No. 18-956

IN THE

# Supreme Court of the United States

GOOGLE, LLC,
*Petitioner*,

v.

ORACLE AMERICA, INC.,
*Respondent*.

ON WRIT OF CERTIORARI
TO THE UNITED STATES COURT OF APPEALS
FOR THE FEDERAL CIRCUIT

## BRIEF FOR AMICI CURIAE PYTHON SOFTWARE FOUNDATION, TIDELIFT, OPEN UK, AND PROTOCOL LABS, IN SUPPORT OF PETITIONER

VAN LINDBERG
JILL WHEATON*
DYKEMA GOSSETT PLLC
112 E. Pecan Street #1800
San Antonio, TX, 78215
jwheaton@dykema.com
vlindberg@dykema.com
Ph. (210) 554-5294
** Counsel of Record*

LUIS VILLA
TIDELIFT, INC.
50 Milk Street
16th Floor
Boston, MA 02109
luis@tidelift.com

## TABLE OF CONTENTS

## TABLE OF CITED AUTHORITIES

## Other Authorities

Robert Eckstein, *James Gosling on Open Sourcing Sun's Java Platform Implementations, Part 1*, (Oct. 2006) web.archive.org/web/20180330223540/www.oracle.com/technetwork/articles/javase/gosling-os1-qa-142025.html ............................................ 18

Robert Eckstein, *James Gosling on Open Sourcing Sun's Java Platform Implementations, Part 2*, (Nov. 2006) web.archive.org/web/20170812190620/www.oracle.com/technetwork/articles/javase/gosling-os2-qa-136546.html .......................................... 4, 19

H.R. REP. NO. 94-1476 (1976) ................................ 6

Peter S. Menell, Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software, 31 HARV. J.L. & TECH. 303 (2018) ............................................................. 14-15

Bruce Perens, The Open Source Definition (Annotated), version 1.9 (Jul. 2006) opensource.org/osd-annotated ......................... 16

Richard Stallman, What is free software?, version 1.165 (Jul. 29, 2019), www.gnu.org/philosophy/free-sw.en.html ....... 17

Alan Reynolds, Microsoft's Acquisition of GitHub Is Not 'Anti-competitive', CATO Institute (Jul. 18, 2018), www.cato.org/publications/commentary/microsofts-acquisition-github-not-anticompetitive....... 23

2016 Global Developer Report, Gitlab, Inc. (2016) page.gitlab.com/2016-Developer-Survey_2016-Developer-Survey.html .................................... 23

## BRIEF OF *AMICI CURIAE* IN SUPPORT OF PETITIONER

## INTEREST OF THE *AMICI CURIAE* [1]

*Amicus* Python Software Foundation is a 501(c)(3) non-profit corporation that holds the intellectual property rights behind the open source Python programming language. It manages the open source licensing for Python version 2.1 and later and owns and protects the trademarks associated with Python.

*Amicus* Tidelift, Inc. is a venture-backed corporation that works directly with open source maintainers to support open source components and the enterprises that use them.

*Amicus* OpenUK is a UK organisation committed to supporting the growth of the open source software, hardware and data communities in

[1] No counsel for any party has authored this brief in whole or in part, and no person other than *amici* or their counsel have made any monetary contribution intended to fund the preparation or submission of this brief. Both parties were provided notice of intention to file ten days before the filing of this brief. In an email dated December 18, 2019, Respondent granted consent for the filing of this brief. Petitioner has given blanket consent to the filing of timely briefs for *amici curiae*. Counsel for *amici curiae* was previously engaged to advise Google in connection with this matter earlier in its history, and represents Google in other matters, but Google has had no involvement with the preparation of this brief.

the UK, and representing the UK "Open" communities globally.

*Amicus* Protocol Labs, Inc. is a research, development, and deployment lab for improving Internet technology through a variety of open source projects.

*Amici* file this brief because all the software at issue in this case is subject to open source licenses. The resolution of the questions presented in the petition will directly affect the scope and enforceability of open source licenses crucial to *amici*'s ongoing operations, or in the case of OpenUK, the operations of its members.

