
Who Authored the Spec? Evidence for the Line-Drawing Question Thaler Left Open

June Kim
Independent Researcher
june@june.kim
ORCID 0009-0005-3153-9396

July 6, 2026

Draft. Coding agents now produce working software from a written specification as the only human input, and no court or agency has said who, if anyone, authored the result. This essay argues everyone is asking at the wrong layer. Software’s authorship has always been analyzed upstream of mechanical transformation (no one asks whether the compiler authored the binary), and the agent moves the human artifact up one layer, from source code to the specification. Two consequences follow, both inside doctrine courts already apply. First, the operative line runs between a prompt, which conveys an unprotectable idea, and a spec, whose unforced selection of constraints can clear copyright’s low originality bar. The strongest doctrine against that claim presses hardest on exactly the specs that compile best. Second, the cases in between turn on evidence rather than new doctrine. The evidence is a dated log of the investigation that produced the change, showing how much the specification actually decided and who made the choices it did not. A shallow log means the problem statement was the work and its writer the author; where function forced the fix, it means there is no author at all. A deep log made by a machine running alone records choices no human may claim. That is the shape of the record the Supreme Court used to settle photography’s authorship in 1884, and the shape courts will need for the line-drawing question the [Thaler](#) cert denial left open: decided record by record, on who chose what.

1 Introduction

The Supreme Court’s March 2, 2026 [denial of certiorari](#) in *Thaler v. Perlmutter* (No. 25-449) settled less about AI authorship than the headlines said. It left in place the D.C. Circuit’s holding that “the Copyright Act of 1976 requires all eligible work to be authored in the first instance by a human being,” and the [opinion](#) is narrow: Dr. Thaler listed his machine as the work’s sole author, so the court decided only that a machine cannot be an author. It said so twice: the human authorship requirement “does not prohibit copyrighting work that was made by or with the assistance of artificial intelligence,” and “line-drawing disagreements over how much artificial intelligence contributed to a particular human author’s work are neither here nor there in this case.”

The line-drawing question describes a fast-growing share of the software now being written. An autonomous pipeline reads a GitHub issue, forms hypotheses about the failure, tests them against the codebase, discards the ones the evidence kills, implements the survivor, and opens a pull request. The code passes the automated test suite. The maintainer merges. [Techdirt](#) argues AI-generated code has no copyright at all; [Vorys](#) warns that vibe coding hollows out open-source licensing; [Bloomberg Law](#) walks firms through how tricky protection has become. All of it asks who authored the code.

The closest existing answer, from 1983, ratified a practice that assumes authorship sits upstream of the machine. In [Apple Computer v. Franklin](#), Franklin argued that copyright protected human-readable source code and stopped at machine-generated object code. The Third Circuit found “no basis in the statute for any such concern” and held that “a computer program, whether in object code or source code, is a ‘literary work’ and is protected from unauthorized copying, whether from its object or source code version.” That is a subject-matter holding, and I will not pretend it decided authorship. What it ratified is a practice with

an authorship assumption built in: source and object code are the same work (the opinion approvingly cites a case holding the object code version of registered source code “is the same work and protected”), and one work has one set of authors.

I am aware of no case in the forty years since that asks whether a compiler authored anything. I offer that as a practitioner’s report of an absence; no database survey stands behind it. That silence is the premise everything downstream rests on: an inference from an absence of litigation, with no holding behind it. The strongest competing explanation is that nobody argues a compiler is an author because the claim would be laughed out of the room. That is the premise restated. The reason it would be laughed out is that no one looks for authorship inside a mechanical transformation. Every choice in the binary was already made in the source, so no authorship is left to ask about. The expressive act sits upstream, and analysis follows it there without noticing the move.

A coding agent executing a specification extends that chain by one link. The extension is weaker than the original at one joint. A compiler produces the same output from the same source, every time; an agent is probabilistic, and two runs on the same spec can produce different code. But how different they are is a function of the prose. The more the specification determines (the interface typed, the acceptance test written, the approach fixed), the less remains for the machine to choose. Today’s inference stacks do not reliably offer bit-identical reruns. The argument does not need them. It needs the split: whatever the spec determined is invariant across runs and traces to the spec’s author, and whatever varies run to run is exactly the residue no spec-writer can claim. In practice the determined set is smaller than spec-writers believe. A spec I thought tight was corrected twenty times by its implementation; §6 puts that record on the table. For the determined set, whatever its size, the authorship analysis lands where Franklin’s practice put it: upstream. Who authored the spec?

