kymata labs

RESEARCH PAPER 02 / AI & OWNERSHIP

You Built It With AI. Do You Own It?

A subscription receipt is not an ownership certificate. The questions an AI-built business still has to answer.



Kymata Labs Research
2026-11-15 · Edition 1.0

Independent analysis
kymatalabs.com

Small studio. Open worlds.

THE LONGER VIEW

Executive summary.

A product can work perfectly while its account of ownership remains incomplete. This paper separates three questions that are often collapsed into one: what an AI supplier agrees to, what human expression may receive copyright protection, and what obligations accompany material originating elsewhere. It reads specific GitHub agreements, the U.S. Copyright Office's January 2025 report and actual software license texts. A hypothetical software transaction makes the distinction concrete, while a four-record framework connects a business promise to the evidence behind it. The conclusion rejects both easy extremes: AI assistance does not automatically eliminate protection for human work, and a supplier declining to claim an output does not settle every right in it. The analysis is U.S.-focused, dated and qualified; it is not a legal opinion on any particular product.

> U.S.-focused analysis of inspected supplier agreements, official product documentation and license texts. The January 2025 Copyright Office report was read through a complete third-party republication; official-host bytes could not be authenticated. Its dated conclusions are not a survey of subsequent law. The business examples and review framework are illustrative, not legal advice. Source access: November 16, 2026 UTC.

01 / One product, three different questions
02 / Read the promise at the right level
03 / Copyright follows human expression, not the subscription
04 / The strongest countercase belongs in the argument
05 / An open-source license is permission with a shape
06 / A missing alert is not a complete history
07 / Build a record that answers the promise
08 / Return to the founder's repository
09 / Ownership should be a claim you can explain
10 / Sources, method and limits
Q&A / Questions worth asking
Sources / References

© 2026 Kymata Labs LLC. All rights reserved.
Read the web edition: kymatalabs.com/research/you-built-it-with-ai-do-you-own-it

Imagine a founder preparing to sell a small software business. The product works. Customers pay for it. An AI assistant helped write the onboarding screens, database migrations, billing logic and internal documentation. The founder has the repository, the deployment credentials and a subscription receipt. When a buyer asks whether the company owns the software, the answer seems obvious.

Then the question becomes more precise. Which agreement covered the generated code? What did people actually write or change? Were any suggestions copied from existing projects? Which notices must accompany a distribution? Can the company support an exclusive ownership promise about every component?

This is an illustrative transaction, not a reported acquisition. Its point is that the word *own* can carry several different promises while sounding like one simple fact.

A supplier can decline to claim your output without establishing that every line is protected by copyright. Human contributions can receive protection even when a larger work includes AI-generated material. And permission from an AI supplier does not, by itself, settle obligations associated with someone else's code. These propositions are compatible. Treating them as contradictions is what makes the ownership debate so confusing.

The practical risk is therefore broader than discovering that a machine-generated passage may lack copyright protection. It is building a business whose promises are more definite than its records. A product can function perfectly while its account of where the code came from remains incomplete.

The useful response is neither to declare AI-built software worthless nor to accept “you own the output” as the end of the inquiry. It is to make the claim specific enough to examine. This paper uses one supplier's inspected agreements, a dated U.S. Copyright Office report and actual license texts to show how. It offers an evidence framework, not a legal opinion on a particular codebase.

One product, three different questions

The first question concerns the relationship with the supplier. Does its contract claim ownership of inputs or outputs? What responsibilities does it assign to the customer? Which account, purchase channel and agreement apply? These are questions about the bargain between identified parties.

The second concerns copyrightability. What original expression did a human contribute, and what protection might attach to that contribution? A contract's treatment of output cannot substitute for that factual inquiry. A supplier's decision not to assert ownership is not an assessment of how a particular work was authored.

The third concerns material originating elsewhere. Does the output contain code for which someone else holds rights? If a license is needed, what does it permit and require? These questions can remain relevant even where the new, machine-generated portions receive no protection of their own.