## SUMMARY OF ARGUMENT

Software is different from other literary works in that it is a mixed work, including both functional and expressive elements. As the Court explained in *Harper & Row, Publishers, Inc. v. Nation Enters.*: "[C]opyright is limited to those aspects of the work—termed 'expression'—that display the stamp of the author's originality." 471 U.S. 539, 547 (1985). Copying the expressive elements of a software work can constitute copyright infringement. Copying the functional elements of a software work is allowed.

In this case, Google copied the declarations for 37 packages from Oracle's Java SE work into

Android, totaling about 11,000 lines of code. The dispositive question in this case is whether the declarations copied by Google were part of the expressive, copyright-protected elements of the Java SE work, or whether they were part of the functional, non-protected elements of the work.

To answer this question, the Court should apply its "conceptual separability" test from *Star Athletica, L.L.C. v. Varsity Brands, Inc.*, 137 S. Ct. 1002 (2017). As this Court held in *Star Athletica*, the expressive elements of a mixed work must be conceptually removed from the functional elements of the work. Only if there are expressive elements that are separable from the functional elements of the useful article does the court proceed to evaluate infringement.

The Federal Circuit's decisions, either with regard to copyrightability ("*Oracle I*," App. at 121a) or fair use ("*Oracle II*," App. at 1a), share the same fundamental error: they fail to deal appropriately with software as a mixed work consisting of both copyrightable expression and uncopyrightable functionality. This Court should apply its existing *Star Athletica* analysis to provide a simplifying and unifying rule for analyzing copyrightability in all mixed works.

The Federal Circuit failed to evaluate properly the copyrightability of the work before it, and compounded its error by failing to weigh Google's fair use arguments in light of Oracle's actions in the marketplace. Before Google released Android, Oracle made a calculated decision to provide its software to the public under a widely used open source license called the GNU General Public License (the "GPL"). Oracle was aware of the tradeoffs involved when it chose to release its software under the GPL. In blog posts celebrating the open source release, vice president James Gosling described Oracle's hope that releasing the software as open source would increase its use and distribution.[2] More significantly, Gosling anticipated that other open source Java implementations would "mine [Oracle's] source for stuff to incorporate into their projects."[3] This is exactly what Google did.

Software developers, especially open source developers, rely on easily-understandable boundaries between copyright-protected expression and non-protected functionality. This Court has the opportunity to make those boundaries clear. Both open source and proprietary software development depend on the understanding that independent reimplementation is a common, pro-competitive and legally permissible activity.

[2] James Gosling was vice president for Sun Microsystems, which was purchased by Oracle in 2010. For clarity, both entities are referred to in this brief as "Oracle."

[3] Robert Eckstein, James Gosling on Open Sourcing Sun's Java Platform Implementations, Part 2, (Nov. 2006) web.archive.org/web/20170812190620/www.oracle.com/technetwork/articles/javase/gosling-os2-qa-136546.html.

## ARGUMENT

### I. Under *Star Athletica*'s "conceptual separability" test, copyright law should not preclude reuse of software declarations and interfaces

Computer software is challenging for copyright because it is a mixed work, incorporating both functional and expressive elements. Copyright cannot be used to monopolize the functionality inherent in such mixed works.

In *Star Athletica,* the Court analyzed the copyrightability of a design included in a "useful article." In its analysis, the Court provided the test for separating a mixed work's functional and expressive elements. If a work's copyrightable elements are "conceptually separable" from the work's functional, "utilitarian" elements, then the work is copyrightable as a whole. But only the separable, expressive elements are protected by copyright. It does not infringe copyright to copy and re-use the functional elements of a mixed work.

Applying the logic from *Star Athletica's* "conceptual separability" test, the declarations contained in the Java software and used by Google in the Android code are inseparable from the functionality embedded in the Java software. Thus, this Court should hold that Google's use of the Java software declarations was allowed under copyright law.