I write software, not briefs. Every case discussed here is quoted from the primary opinion, and the verification trail is described in the disclosure at the end. The essay’s authority is the doctrine cited and the artifacts linked, and the reader should discount it exactly as much as that warrants.

2 Prompts as Ideas, Specifications as Expression

The upstream analysis splits into two that the current debate blurs.

Is the specification itself protectable expression? This requires no AI doctrine at all. A spec is a text a human wrote; it is analyzed as a literary work under principles that predate the transistor. [Section 102\(b\)](#) draws the floor: no protection for “any idea, procedure, process, system, method of operation, concept, principle, or discovery.” A prompt sits below that floor. “Make me a website” conveys a desire; so does “fix the crash on save.” The [Copyright Office’s January 2025 report](#) on copyrightability says this in one sentence: “Prompts essentially function as instructions that convey unprotectible ideas.”

[Feist v. Rural Telephone](#) draws the ceiling, and it is low. Originality requires only independent creation “plus a modicum of creativity,” and that creativity can live entirely in how unprotectable elements are combined: “choices as to selection and arrangement, so long as they are made independently by the compiler and entail a minimal degree of creativity, are sufficiently original” for protection. A specification is a selection-and-arrangement work in exactly Feist’s sense. An engineering spec that commits to a particular geometry for pricing ad auctions, a privacy guarantee enforced by the server’s hardware, a billing rule that charges only once per placement, and a one-toggle interface for the publisher combines four elements that nothing forced together. Each element alone is an unprotectable idea. The selection is where the author shows up.

That line, between instruction and unforced selection, is two centuries old. Nothing about AI moves it.

On the second, whether the agent’s output carries the spec’s authorship into the code, the Copyright Office’s report is routinely overread in both directions. Its objection to prompt-based authorship concerns control over execution, and the report is explicit that the objection is technological rather than conceptual: “While highly detailed prompts could contain the user’s desired expressive elements, at present they do not control how the AI system processes them in generating the output,” and “there may come a time when prompts can sufficiently control expressive elements in AI-generated outputs to reflect human authorship.” The report does not hold that a detailed specification is an idea. It holds that today’s systems execute loosely. That is a claim about compilers, and it is the sort engineering practice falsifies on a schedule: acceptance tests, typed interfaces, and continuous-integration gates (the checks every change must pass before it merges) exist to force an implementation to match a specification.

Downstream of the spec, permission and authorship come apart. Whether the code is a [derivative work](#) of the spec is permission: it governs who must license whom, and no court has answered it. (A derivative work is “a work based upon one or more preexisting works, such as a translation... or any other form in which a work may be recast, transformed, or adapted.”) Who authored the code is separate, and derivative status would not settle it; a screenplay is a derivative work of the novel it adapts, and its author is still the screenwriter. This essay is about authorship. It does not predict the answer. It argues where the analysis lives and what evidence will decide it.

3 Authorship Upstream of a Machine: Burrow-Giles and Aalmuhammed

The Supreme Court has faced machine authorship once before, and what it did then is a better guide than anything decided since. In [Burrow-Giles v. Sarony](#) (1884), a lithographer who had copied Napoleon Sarony’s photograph of Oscar Wilde argued that a photograph is “the mere mechanical reproduction of the physical features or outlines of some object,” with “no place for novelty, invention, or originality” beyond the manual operation of the device.

The Court did not declare photography creative. It said of the ordinary photograph: “This may be true in regard to the ordinary production of a photograph, and that in such case a copyright is no protection. On the question as thus stated we decide nothing.” Then it decided the case on the record before it, quoting the trial findings that Sarony produced the picture “entirely from his own original mental conception, to which he gave visible form by posing the said Oscar Wilde in front of the camera, selecting and arranging the costume, draperies, and other various accessories in said photograph, arranging the subject so as to present graceful outlines, arranging and disposing the light and shade, suggesting and evoking the desired expression.” Those findings, the Court held, “show this photograph to be an original work of art, the product of plaintiff’s intellectual invention.”