Consider a single module containing a developer's original implementation, a generated helper function and an intentionally incorporated open-source library. Asking who owns “the module” conceals distinctions between its parts. The developer's contribution, the helper's authorship and the library's

license each need their own account. Combining them into one file does not make the questions identical.

This does not mean every file requires a legal essay. It means a team should know which claim a piece of evidence supports. A purchase receipt identifies a commercial relationship. A version history can show changes. A license file records permissions and conditions. None is a universal certificate for the other two questions.

FIGURE 01 RIGHTS ARE NOT INTERCHANGEABLE

One product. Three different questions.

A supplier's promise, a human contribution and an incoming license answer different questions about the same product.

01

Permission

Supplier contract

What did this supplier agree to? Preserve the agreement, account, purchase channel and date.

02

Protection

Human authorship

What expression did people contribute? Preserve authored material and substantive changes.

03

Provenance

Third-party material

Which permissions and conditions apply? Preserve sources, license texts and required notices.

One answer does not settle the others. This framework does not determine rights in a particular work.

GitHub terms, the Copyright Office's January 2025 report, and inspected license texts. [1] [2] [4] [5] [6] [7]

Read the promise at the right level

GitHub provides a particularly clear example because its inspected terms separate these issues in adjacent passages. Its Terms of Service, carrying an effective date of April 27, 2026, state: "GitHub does not claim ownership of your Input or Output." The next paragraph says outputs may contain material subject to third-party copyrights or open-source license terms, and makes the user responsible for determining whether a third-party license is required and complying with it.[1]

That pairing matters. The first sentence describes GitHub's ownership position. The second prevents a reader from treating that position as a promise that no other rights are involved. The same section says output may be inaccurate, incomplete or nonfunctional, and requires users to review, test and validate it before use.[1] Functional acceptance and rights review therefore address different parts of the supplier's own stated limitations.

Purchase channel also matters. GitHub's Generative AI Services Terms, version March 2026, apply to purchases directly from GitHub under a volume licensing agreement. They expressly exclude personal customers and customers purchasing through Microsoft, directing those groups to different terms. Their ownership provision says: “GitHub does not own Inputs or Outputs. You retain any ownership you already have in your Inputs.”[2]

The wording is narrower than an assurance that every output creates a new, enforceable intellectual property right for the customer. It also places responsibility for resulting applications or agents, including applicable licensing requirements, on the customer. Those are contractual statements from a named supplier, not findings about all AI products.[2]

Version control belongs in this inquiry too. The older GitHub Copilot Product Specific Terms, version October 2024, now carry an explicit notice that they were deprecated effective March 5, 2026 for new subscriptions and renewals from that date.[10] A recognizable product name and a search result are insufficient to identify the applicable bargain.

Nor should a defense commitment become shorthand for guaranteed clearance. The March 2026 AI terms say that *if* the customer's agreement provides for defense of third-party claims, that provision applies to outputs. The inspected General Terms separately describe notice, cooperation and control-of-defense conditions.[2][3] Determining the coverage of a particular dispute requires the actual agreement and facts. A contractual mechanism for responding to a claim is a different thing from a conclusion that no claim can arise.

The practical task is modest: preserve the applicable agreement, date, purchasing entity and relevant configuration with the project's records. A screenshot of a reassuring sentence is less useful than the document that explains its scope.

Copyright follows human expression, not the subscription

The U.S. Copyright Office's January 2025 report on copyrightability supplies the central distinction. Its executive summary says copyright protects original expression created by a human author even when the work also includes AI-generated material. It also says protection does not extend to purely AI-generated material or material with insufficient human control over expressive elements. Whether a contribution is sufficient must be assessed case by case.[4] (p. iii)

These are the Office's dated conclusions, not a ruling about the hypothetical founder's software. They are nevertheless a useful corrective to two opposite shortcuts: that an AI subscription delivers copyright automatically, and that any AI involvement eliminates it.