### A. Software is a "useful article" as defined in the copyright statute, containing both functional and expressive elements

Computer software is protectable as a literary work under the Copyright Act. *See* 17 U.S.C. §§ 101, 102(a)(1) (literary works are the subject of copyright protection); *see also* H.R. REP. NO. 94-1476, at 54 (1976)("The term 'literary works' ... includes ... computer programs to the extent that they incorporate authorship in the programmer's expression of original ideas, as distinguished from the ideas themselves").

But functional aspects of a work are not protected under copyright: "In no case does copyright protection for an original work of authorship extend to any idea, procedure, process, system, method of operation, concept, principle, or discovery, regardless of the form in which it is described, explained, illustrated, or embodied in such work." 17 U.S.C. § 102(b).

Software in particular has many such functional elements, and its structure and contents are primarily dictated by utilitarian concerns. As the Ninth Circuit correctly stated in *Sega Enters. Ltd. v. Accolade, Inc.*: "[C]omputer programs are, in essence, utilitarian articles—articles that accomplish tasks. As such, they contain many logical, structural, and visual display elements that are dictated by external factors such as compatibility requirements and industry demands...." 977 F.2d 1510, 1525 (9th Cir. 1992).

The copyright statute already includes a term to describe mixed works like software: a "useful article." A "useful article is an article having an intrinsic utilitarian function that is not merely to portray the appearance of the article or to convey information." 17 U.S.C. § 101. This definition exactly describes software. Software can include many expressive and informative elements, but the predominant value of software is found in its utilitarian functions.

The statute's definition of a computer program further reinforces the identification of software as a "useful article": a computer program is "a set of statements or instructions to be *used* directly or indirectly in a computer in order *to bring about a certain result*." *Id.* (emphasis added). Only a useful article is intrinsically capable of being "used" to "bring about a certain result."

Oracle may argue that the statute categorically excludes useful articles from copyright. That is not so. Labeling software as a "useful article" does not diminish the copyrightability of a software work as a whole. The copyright statute explicitly states that a useful article can be protected by copyright if it has "features that can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article."[4]

[4] *Id.*, See definition of "Pictorial, graphic, and sculptural works" ("PGS works"). The statute specifically protects PGS works incorporated into useful articles, but the definition of "useful article" is not limited to PGS works.

### B. *Star Athletica* states the correct test for distinguishing between a work's functional and expressive elements

This Court recently addressed copyright protection for works that contain both functional and expressive elements. Although the present case concerns computer code rather than clothing, the issues are conceptually similar to those in *Star Athletica, L.L.C. v. Varsity Brands, Inc.*, 137 S. Ct. 1002 (2017).

In *Star Athletica*, the Court evaluated the copyrightability of designs on a cheerleading uniform. *Id.* at 1016. As the Court explained, the expressive elements of a mixed work must be conceptually removed from the functional elements of the work. *Id.* at 1010. Only if there are expressive elements that are separable from the functional elements of the useful article does the court proceed to evaluate infringement.

In *Star Athletica*, the Court expressly acknowledged that copying functional elements of a useful article is allowed: "To be clear, the only feature of the cheerleading uniform eligible for copyright in this case is the two-dimensional work of art.... respondents have no right to prohibit any person from manufacturing a cheerleading uniform of identical shape, cut, and dimensions to the one on which the decorations ... appear." *Id.* at 1013.

The *Star Athletica* Court evaluated mixed works in the context of two-dimensional, graphical art, but the principle undergirding the Court's analysis directly applies to software. The principle is the same: the functional, utilitarian elements of a mixed work are not protectable under copyright, and the existence of a copyright in one part of a mixed work must not prohibit others' use of the utilitarian, functional aspects of the mixed work.

### C. Because Google only copied the software's inseparable functional elements, it did not infringe Oracle's copyright

Google copied the declarations for 37 packages from the Java SE work into Android. These "declarations" are names and phrases that allow programmers to refer to and use specific functional elements within the work. JA34-35; JA38-39; see also Pet. App. 223a-225a. The dispositive question in this case is whether the declarations copied by Google were part of the expressive, copyright-protected elements of the Java SE work, or whether they were part of the functional, non-protected elements of the work. *Star Athletica*'s "conceptual separability" test substantially simplifies the analysis.