Three features of that holding transfer directly to the pipeline scenario. First, authorship attached to the documented choices upstream of the mechanical step, and that step did not dilute it. Second, the Court refused a categorical rule for the technology and made the outcome turn on the specific work’s record; the “ordinary” photograph was left undecided, a gradient built into the doctrine from its first machine case. Third, the win was evidentiary. Sarony prevailed because findings of fact enumerated his creative decisions one by one. Feist later quoted Burrow-Giles for the standard a claimant must meet: proof of “the existence of... intellectual production, of thought, and conception.” In this line of cases, authorship is a thing you prove about a particular work with a record of particular choices.

The issue writer whose spec-grade bug report fully determines the fix is Sarony. The pipeline is the camera. And the two-sentence problem report (“it crashes when I click save”) is the ordinary photograph, the case Burrow-Giles pointedly declined to decide.

When more than one contributor has a plausible claim, joint-work doctrine sorts them by the same superintendence test. [Section 101](#) defines a joint work as one “prepared by two or more authors with the intention that their contributions be merged into inseparable or interdependent parts of a unitary whole.” [Aalmuhammed v. Lee](#) (9th Cir. 2000), the leading case on sorting contributors, took its test straight from Burrow-Giles: an author “superintend[s]” the work, is “the inventive or master mind” who “creates, or gives effect to the idea”; “a creative contribution does not suffice to establish authorship,” and “absence of control is strong evidence of the absence of co-authorship.” Oscar Wilde posed, and, the court noted, “doubtless offered some creative advice”; the photographer was still the author.

Run that machinery on the pipeline scenario and the multi-contributor problem collapses to one inquiry. The machine cannot be a joint author; Thaler forecloses it. The issue writer and the pipeline’s operator rarely, if ever, document a shared intent to co-author (no issue tracker I have seen carries a Gilbert-and-Sullivan credit), so the joint-work route closes too. What remains is the Burrow-Giles inquiry, singular and evidentiary: which contributor, if any, superintended the selections. The Copyright Office already applies the same standard; Zarya of the Dawn asked whether the Midjourney user could “be treated as the ‘master mind’” behind the images. Who controlled the selections is a question of fact.

4 Method of Operation and Filtration: Lotus and Altai Against the Spec

The strongest cases against this argument target the very property that makes a specification compilable.

[Lotus v. Borland](#) (1st Cir. 1995) held the Lotus 1-2-3 menu command hierarchy uncopyrightable as a “method of operation” under §102(b), and it did so over an express finding of creative choice: “The fact that Lotus developers could have designed the Lotus menu command hierarchy differently is immaterial to the

question of whether it is a ‘method of operation.’” Expressive choices in naming and arranging the commands “do not magically change the uncopyrightable menu command hierarchy into copyrightable subject matter.” Aim that reasoning at a specification and it bites hard. On one reading, a spec precise enough for an agent to execute is a method for operating the agent. The tighter it constrains the machine, the more it resembles the thing Lotus excluded, and Lotus has already said that the existence of alternatives does not save it.

The holding’s reach is limited. The Supreme Court [affirmed 4-4](#), so the holding binds only the First Circuit; the Federal Circuit in [Oracle v. Google](#) declined to follow it, noting that “no other circuit has adopted the First Circuit’s ‘method of operation’ analysis”; and the Supreme Court later [resolved that dispute on fair use](#) without touching copyrightability. But limited is not overruled. A spec-authorship case filed in Boston meets Lotus at the door.

[Computer Associates v. Altai](#) (2d Cir. 1992) supplies the second counter-case: after filtration, a specification may have nothing protectable left. Before comparing two programs, a court must filter out everything unprotectable: elements “dictated by considerations of efficiency,” elements “required by factors external to the program itself” (the court enumerates mechanical specifications, compatibility requirements, manufacturers’ design standards, industry demands, and accepted programming practice), and elements from the public domain. Only then does it ask what remains, and the court’s own phrasing concedes the answer can be nothing: “there may remain a core of protectable expression.” Run a specification through that filter and much of it drains away. The locking strategy in the bugfix spec is dictated by the platform’s concurrency rules. The API shape is dictated by compatibility. The acceptance test restates the bug. Altai also carries the merger doctrine within it, citing [Baker v. Selden](#) and [Morrissey v. Procter & Gamble](#): “when there is essentially only one way to express an idea, the idea and its expression are inseparable and copyright is no bar to copying that expression.” A spec for a narrow fix, in a codebase whose conventions dictate the form, can merge with the fix itself.