The report emphasizes how a system is used. Assistive uses that enhance human expression do not limit protection; uses in which the system makes expressive choices need further analysis.[4] (p. 12) “AI-assisted” is therefore a description requiring detail, not a complete account of authorship.

Prompting illustrates the distinction. The Office concluded that, given generally available technology at the time, prompts alone did not provide sufficient human control to make the user the author of the output. It also rejected effort itself as the test: revising prompts repeatedly can require time and skill without establishing authorship of the resulting expression.[4] (pp. 18–21)

The qualification matters. A January 2025 assessment of then-available technology should not be rewritten as a timeless assertion about every interface or future system. The report distinguishes potentially copyrightable prompts from the output they request and acknowledges that greater human control could change the analysis.[4] (pp. 13, 21) Neither prompt length nor prompt count supplies a universal threshold.

A concrete example in the report makes the layered result easier to understand. In its discussion of *Rose Enigma*, the Office describes a human drawing used as an input and identifies human expressive elements that remained visible in the result. It quotes a registration annotation limiting the registration to "unaltered human pictorial authorship" that was perceptible and separable from the excluded non-human expression.[4] (p. 23)

We inspected the report's account, not the underlying registration record. The example concerns visual art, not software. Its relevance here is conceptual: a mixed work need not receive an all-or-nothing answer. Human expression can remain identifiable inside something a machine also helped produce.

The report likewise recognizes sufficiently creative human selection, coordination, arrangement and modification. Protection would extend to the human contribution, rather than automatically to underlying generated material.[4] (pp. 24–25) Simply choosing a pleasing output is not the same thing as making a sufficiently creative arrangement. Nor does this evidence establish that renaming variables, accepting suggestions or running tests necessarily meets the authorship standard.

For a development team, the useful question becomes concrete: what expression did people determine and contribute? A factual record of those decisions is more informative than attaching a percentage of "human code" to the repository. The report provides no universal percentage formula.

The strongest countercase belongs in the argument

There is a serious objection to an ownership paper that concentrates only on uncertainty: human creators have always used tools, collaborators and existing components. A software product should not be presumed legally hollow because its developers used an assistant.

The Office's report supports this objection directly: "The use of AI tools to assist rather than stand in for human creativity does not affect the availability of copyright protection for the output."[4] (p. iii) It also says that including AI-generated elements in a larger human-authored work does not affect the copyrightability of that larger work as a whole, while illustrating the point with a film containing generated effects or artwork.[4] (p. 27)

Applied cautiously to our software example, that distinction leaves room for substantial human-authored material in a product that also contains generated parts. It does not establish protection for any specific function, screen or application. It does explain why “you used AI, therefore you own nothing” is a poor starting point.

The commercial countercase is equally important. A business may compete through dependable service, customer knowledge, integration, support and continued improvement. Exclusive control over every implementation detail need not be its only source of value. This is a business argument, not a measured valuation claim.

That objection changes the prescription. The aim should be a truthful account of the asset, not an attempt to manufacture purity. Teams can decide which expressive contributions matter, which third-party components they are comfortable using and which promises they can support. Sometimes licensed reuse will be a better choice than attempting to make every part exclusive.

Uncertainty deserves attention where it affects a real decision: a distribution, a customer warranty, a licensing strategy or an acquisition representation. It should not become a reason to treat all AI assistance as disqualifying.

An open-source license is permission with a shape

The third-party question predates generative AI. The new interface can obscure it by delivering code as an answer rather than as a file accompanied by a recognizable project and license. The legal instrument, where applicable, still deserves to be read.

The MIT license is short and permissive. It grants rights including use, copying, modification, distribution, sublicensing and sale, subject to conditions. Its notice requirement says the copyright and permission notices must be included in “all copies or substantial portions of the Software.”[5] “Permissive” therefore does not mean “no conditions.” Equally, those conditions should not be inflated into a claim that commercial use is forbidden.