A software component satisfies the conceptual separability test if it "(1) can be identified separately from, and (2) is capable of existing independently of, the utilitarian aspects of the article." 137 S. Ct. 1002, 1010 (internal quotation marks removed).

As in *Star Athletica*, the first requirement "is not onerous"—there must simply be some identifiable element of the work that includes possibly-copyrightable content. *Id.* In this case, the Federal

Circuit identified the declarations and the non-literal "structure, sequence, and organization" (SSO) of those declarations within the larger Java SE work. Pet. App. at 158a-166a.

The second requirement is that "the separately identified feature has the capacity to exist apart from the utilitarian aspects of the article." 137 S. Ct. 1002, 1010. The declarations fail this part of the test, because accessing the functional aspects of the work *requires* the use of the *exact names and phrases* copied by Google out of the Java SE work—the functional aspect cannot "exist apart" from them. As the District Court found:

> "Significantly, the rules of Java dictate the precise form of certain necessary lines of code called declarations, whose precise and necessary form explains why Android and Java *must be identical when it comes to those particular lines of code*."[5]

Or as explained by Oracle's expert witness:

> Q. ...[I]f instead you had written completely new APIs, would programmers be able to *access these functionalities using the names that they have memorized* and have used for years?

[5] Pet. App. 221a (emphasis added).

A. The new APIs use different names, *then the old names would not work.*"[6]

As shown, the purpose of the declarations was to "access... functionalities" within the code—a utilitarian feature. The precise organization of the declarations had to be "identical" or the code "would not work." *See* Pet. App. 221a; JA 70.

Because there was no way to access the Java language functionality outside of the declarations, the declarations are not "conceptually separable" from the functionality they expose, and are part of the non-copyrightable, functional, utilitarian elements of the work.

Thus, as Google only copied the software's inseparable functional elements, it did not infringe Oracle's copyright.

### D. Using the *Star Athletica* "conceptual separability" test avoids pitfalls associated with tests employed by the lower courts

The leading case addressing the copyrightability of mixed works is *Baker v. Selden*, 101 U.S. 99 (1880). In *Baker*, the Court held that copyright protection did not extend to the forms that embodied an accounting method included in Selden's book, giving rise to the "idea/expression" dichotomy

[6] JA 70 (emphasis added).

and the "merger" doctrines briefed by other parties before this Court. *Id.* at 106-07.

Unfortunately, lower courts have been unable to consistently separate the functional and expressive elements in copyrighted works. To quote Judge Learned Hand: "The test for infringement of a copyright is of necessity vague. . . . Decisions must therefore inevitably be ad hoc." *Peter Pan Fabrics, Inc. v. Martin Weiner Corp.*, 274 F.2d 487, 489 (2d Cir. 1960).

The confusion in the lower courts is exemplified by the two Federal Circuit rulings below. In its first decision on copyrightability ("*Oracle I*," App. at 121a), the Federal Circuit held that functional interoperability is "irrelevant to copyrightability." *Id.* at 166a. But in its second decision on fair use ("*Oracle II*," App. at 1a) the *same* Federal Circuit panel concluded that the fact that Google's software "perform[s] the same functions" as Oracle's software is evidence that Google infringed Oracle's copyrights.[7] Functionality cannot be both "irrelevant" to copyrightability and evidence of infringement.

Had the Federal Circuit applied the "conceptual separability" test in *Oracle I* and *Oracle II*, its holdings would have been both more clear and consistent. If the *Oracle I* court had correctly separated

[7] The Federal Circuit rejected Google's arguments regarding the significance of the smartphone context, even while recognizing that "the declarations and SSO may *perform the same functions* in Android and Java." *Oracle II.* at 31a (emphasis added).

the functional and expressive elements of the Java SE work, the test would have shown that no expressive elements were copied, resulting in a holding of no infringement. Alternatively, the *Oracle II* court would have held that the separable expressive elements copied by Google were truly *de minimis*, and qualified as fair use.