The more completely a spec determines the implementation, the more it looks like a method of operation. And the more its content was dictated rather than chosen, the thinner the protectable residue after filtration. Feist alone does not answer that objection. Both counter-cases, though, need something to operate. Altai filtration is a per-work factual inquiry into which elements were dictated by external factors and which were free choices. Lotus asks whether a given text functions as the means of operating a machine or as an expressive account of one. Neither inquiry can run without a record of which decisions the problem forced and which the author made among live alternatives. The doctrine that most threatens spec authorship is also the doctrine that most needs the evidence described next.

5 The Investigation Log as Evidence of Selection

The record is the investigation’s own log, kept as it runs: each candidate explanation of the failure, the test run against it, the evidence that came back, which alternatives died and which survived to become the fix. The pipeline described in §1 commits one such record per investigation, timestamped, alongside the code. One pipeline has published [184 of them](#). That pipeline is mine, and so is the repository. Read the number as an interested party’s proof of concept, and discount the proposed standard accordingly. No court has seen such a record either; §8 states how untested the instrument is. The claim is about shape: the record answers, item by item, what the doctrine already asks.

Every doctrine above turns on a single quantity: how much selection intervened between the problem statement and the solution. The record’s depth measures it.

A shallow record points authorship at the issue writer. One candidate explanation, confirmed immediately, the fix falling out mechanically: that is a machine-generated finding that the problem statement dictated the solution. Read against Altai, everything in the fix was “required by factors external”: the issue specified the failure, the acceptance criterion, and the approach. The investigation confirmed; it chose nothing. Whatever selection happened, the issue writer’s text did it.

A deep record shows where originality entered and who supplied it. It branches: five candidate explanations, three killed by evidence, one split, the survivor selected between two implementations that both passed the stated criteria, on grounds the issue never mentioned. That record documents unforced selection among live alternatives, which is Feist’s definition of where originality lives. It also documents who selected. When a human directed the investigation, choosing which alternatives to pursue and which implementation to keep, the record is Burrow-Giles’s findings of fact kept contemporaneously: “intellectual production, of thought, and conception,” enumerated and attached to a name. When the pipeline ran autonomously, the same record shows selections no human made, and under Thaler a machine’s unforced choice belongs to no one. A deep

autonomous record is evidence against every human claimant in the chain, the issue writer included. An instrument that can return “no author” is the only kind worth building.

The relevant institutions already reason this way. The Copyright Office, explaining in [Zarya of the Dawn](#) why Midjourney’s user was not the author of its images, wrote that “because of the significant distance between what a user may direct Midjourney to create and the visual material Midjourney actually produces, Midjourney users lack sufficient control over generated images.” The Office’s own test is a distance between direction and output. It had no instrument, so it estimated the distance from the tool’s general behavior. The investigation record is that instrument for a specific work: it does not assert the distance, it exhibits it, decision by decision, with dates. The filtration inquiry of §4 asks which elements were dictated and which were chosen; the record’s killed branches are the dictated elements (the evidence, an external factor if ever there was one, eliminated them) and its surviving selections are the choices. A deep record is filtration computed at development time instead of reconstructed by experts in discovery. And §3 already showed what enumerated findings of choice are worth in court; the investigation log is those findings, kept contemporaneously by the process itself.

6 The Log Applied Against the Author’s Own Specification

A provenance record is only evidence if it can cut against the person who keeps it. Here is where mine cut against me.

The spec-as-source claim survived its first test and failed its second. A [blog series](#) specifying an ad exchange (auction math, privacy architecture, billing model, interface) compiled, through a coding agent, into a [working system](#) in a weekend. It was greenfield work, and the chain from prose to running code held. Then I ran the same method against an existing codebase, and [the record shows it failing](#). The spec could not carry the constraints that decided correctness: type contracts, guards against concurrent writes, checks that data dimensions line up. Code review found ten bugs the prose never guarded against. The architecture survived; the implementation corrected the spec roughly twenty times.

For that work, a claim tracing the artifact’s expression to my spec would overclaim, and my own record proves it: twenty selection events happened downstream of the text I wrote. Whoever, or whatever, made those corrections did creative work my spec cannot absorb. The record attributes them where they occurred instead of where I would prefer them.

That is the difference between provenance and advocacy. A record that always vindicates its keeper is a press release. This one assigns shallow depth to my contribution when the investigation did the work, and it assigns a well-specified change to the issue writer who never touched the pipeline. Evidence that can convict you is evidence a court can use.