Apache License 2.0 makes the structure more explicit. It grants copyright permissions and, separately, a patent license within stated bounds. Its redistribution conditions include supplying the license, marking changed files, retaining specified notices, and carrying applicable attribution notices from a NOTICE file when one exists.[6] (sections 2–4) These requirements illustrate why a license identifier alone is insufficient for a release process: the relevant files and notices matter too.

GNU GPL version 3 introduces different conditions for conveying covered works. Its provisions distinguish verbatim source copies, modified source versions and non-source forms; the latter involve Corresponding Source requirements. For modified source versions, section 5 includes licensing the covered work as a whole under the GPL, while separately explaining that an aggregate of independent works does not automatically bring its other parts under the license.[7] (sections 4–6)

The distinctions prevent two misleading shortcuts. One is assuming that every license is equivalent to a simple attribution request. The other is assuming that any contact with GPL software requires publication of everything a business owns. The text expressly permits making and running covered works without conveying them, subject to its stated conditions. It also says mere network interaction without transferring a copy is not conveying.[7] (sections 0, 2) Other licenses may make different choices; this is not a universal rule for network services.

Another useful boundary appears in GPL section 2: output from running a covered work is covered only if its content constitutes a covered work. That does not resolve questions about AI training or any particular generated suggestion. It does show why a tool's license and the legal status of everything produced using the tool should not casually be equated.[7]

For a concrete repository example, the Go project's license permits redistribution and use in source and binary forms with conditions. Source distributions must retain the notice, conditions and disclaimer; binary distributions must reproduce them in accompanying documentation or other materials.[8] Broad reuse rights and retained author notices coexist comfortably in ordinary software development.

None of these texts proves that a particular generated fragment copies protected material, requires a license or infringes a right. They show what obligations can look like when licensed material is actually involved. Whether a specific match matters requires more than spotting a familiar line.

A missing alert is not a complete history

GitHub's documentation describes code references that can help make this inquiry practical. References can identify matching public repositories and detected licenses. For inline suggestions, the documented comparison includes the potential suggestion and approximately 150 characters of surrounding code.[9]

The same documentation states important limits. Its public-code index excludes private repositories and code hosted outside GitHub. It is refreshed periodically, so recently added material may be absent and references may point to code that moved or disappeared. It also says that only accepted, unchanged inline suggestions are checked in the described way.[9]

Those boundaries determine what a missing reference means. It is not a record of every possible source, and it is not a legal judgment that the code is original or unrestricted. Conversely, a reference is a reason to investigate source and licensing; it does not independently establish infringement.

This is an argument for using the feature accurately, not abandoning it. A matching repository URL and license can turn an otherwise opaque suggestion into something reviewable. The problem begins when a bounded detector is asked to support an unbounded promise.

The ownership framework should therefore preserve both findings and limits. Record a relevant match and the disposition. Record the tool and surface that supplied the check. Avoid converting “nothing reported by this mechanism” into “nothing else could have rights.”

FIGURE 02 A DOCUMENTED COVERAGE BOUNDARY

A reference is a lead. Not a clearance.

GitHub's public-code references can identify matching repositories and detected licenses within a limited index.

01

Public GitHub

Within the described index

A match can identify a source to inspect. It is not independently a finding of infringement.

02

Private / external

Outside the index

Private repositories and code outside GitHub are excluded from the documented index.

03

Incomplete history

Between refreshes

New, moved or removed material can affect references. A missing alert is not complete provenance.

For inline suggestions, the described check covers accepted, unchanged suggestions and approximately 150 surrounding characters. Geometry does not measure coverage.

GitHub's documented behavior; not an independently tested detector. [9]

Build a record that answers the promise

The following framework is this paper's proposed working method. It has not been validated as a legal safe harbor. Its purpose is to connect a business claim to evidence while keeping unresolved questions visible.