Conceptually separating the functional and expressive elements of the analyzed work helps avoid other analytical pitfalls illustrated in this case. For example, in its Brief in Opposition, Oracle highlights its "well-settled understanding that §102 codifies the idea/expression dichotomy," but then proceeds to argue that a work cannot be "both an original work ... protectable under Section 102(a) and a 'method of operation' ... under 102(b)." Br. in Opp. 16-17, quoting U.S. Br. 13.

Applying the *Star Athletica* test makes Oracle's error clear: a work may be copyrightable as a whole, but only those expressive elements that are conceptually separable from any utilitarian functions receive copyright protection.

Similarly, there is confusion about the merger doctrine. The merger doctrine states that if there are only a limited number of ways to express a particular idea, then the idea and the expression "merge" and the merged expression cannot be restricted by copyright. *See, e.g., Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 815 n.9 (1st Cir. 1995), *aff'd by an equally divided Court*, 516 U.S. 233 (1996). In this case, the *Oracle I* court reasoned that the merger doctrine was inapplicable because

Sun could have written the declarations in more than one way. App. at 142a-143a, 148a, 150a-151a.

From the perspective of conceptual separability, however, it does not matter whether there was one way or many ways to write the code that brought about the desired functionality. Regardless of the choice Oracle made, *all* the available methods related to the utilitarian, functional elements of the Java SE work, not to the expressive elements of the work. Under a *Star Athletica* analysis the declarations and SSO cannot be conceptually separated from the functional elements of the work. They should therefore be excluded from consideration of copyright infringement.

Relatedly, the Federal Circuit also emphasized the "creativity" needed for software development. *See, e.g.*, App. at 53a. However, an emphasis on creativity without considering function can be misleading. Creativity is shown both in copyrightable expression as well as in functional problem-solving. Selden's creativity in the development of his accounting system did not give him the exclusive right to use the forms that gave effect to his system. *See Baker*, 101 U.S. at 102. As in *Baker*, other people can copy the creative-but-functional elements in a mixed work. Copyright only prohibits copying elements separably recognizable as creative expression.

Using the *Star Athletica* test clarifies and simplifies how copyright applies to mixed works like software. In this very context, Professor Peter Menell commented on the usefulness of the separability test:

> The separability doctrine used to analyze copyright protection for pictorial, graphic, and sculptural ("PGS") works provides a useful model for applying the idea-expression dichotomy to API declarations. Courts must determine whether the expressive features of a PGS work... can be identified separately from, and are capable of existing independently of, the utilitarian aspects of the article." [8]

This Court has the opportunity to unify and simplify the copyrightability analysis for all types of mixed works simply by applying its existing *Star Athletica* "conceptual separability" test. The Court should do so.

## II. Re-implementation of software declarations and interfaces under open source licensing constitutes fair use

The dispute in *Oracle II* focused on the scope of the fair use doctrine as applied to Oracle's software. As part of that analysis, the Federal Circuit gave heavy weight under fair use factors one ("the nature and character of the use") and four

[8] Peter S. Menell, Rise of the API Copyright Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software, 31 HARV. J.L. & TECH. 303, 445 (2018).

("the effect of the use on the market") to Oracle's failures to secure commercial, royalty-bearing licenses for Java. *See, e.g. Oracle II*, App. at 51a. However, the Federal Circuit's analysis noted, but disregarded, a key fact: all the software in this case is provided under "open source" licenses granting broad permissions, including the permission to reimplement functionality, including for commercial purposes. *Id.* at 6a.

Google did not use Java SE under Oracle's open source license. No license was necessary to use the declarations. But the jury heard how Oracle positioned Java SE as "free and open" in the marketplace, and how Oracle's release of Java SE under an open source license affected the market for commercial use. JA 55, 88, 124. The jury was entitled to consider that evidence in evaluating Google's conduct and finding fair use.

## A. Open source licenses are designed to facilitate knowledge transfer and re-implementation

"Open source" is a method of licensing intellectual property designed to encourage knowledge-sharing and cooperation between parties. Open source licenses encourage cooperation by licensing copyrighted material according to a set of rules—rules that grant broad latitude to licensees.[9] This latitude even includes permission to act in ways contrary to the wishes of the copyright holder, including commercial exploitation.

