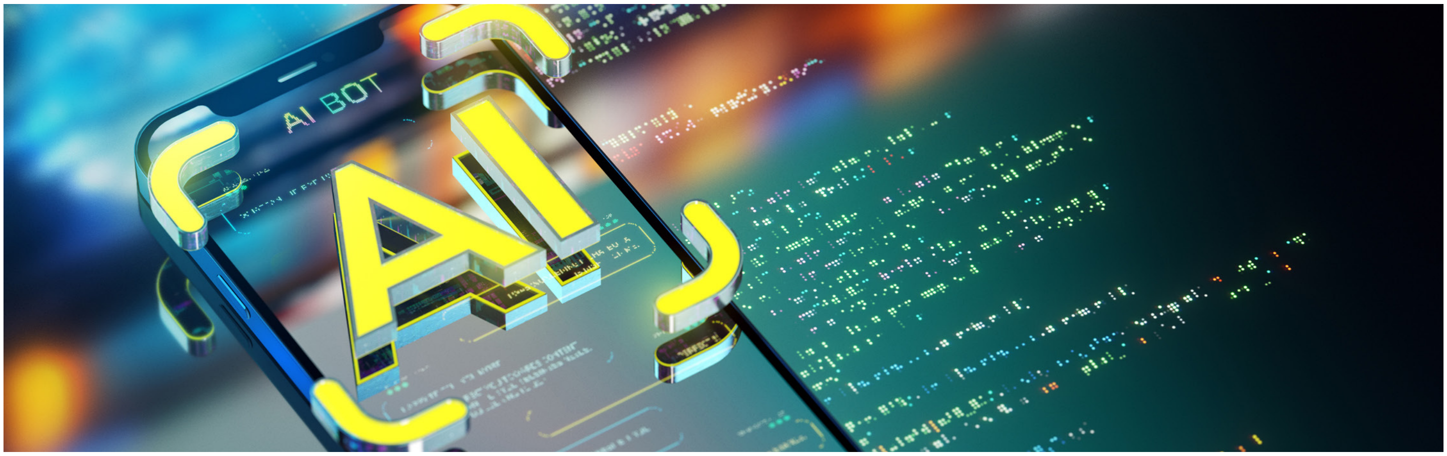
SheppardMullin



Court Rules in Landmark AI Code Generator Case – DMCA Claims Dismissed But Breach of Contract Claims Remain

By: James Gatto

The battle between open source software developers and the leading AI code generators will rage on. Despite the Court dismissing Plaintiffs' Digital Millennium Copyright Act (DMCA) Section 1202(b) claims with prejudice, it declined to dismiss Plaintiffs' claim for breach of contract of open source license violations by Defendants. However, the Court also dismissed Plaintiffs' request for of unjust enrichment and punitive damages.

This case is pivotal for the future of AI code generators, one of the most powerful productivity enhancement tools for software developers. Code generators are trained on open source software found in public repositories, such as GitHub, and provide suggestions to developers as they write code. Because the open source code is subject to license terms with which the code generators allegedly do not comply, the developers contend that their operation violates the licenses.

This is the second round of motions to dismiss in this case. Initially, the Plaintiffs' alleged, among other things, that Defendants used their source code to train AI models upon which the code generators are based and these code generators reproduce portions of the Plaintiffs' source code without providing the attribution mandated by the open source licenses. They alleged this is a breach of the licenses, and a violation of the DMCA, which prohibits the removal of certain copyright management information (such as copyright notices), from copyrighted works (i.e., the source code).

Following the last round of briefing, the Court dismissed Plaintiffs' state law claims for intentional and negligent interference with prospective economic relations, unjust enrichment, negligence, and unfair competition with prejudice. The Court also dismissed Plaintiffs' claim under Section 1202(b) of the DMCA, reasoning that Plaintiffs failed to meet Section 1202(b)'s identicality requirement. However, Plaintiffs were granted leave to amend their DMCA claim.

Plaintiffs filed a second amended complaint (SAC). In the SAC, Plaintiff alleges violation of DMCA Section 1202(b)(1) and 1202(b)(3) and breach of contract for violation of the open source licenses. Defendants moved the Court to dismiss Plaintiffs' claims.