Start with the supplier record. Identify the product, account owner, purchase channel, applicable agreement and agreement date. Keep material settings that affect the workflow with that record. This establishes what contract the team is discussing; it does not establish authorship or clear third-party rights.

Next, record the human contribution where it matters. Preserve significant drafts, authored source, changes and explanations of substantive choices. Identify who made those contributions and whether the company's relationship with those contributors needs review. This is a factual account, not a point-scoring exercise in which enough commits purchase copyright.

Then record incoming material. Keep dependency manifests, relevant source locations, license texts, notices and the disposition of meaningful code matches. Where a source or license is unknown, say so. A blank field should not silently become a clearance decision. Where a known component is acceptable, record the obligations that accompany its use.

Finally, connect those records to the intended promise. Shipping a hosted service, distributing a binary, offering a source license and making a contractual representation may raise different questions. Identify what is actually being delivered and asserted. Escalate uncertainty that would change that decision rather than treating every generated line as equally consequential.

This approach can be proportionate. A routine suggestion might need ordinary review. A large, distinctive implementation proposed for an exclusive source license deserves a more specific account. The trigger is the importance of the claim and the material, not an arbitrary quota of saved prompts.

The resulting review should allow several honest outcomes: documented permission with obligations assigned; human contribution identified but scope unresolved; a component needing further investigation; or a promise narrowed because the available evidence does not support it. These are more useful states than a single green “AI compliant” badge.

Records also have limits. A prompt archive can show instructions without proving authorship of the resulting expression. A successful test suite can show behavior without resolving provenance. A license scan can support a review without supplying a complete history. Keeping those distinctions explicit is the method's central discipline.

FIGURE 03 A PROPOSED REVIEW RECORD

Make the promise answerable.

Keep each business claim connected to the kind of evidence that can support it.

01

01

Supplier record

Which bargain applies? This does not establish authorship.

02

02

Human contribution

What did people determine? This does not clear incoming material.

03

03

Incoming material

What sources and conditions are known? This is not complete provenance.

04

04

Intended promise

What will the business deliver or assert? Keep unresolved legal questions visible.

Possible outcomes include documented obligations, unresolved scope, further investigation or a narrower promise. This is not a legal safe harbor.

Kymata Labs proposed working method, informed by the cited agreements and texts. [1] [2] [4] [5] [6] [7] [9]

Return to the founder's repository

Suppose the hypothetical company reviews three components before making its ownership representation.

The first is its scheduling logic. A developer wrote the initial implementation and substantial revisions; an assistant suggested tests and a small helper. The repository preserves those drafts and changes. This gives the reviewer a better account of human contribution than “built with AI,” although it does not determine protection for every element. The generated helper remains a separate question where its importance warrants one.

The second is a reporting feature that deliberately incorporates an MIT-licensed component. Here, uncertainty about provenance is low because the team knows the source. The practical issue is to retain the applicable notices and make a promise consistent with the license. The company should not describe known third-party material as exclusively its own merely because the final product is sold under its brand.

The third is a large generated parser whose source history is unclear. It passes functional tests. That supports a behavior claim, not the desired ownership representation. The team can investigate recognizable matches, seek a suitable known implementation, change its product promise or obtain advice on the unresolved issue. Replacing the parser is an engineering option, not a guarantee that a fresh generation is legally different.

These components belong to one working product, but the review produces three different accounts. That is a strength. The alternative is an overbroad sentence that conceals what the team knows and what it does not.

Notice also what the method does not demand: a claim that every generated contribution is risky, that every similarity requires a license, or that extensive human editing automatically cures any problem. Each of those statements would outrun the evidence.

The company emerges with something more useful than reassurance. It knows which supplier bargain applies, where its human work is documented, which permissions it relies on and which representations need to be qualified. Those records will not make every legal question disappear. They make the next question answerable.

Ownership should be a claim you can explain

AI makes it easier to produce working software without receiving the familiar documentary trail that accompanied a borrowed library or a colleague's contribution. The interface delivers an answer. The business still has to decide what it can promise about that answer when it becomes part of a product.

