

Model Context Protocol. What is the Model Context Protocol (MCP)? Specification version 2026-07-28, https://modelcontextprotocol.io, 2026b.

npm. package.json. npm Docs, https://docs.npmjs.com/cli/v11/configuring-npm/package-json/, 2026.

Philippe Ombredanne, C.A.M. Gerlach, and Karolina Surma. PEP 639: Improving license clarity with better package metadata. Python Enhancement Proposal, https://peps.python.org/pep-0639/, 2024.

Open Source Initiative. The 3-clause BSD license. https://opensource.org/license/bsd-3-clause, 2026a.

Open Source Initiative. The MIT license. https://opensource.org/license/mit, 2026b.

OpenAI. Build skills. Codex documentation, https://developers.openai.com/codex/skills, 2026a.

OpenAI. Package your plugin. OpenAI Developers documentation, https://developers.openai.com/plugins/build/plugins, 2026b.

OpenAI. Skills. OpenAI API documentation, https://developers.openai.com/api/docs/guides/tools-skills, 2026c.

Federica Pepe, Vittoria Nardone, Antonio Mastropaolo, Gabriele Bavota, Gerardo Canfora, and Massimiliano Di Penta. How do Hugging Face models document datasets, bias, and licenses? An empirical study. In *International Conference on Program Comprehension*, 2024. doi: 10.1145/3643916.3644412. URL https://doi.org/10.1145/3643916.3644412.

Trevor Stalnaker, Nathan Wintersgill, Oscar Chaparro, Laura A. Heymann, Massimiliano Di Penta, Daniel M. German, and Denys Poshyvanyk. An empirical analysis of machine learning model and dataset documentation, supply chain, and licensing challenges on Hugging Face. *ACM Transactions on Software Engineering and Methodology*, 35(8):1–49, 2026. doi: 10.1145/3776739. URL https://doi.org/10.1145/3776739. Preprint available at https://arxiv.org/abs/2502.04484.

Benny Toeppe, Amine Barrak, and Emna Ksontini. A large-scale dataset of MCP implementations on GitHub. In *Proceedings of the 23rd International Conference on Mining Software Repositories*, pp. 610–614. ACM, 2026. doi: 10.1145/3793302.3793311. URL https://doi.org/10.1145/3793302.3793311.

U.S. Copyright Office. Copyright notice. Circular 3, https://www.copyright.gov/circs/circ03.pdf, 2021.

Sander van der Burg, Eelco Dolstra, Shane McIntosh, Julius Davies, Daniel M. German, and Armijn Hemel. Tracing software build processes to uncover license compliance inconsistencies. In *IEEE/ACM International Conference on Automated Software Engineering*, 2014. doi: 10.1145/2642937.2643013. URL https://doi.org/10.1145/2642937.2643013.

Vercel. find-skills. https://skills.sh/vercel-labs/skills/find-skills, 2026. Accessed September 26, 2026. Displayed installation count: 3.6 million.

Welte v. D-Link Deutschland GmbH. Judgment, docket no. 2-6 o 224/06. Landgericht Frankfurt am Main, 6 September 2006, 2006. URL https://gpl-violations.org/news/20061110-dlink-judgement_frankfurt/.

Jiaying Zhu, Lyuye Zhang, Wenbo Guo, and Yang Liu. Latent reuse in agent skills: Multi-modal clone detection at ecosystem scale. In *IEEE/ACM International Conference on Automated Software Engineering*, 2026. URL https://arxiv.org/abs/2603.22447. arXiv:2603.22447; ASE 2026, as reported by the authors.