Placing software under an open source license is a deliberate, strategic decision to forgo the tight control allowed by copyright to try to gain an alternative benefit such as broader distribution or use. In return, licensees are able to rely on the permissions granted in the license to provide a stable legal foundation for independent development.

In this case, Oracle made the decision to license its software under the "GPL," an existing, well-known open source license. The GPL's authors have long provided official commentary including an explanation that one of the purposes of the license is to ensure that any licensee is able to "study how the program works" for any purpose whatsoever, including reimplementing the software.[10]

[9] For a listing of these rules, see Open Source Initiative, "The Open Source Definition (Annotated)" at opensource.org/osd-annotated.

[10] Richard Stallman, What is free software?, version 1.165 (Jul. 29, 2019), www.gnu.org/philosophy/free-sw.en.html.

### B. Oracle anticipated and understood that releasing its code under an open source license would facilitate re-implementation by third parties

Oracle was aware of and counted on the software developers' established interpretations of open source licenses when it chose to release its software under the GPL. At the time when Oracle decided to make Java available under the GPL, software developers were unwilling to invest heavily in software that didn't provide the full range of permissions granted under open source licenses. As then-vice president James Gosling explained in an interview:

> Q: What does [Oracle] hope to accomplish by open sourcing [Java]?
>
> A: We want better conversations with the developer community, a more collaborative relationship. We want to have better relationships with many of the Linux distributions, and a lot of the Linux distributions are very sensitive about precisely which license one uses. We want to have better relationships with the open-source community, which leads to better distribution and makes it easy for people to collaborate with us to evolve the

> platform, to use it in even more interesting ways and in more interesting areas.[11]

Even more significantly, Oracle publicly acknowledged its belief—shared by others in the broader open source community—that releasing its work under the GPL would allow other open source implementations to learn from, take, and reimplement portions of its code. Again quoting Gosling:

> Q: What kinds of things can a developer do with the open-source Java SE platform pieces right away?
>
> A: Probably the most useful thing you can do with it is look at it and learn from it. It is somewhat traditional, but I always say that the source [code] is the documentation of last resort....
>
> Q: How do you think this move will affect other open-source implementations of the Java programming language—for example, Apache Harmony or GNU Classpath?

[11] Robert Eckstein, James Gosling on Open Sourcing Sun's Java Platform Implementations, Part 1, (Oct. 2006) www.oracle.com/technetwork/articles/javase/gosling-os1-qa-142025.html.

> A: It's hard to know. They'll certainly be able to mine our source for stuff to incorporate into their projects.[12]

Google did not take any code from Oracle's GPL-licensed Java implementation. Instead, it took Oracle at its word: it based Android on the Apache Harmony implementation of Java, and incorporated "stuff... into [its] project" to help programmers already familiar with Oracle's Java platform. JA 50-51. No license from Oracle was necessary for this use. Google's actions were consistent with both Oracle's statements, its course of dealing in Java, and the usage of trade concerning open source licenses and in particular the GPL.

### C. The Federal Circuit's fair use analysis under factors one and four disregarded the evidence that Oracle reaped the commercial benefit provided by Java SE's open source license

The Federal Circuit held that the commercial nature of Android (factor 1 of the fair use test) and the effect of the use on the market (factor 4) weighed against a finding of fair use by Google. App. at 25a-28a, 47a-53a. In doing so, however, the Federal Circuit's decision on fair use failed to take

[12] Robert Eckstein, James Gosling on Open Sourcing Sun's Java Platform Implementations, Part 2, (Nov. 2006) www.oracle.com/technetwork/articles/javase/gosling-os2-qa-136546.html.

into account the open source context already established by the lower court.