The evidence here does not support a universal verdict that AI-built products are unownable. The Copyright Office's report expressly preserves protection for human expression and recognizes mixed works. Nor does it support treating a supplier's refusal to claim outputs as a universal title guarantee. GitHub's own terms place third-party obligations beside that refusal.

The durable distinction is between having a file, having permission to use material, and being able to support the particular rights one claims. A founder does not need a frightening theory about every line of code. The founder needs a truthful account of the important parts.

“We built it with AI” describes a tool choice. “Here is what our people contributed, the agreement we used, the licensed material we included and the questions we have resolved” describes an asset a business can examine.

Sources, method and limits

Research was conducted on November 16, 2026 UTC (November 15 in Los Angeles). We inspected GitHub's official policy repository, official customer-agreement pages, official documentation source, SPDX-hosted license texts and the Go repository's license. Repository documents were captured with commit identifiers; dynamic pages and files were captured with retrieval dates and hashes. These captures support the quoted provisions, not a promise that the same terms govern every customer or remain unchanged.

The January 2025 U.S. Copyright Office report was read through a complete third-party GitHub republication. The official reference URL is supplied with the accessed copy in reference [4]. Direct access to the official host was blocked, so we could not compare the republication's bytes with the official-host file. Its institutional identification and metadata were internally consistent; that is not independent authentication.

The analysis is U.S.-focused and uses that report's dated conclusions. It is not a current survey of every court ruling, a review of AI-training legality or legal advice. We did not establish subsequent case developments, a rate of infringement, insurance coverage, acquisition discounts or a universal authorship threshold. The transaction, components and review framework are explicitly illustrative.

Questions worth asking.

Do I own code an AI assistant writes?

That question combines supplier terms, copyrightability and third-party rights. The inspected GitHub terms disclaim supplier ownership while separately assigning responsibility for applicable third-party licenses. They do not decide every rights question about a particular output. [1][2]

Can an AI-assisted application receive copyright protection?

The Copyright Office’s January 2025 report says human expression can receive protection within work containing generated material. Sufficiency and scope remain case-specific. It does not establish that every generated component is protected. [4]

Does extensive prompting guarantee authorship?

No such guarantee appears in the inspected report. Its conclusion about then-available technology says prompts alone did not provide sufficient control over outputs. Effort and prompt count are not universal authorship tests. [4]

Does a clean code-reference result settle licensing?

No. GitHub documents limits to its index and the suggestions checked. A missing reference is not complete provenance or a legal clearance decision. A detected match also requires interpretation; it does not itself prove infringement. [9]

What should a small team preserve first?

Keep the applicable supplier agreement, significant human drafts and changes, and known third-party licenses and notices. Connect those records to the promises the business intends to make. This is the paper's practical recommendation, not a guarantee of legal sufficiency.

SOURCES, NOT SCENERY

References.

1. **GitHub Terms of Service, Section J: AI Features, Training, and Your Data**
   GitHub · Effective April 27, 2026 · Primary vendor terms, official policy repository

   Inspected November 16, 2026 at pinned repository commit b9578b5. Section J.2 distinguishes supplier ownership from third-party licensing; J.4 and J.5 describe review and responsibility. Repository document, not proof of the agreement governing any particular customer.

   https://github.com/github/site-policy/blob/b9578b546d2506febda1da2cd7431644d58e512c/Policies/github-terms/github-terms-of-service.md

2. **GitHub Generative AI Services Terms**
   GitHub · March 2026 version · Primary vendor terms, official customer-agreement page

   Inspected November 16, 2026. Applies to direct GitHub purchases under volume licensing; excludes personal and Microsoft purchasing channels. Sections 1, 2, 4, 5. Replaces pre-March 5, 2026 generative AI product terms.