7 Why Courts Will Settle This Late, and What to Record Meanwhile

Doctrine in this area arrives late, narrowly, and after practice has settled. The pattern is old enough to plan around. Congress protected photographs by statute in 1865; the Supreme Court upheld that statute nineteen years later, on the single record §3 describes, for that photograph alone. Feist repudiated sweat-of-the-brow in 1991, after decades in which lower courts had drifted from the principle. Thaler (§1) decided the narrowest question presented, and the Court declined to hear even that. The line-drawing question is open at every level. The institution that has said the most about it, the Copyright Office, qualifies its own position as contingent on “current generally available technology” and subject to revision.

That contingency is institutional as well as technical. The Register of Copyrights who issued the report was [fired by White House email on May 10, 2025](#), the day after her office released its [pre-publication report on AI training](#). The D.C. Circuit [restored her by injunction](#), and on June 30, 2026 the Supreme Court [declined to disturb it](#) (No. 25A478). The nation’s copyright regulator holds office through preliminary relief in a dispute that began the day after she published on this exact subject. Guidance from an office in that position is a weathervane.

If the thesis is right, there will be many line-drawing cases and no landmark. When they arrive, courts will do what Burrow-Giles did: refuse the categorical ruling and decide the work in front of them on its record. The parties who win those cases, whichever way each line runs, are the ones holding contemporaneous, dated, third-party-verifiable records of who decided what. Chat transcripts are transient, trapped in vendor logs, and silent about which alternatives were considered. An investigation record committed to a public repository is none of those things. Timestamps do not decay while doctrine deliberates; a 2026 commit is still a 2026 commit when the case is argued. The engineering practice worth adopting now costs nothing

beyond what a disciplined investigation already produces: keep the record the eventual court will subpoena anyway, in a form it can read.

8 Limitations

Nothing here is legal advice. Every quotation was checked against the primary opinion or report, and the disclosure below describes how. Checking quotes is not legal training, and a litigator will see moves here that a court would not permit.

No court has answered either half of what §2 leaves downstream of the spec: whether agent-compiled code is a derivative work of the specification, and where authorship vests in the compiled change. On the second, §4 describes a live doctrinal path to an answer that cuts against the spec-writer. Lotus remains good law in the First Circuit. Altai filtration may leave a given spec with no protectable residue, and merger will swallow the narrow ones.

Depth is an evidentiary proxy, and proxies invite gaming. An adversary can pad a record with alternatives it never seriously entertained. The tamper-resistance lives in the hosting. A git commit's author date is metadata its committer can rewrite at will, so the file format proves nothing on its own. The evidentiary weight sits in the server-side trail: the host's receipt of each push, pull-request and test-run timestamps, third-party mirrors and archives. That trail reaches a court through the ordinary machinery for electronic records, Rule 901 authentication and the Rule 902(13) and (14) certification routes for system-generated records and hash-verified copies.

Padding has a concrete tell under that machinery: a genuinely explored dead end leaves execution residue (the failing test run, the reverted diff, the job log), while a branch narrated after the fact leaves only prose. Still, no court has admitted such a record, weighed one, or seen one. The claim this essay makes for it is that it is the right shape of evidence for the inquiry the doctrine already runs, and that claim is falsifiable in the cheapest possible way: by the first court that looks at one and finds it unhelpful. And the instrument cuts against the spec-writers it would otherwise serve; run autonomously at depth, it returns no author (§5), a reading I accept.

The Copyright Office's prompt analysis is explicitly technology-contingent, so the prompt/spec line drawn in §2 inherits that contingency. A future system that executes loosely against detailed specs weakens the compiler analogy; a future report could redraw the Office's line in either direction.

LLM collaboration disclosure

LLMs enter this essay in two roles. Instrument: research agents fetched the primary sources (opinions, dockets, the Copyright Office reports) and extracted the quoted passages; the verification log, mapping each claim to its source and quote, was reviewed by the author, and the extracted source texts were retained. Writing aid: the prose was drafted and revised with Anthropic's Claude from the author's five prior essays on this subject and his direction; the argument, the through-line, and the decisions about what to claim and what to concede are the author's. No LLM decided what to publish.

Funding

This work was conducted independently, with no external, institutional, or commercial funding, and no client relationship with any party named.