Section 1202(b) Claims

The Court dismissed Plaintiffs Section 1202(b) claims for failing the identicality requirement. The Court found that the SAC does not identify even a single example of the code generators producing an identical copy of any work. While Plaintiffs alleged that, should a user elect to not use the duplication-detection feature[1], a user could conceivably view an identical match of 150 characters, and use it without attribution. Plaintiffs also submitted a recent academic research study which shows that the likelihood developers code would be emitted verbatim is only increasing.[2]

The Court found that while Plaintiffs aver that the duplication-detection tool by definition establishes the ability to reproduce verbatim copies of code they do not explain how the tool makes it plausible that it will in fact do so through its normal operation or how any such verbatim outputs are likely to be anything beyond short and common boilerplate functions.[3]

Additionally, the Court found that reliance on the research paper was unpersuasive because at most it showed that when models are "prompted appropriately, they will emit the memorized training data verbatim." However, it also noted that the AI code generators rarely emit memorized code in benign situations, and most memorization occurs only when the model has been prompted with long code excerpts that are very similar to the training data.

For at least these reasons, the Court dismissed Plaintiffs' Section 1202(b) claims with prejudice.

Breach of contract

Defendants moved to dismiss Plaintiffs' breach of contract claim for violation of open source licenses, arguing: (1) Plaintiffs fail to state a claim based on the specific operation of the code generators at issue (Codex and Copilot); and (2) Plaintiffs' theory based on Copilot fails on the merits because the attribution and notice terms are conditions and do not give rise to a contract claim.

In its prior order denying Defendants' motions to dismiss Plaintiffs' breach of contract claim, the Court explained: Plaintiffs advance claims for breach of the eleven suggested licenses GitHub presents to users that require (1) attribution to the owner, (2) inclusion of a copyright notice, and (3) inclusion of the license terms. Plaintiffs allege that use of licensed code is allowed only pursuant to the terms of the applicable License, and that each such license requires that any derivative work or copy include attribution, a copyright notice, and the license terms. Plaintiffs further allege that the code generators reproduce licensed code as output without attribution, copyright notice, or license terms, thereby violating the relevant provisions of each license. The Court noted that while Plaintiffs do not identify the specific subsections of each suggested license that correspond to each of these requirements, the Court

[1] The complaint alleges that code generators include a user-settable filter called the "duplication-detection feature." Users can set the filter to either allow or block code completion suggestions that match publicly available code. If a user chooses to block suggestions that match public code, the tool checks code completion suggestions with their surrounding code of about 150 characters against public code on GitHub. If there is a match, or a near match, the suggestion is not shown to the user. However, this feature is entirely optional to users, and provides no such optionality to licensors. Based on this, Plaintiffs' alleged users who want to receive identical code or do not want to exclude it, may do so. Thus, Plaintiffs believe it is likely that their licensed code is output in violation of the open source licenses and that there is a substantial risk, if not certainty, that identical code will be emitted in the future.

[2] See, *Quantifying Memorization Across Neural Language Models* by Nicholas Carlini et al., which reasoned that memorization significantly grows as we increase (1) the capacity of a model, (2) the number of times an example has been duplicated, and (3) the number of tokens of context used to prompt the model. Accordingly, Plaintiffs assert that as generative AI models increase capacity and continue to scale, it becomes more likely that training data will become memorized and emitted verbatim, i.e., as an exact duplicate."

[3] Presumably, this reference to "short and common boilerplate functions" is an implicit assumption by the Court that this would not include copyright protectable works.

finds that Plaintiffs have sufficiently identified the contractual obligations allegedly breached, as required to plead a breach of contract claim. Having already determined that Plaintiffs adequately stated a breach of contract claim, the Court declines to reanalyze this issue and refused to dismiss the breach of contract claims.

Conditions Versus Covenants

Defendants argued that even if they violated the attribution and notice terms of the Licenses, those breaches were of conditions that sound in copyright law, and therefore Plaintiffs' claims for breach of contract must be dismissed.[4]

The Court found that while the attribution and notice terms in the Licenses at issue are conditions, this does not impede Plaintiffs' ability to bring a breach of contract claim, noting it is common for courts to say that if there is a material breach of a condition of the license, the copyright owner has the option of suing for copyright infringement or breach of contract, but if there is a violation of a covenant, only a breach-of contract claim will lie. Accordingly, the Court declined to read in a requirement that a plaintiff must bring suit for copyright infringement in the event of a breach of condition. Rather, it held that suing for copyright infringement is not the exclusive avenue a plaintiff must pursue in the event of a breach of a condition of a license—it is simply one option a plaintiff may elect. For at least these reasons, the Court concluded that Plaintiffs have alleged a breach of contract claim for violation of open-source licenses.