Under fair use factor one, "the purpose and character of the use," Oracle anticipated that providing Java SE under an open source license would allow others to commercially exploit the work. But commercial exploitation of the Java SE work was already explicitly allowed due to the work's open source license. As the trial court held:

> [B]efore Android was released, [Oracle] made all of the Java API available as free and open source ... *Anyone could have duplicated, for commercial purposes, the very same 37 packages as wound up in Android with the very same SSO and done so without any fee*... [O]ur jury could reasonably have found that Android's impact on the market for the copyrighted works paralleled what [Oracle] already expected via its OpenJDK. [13]

Under fair use factor four, "the effect of the use upon the potential market," Google's use of Java in Android delivered to Oracle exactly the business benefit it hoped for: a massive increase in the number of programmers familiar with and

[13] Order Denying Renewed Motion for Judgment as a Matter of Law and Motion for a New Trial of the United States District Court for the Northern District of California, (September 27, 2016), App. at 115a (emphasis added).

using Java.[14] It was for this reason that the then-CEO of Sun, Jonathan Schwartz, explained to the jury that Android would "strap[] another set of rockets to the community's momentum." JA 132. Mr. Schwartz further explained:

> As we promoted that language and as we promoted that technology, that created – that opened that market that historically we couldn't have gone after. But if you were using Java, then everything else that [Oracle] sold we could sell to you. [15]

Even disregarding Google's reasonable reliance on Oracle's public statements, the Federal Circuit's fair use analysis missed the significance of Oracle's actions. Google's use was explicitly contemplated and anticipated by Oracle, and was consistent with Oracle's goals in making the work available under the GPL.

It is not surprising that Oracle would have preferred to receive these benefits in addition to payment from Google—just as it is asking here. But open source licenses present a tradeoff: give up control and direct licensing revenue in order to reach a broader market. It is unreasonable to put a work on the market with an open source license, and then complain to the courts when the work succeeds in reaching a broader market.

[14] *See generally, supra*, Part II.B

[15] JA 122.

## III. The implications of this case go far beyond the companies at issue and affect the software industry generally and open source software in particular

Google's brief highlights the stakes for the two companies at issue. But *amici* want to emphasize that the effects of this decision go far beyond the two parties.

The Federal Circuit's decisions are destabilizing because they upset the settled expectations of thousands of software developers—and particularly open source software licensees—across all aspects of the economy. If these Federal Circuit decisions are allowed to stand, an immediate result will be the imposition of copyright on functional software elements—elements which had been previously understood to be purely excluded from copyright under Section 102(b) of the Copyright Act.

The open source licensing context is important because open source usage has become ubiquitous in software development. Recent estimates indicate that 98% of all software includes one or more open source-licensed components.[16] Changes to established interpretations of open

[16] See 2016 Global Developer Report, Gitlab, Inc. (2016) page.gitlab.com/2016-Developer-Survey_2016-Developer-Survey.html; Alan Reynolds, Microsoft's Acquisition of GitHub Is Not 'Anticompetitive', CATO Institute (Jul. 18, 2018), www.cato.org/publications/commentary/microsofts-acquisition-github-not-anticompetitive.

source licenses and the permissions they grant to licensees will have effects far beyond this case.

Open source licensees depend on stable interpretations of what is “functional” and what is “expressive.” The ability to reimplement APIs, including in commercial contexts, has long been considered “fair game”—and fair use—if not excluded from copyrightability completely under the merger doctrine or as a “useful article.” Millions of lines of open source code, with commercial value in the hundreds of billions of dollars, have been written with these settled expectations in mind.

## CONCLUSION

The Federal Circuit's decisions failed to deal appropriately with the software in this case. The Court should reverse the decisions of the Federal Circuit to prevent copyright holders from using copyright to gain patent-like control over functionality.

Respectfully submitted,

VAN LINDBERG
JILL WHEATON*
DYKEMA GOSSETT PLLC
112 E. Pecan Street
Suite 1800
San Antonio, TX, 78215
vlindberg@dykema.com
jwheaton@dykema.com
Ph. (210) 554-5294
*Counsel for Amici Curiae Python Software Foundation, Open UK, and Protocol Labs, Inc.*

LUIS VILLA
TIDELIFT, INC.
50 Milk Street
16th Floor
Boston, MA 02109
luis@tidelift.com
*Counsel for Amicus Curiae Tidelift, Inc.*

** Counsel of Record*

January 13, 2020