   https://github.com/customer-terms/github-generative-ai-services-terms

3. **GitHub Customer Agreement: General Terms**
   GitHub · March 2025 version · Primary vendor terms, official customer-agreement page

   Inspected November 16, 2026. Sections 6.1–6.3 describe defense of third-party claims and conditions. No conclusion about coverage of a particular generated-code dispute; actual agreement and facts matter.

   https://github.com/customer-terms/general-terms

4. **Copyright and Artificial Intelligence, Part 2: Copyrightability**
   U.S. Copyright Office · January 2025 · Government report, inspected through third-party republication

   January 2025 report, printed pages iii, 12–13, 18–25, 27 and 41. Inspected through the linked complete third-party republication; official-host bytes could not be independently authenticated. A dated Office report, not a court ruling or an update on subsequent law.

   https://www.copyright.gov/ai/Copyright-and-Artificial-Intelligence-Part-2-Copyrightability-Report.pdf
   Accessed copy: https://raw.githubusercontent.com/ai-in-pm/Copyright-and-Artificial-Intelligence-Part-2-Copyrightability-Report/857b2b12fa7e542a48f737ad2e450c5f7f4b3f34/Copyright-and-Artificial-Intelligence-Part-2-Copyrightability-Report.pdf

5. **MIT License**
   SPDX License List · Undated license text; retrieved November 16, 2026 · Standard license text, SPDX reproduction

   Permission grant, notice condition and warranty disclaimer inspected. This generic text does not establish the license of any particular generated fragment.

   https://raw.githubusercontent.com/spdx/license-list-data/main/text/MIT.txt

6. **Apache License, Version 2.0**
   Apache Software Foundation; text reproduced by SPDX · January 2004 license version · Standard license text, SPDX reproduction

   Retrieved November 16, 2026. Sections 2–4 contain copyright/patent grants and redistribution conditions. NOTICE requirements are conditional on the Work including a NOTICE text file. No determination that a particular output is covered.

   https://raw.githubusercontent.com/spdx/license-list-data/main/text/Apache-2.0.txt

7. **GNU General Public License, Version 3**
   Free Software Foundation; text reproduced by SPDX · June 29, 2007 license version · Standard license text, SPDX reproduction

   Retrieved November 16, 2026. Sections 0, 2, 4–6 distinguish conveyance, private use, program output, aggregates and source requirements. Specific license and application matter; no conclusion about training legality or any particular AI output.

   https://raw.githubusercontent.com/spdx/license-list-data/main/text/GPL-3.0-only.txt

8. **The Go Project LICENSE**
   The Go Authors · Undated current file; retrieved November 16, 2026 · License in original software project repository

   File bears Copyright 2009 The Go Authors; 2009 is a copyright notice, not a verified publication date for this version. Source/binary redistribution notice conditions inspected. Concrete example only, not evidence that a generated suggestion contains Go code.

   https://raw.githubusercontent.com/golang/go/master/LICENSE

9. **GitHub Copilot in IDEs: References to matching public code**
   GitHub Docs · Undated document snapshot; retrieved November 16, 2026 · Primary vendor documentation, official source repository

   Pinned commit d59b263, repository HEAD November 13, 2026; HEAD date is not article publication date. Documents public GitHub index exclusions, periodic refresh, approximately 150 surrounding characters and accepted unchanged inline suggestions. Product description, not independent validation or a legal clearance method.

   https://github.com/github/docs/blob/d59b2631959db46d71ca5f7748ec04311c6d0870/content/copilot/concepts/copilot-surfaces/copilot-in-ides.md

10. **GitHub Copilot Product Specific Terms [Archive]**
    GitHub · October 2024 version; deprecation notice effective March 5, 2026 · Archived primary vendor terms

    Inspected November 16, 2026. Page says new subscriptions and renewals on/after March 5, 2026 are governed by Generative AI Services Terms. Cited only to illustrate version/applicability checks; not presented as the current general Copilot contract.

    https://github.com/customer-terms/github-copilot-product-specific-terms