Unjust Enrichment and Punitive Damage Requests

The Court dismissed Plaintiffs' requests for unjust enrichment and punitive damages. It found unjust enrichment is not a cause of action or even a remedy, but rather a general principle underlying various legal doctrines and remedies. It is synonymous with restitution. California law supports this position: absent an exception, "a quasi-contract action for unjust enrichment does not lie where, as here, express binding agreements exist and define the parties' rights."

Accordingly, the Court found unjust enrichment damages are not available.

Punitive Damages

On the arguments concerning punitive damages, the Court agrees that these claims must be dismissed, in part, because Plaintiffs' opposition failed to address this basis for the motion to dismiss. However, the Court added that the sole remaining causes of action are for breach of contract, for which punitive damages are generally not recoverable.

Conclusion

While Defendants have been successful in whittling away many of the claims in this case, it appears this case will go forward on the core breach of contract claims for failing to comply with the conditions of the open source licenses.

It is important to note that the defendants in this case are the developers of the AI code generators, not the users. So what does this case mean for users? Arguably, if the code output from these AI code generators does not comply with the terms of the applicable open source licenses, users of that code may be liable for breach of contract as well. This issue will not be addressed in this case.

[4] Various open source disputes have addressed the significance of the intersection between covenants, conditions, and copyright law. The Court summarized some of these cases noting as follows. "Generally, a copyright owner who grants a nonexclusive license to use his copyrighted material waives his right to sue the licensee for copyright infringement and can sue only for breach of contract." *Jacobsen v. Katzer*, 535 F.3d 1373, 1380 (Fed. Cir. 2008). "If, however, a license is limited in scope and the licensee acts outside the scope, the licensor can bring an action for copyright infringement." *Jacobsen*, 535 F.3d at 1380. "[C]ontractual terms that limit a license's scope [are] 'conditions,' the breach of which constitute copyright infringement." "[A]ll other license terms [are] 'covenants,' the breach of which is actionable only under contract law." Id.

However, users of these tools can take steps to avoid potential liability. As I have previously addressed, companies should update their open source policies to reflect the use of AI code generators. In updating these policies, companies should consider using the duplication-detection feature and/or the code reference feature provided by some of these tools. The code reference feature identifies code output by the code generator that matches publicly available code on GitHub and provides details about the matching code when you accept such suggestions. This helps you to determine any applicable license, so you can determine and comply with any license conditions. For more information on addressing legal issues with AI code generators, see Solving Open Source Problems With AI Code Generators – Legal Issues and Solutions.

For companies that do not yet have an open source policy, the pervasive use of AI code generators by your developers should be a good impetus to do so. For more information, see Open Source Software Policies – Why You Need Them And What They Should Include. For companies that have not yet developed open source policies or updated them to address the use of AI code generators, it is often helpful to have a knowledgeable attorney conduct an in-house presentation and help guide you in the development or updating of these policies.

For more details on generative AI or open source, please contact:

James Gatto
Artificial Intelligence Team Leader
Open Source Team Leader
bio
202.747.1945
jgatto@sheppardmullin.com

James Gatto is a partner in the D.C. office of Sheppard Mullin and is Co-Leader of its Artificial Intelligence Team and leader of the Open Source Team. For over 35 years he has been a thought leader on legal issues with emerging technologies and business models, including over 20 years of experience with AI and open source. He provides strategic advice on all IP, tech transactions and tech regulatory issues. He is an adjunct professor at Ole Miss Law School where he teaches Legal Issues with AI. He has been an invited speaker for the US Copyright Office Listening Session on AI Authorship, the USPTO Listening Session on AI Inventorship Issues and the Korean Copyright Commission conference on AI and open source. He is an appointed member of the AI/Machine Learning Task Force of the American Bar Association's IP Section. He is also leader of the AI subcommittee of the American Intellectual Property Law Association (AIPLA) member of the AI committee of the International Technology Law Association.

This alert is provided for information purposes only and does not constitute legal advice and is not intended to form an attorney client relationship. Please contact your Sheppard Mullin attorney contact for additional information.