TRIGGERING INFECTION: DISTRIBUTION AND DERIVATIVE WORKS UNDER THE GNU GENERAL PUBLIC LICENSE

Theresa Gue†

Abstract

Imagine that Microsoft discovers that its profitable program, Microsoft Word, contains lines of code borrowed from a free and open source software program. Further imagine that as a result of this oversight, all users of Microsoft Word now have a license to freely distribute, reproduce, and modify Word, and Microsoft is required to provide the source code to users in order to facilitate such actions. This is the exact scenario envisioned and feared by many corporations today. It is also the reason why the GNU General Public License ("GPL"), the most popular free and open source software license in the world, is also the most feared. The license contains a viral "copyleft" provision, which requires that all derivative works of a GPL-covered work that are distributed be licensed under the GPL or a compatible license. Copyleft can eviscerate the value of proprietary software by granting all users of the software the freedom to modify, distribute, copy, and reproduce the software.

Much uncertainty shrouds the copyleft clause and its twin triggers of distribution and derivative works—so much so that attorney Lawrence Rosen likened public reaction to the GPL to the early days of the AIDS epidemic.[1] *While scholars have explored derivative works under the GPL, the topic of distribution has not been thoroughly discussed. This Article seeks to fill the void by offering a comprehensive overview of both distribution and derivative works. It advocates applying copyright law's limited publication doctrine to distinguish between limited publications that should not trigger copyleft and distributions to the public that should trigger copyleft. Courts developed the limited publication doctrine to mitigate the effects of authors losing their copyrights by publishing without notice. The doctrine provides a useful framework for concluding that scenarios such as cloud-computing and outsourcing should not constitute distributions. The Article next analyzes*

† Judicial law clerk to the Honorable Legrome D. Davis of the U.S. District Court for the Eastern District of Pennsylvania. I would like to thank Nader Mousavi for his guidance and input on this piece.

[1] *See generally* Lawrence Rosen, *The Unreasonable Fear of Infection*, ROSENLAW & EINSCHLAG (2001), http://www.rosenlaw.com/html/GPL.PDF.

whether situations such as static linking, dynamic linking, use of Linux kernel modules, and intermingling software in a virtual cloud computing environment create derivative works. The Article offers advice for using free and open source software without infecting proprietary software, and seeks to be a practical guide that enables individuals and companies alike to take advantage of the benefits of free and open source software with ease of mind—and without fear of infection.

TABLE OF CONTENTS

I. INTRODUCTION

Free and open source software (FOSS)[2] is software distributed under a license that grants users the freedom to exercise many of the exclusive rights normally reserved to the copyright owner by the Copyright Act, such as the rights to modify, distribute, and copy the software.[3] The GNU General Public License, or GPL for short, is the most popular of such licenses. FOSS has been adopted by companies due to its ease of use and adaptability. It has been embraced by programmers because it allows users to adapt software to their own needs, encourages innovation, and fosters a community of openness and cooperation.[4]

However, such freedom comes at a cost. The GPL contains an infectious provision requiring any user who creates and distributes a work based on a GPL-covered work to license the new work under the GPL or an equivalent license that grants users the right to modify, reproduce, and distribute that work.[5] Copyleft has been both celebrated and reviled. For supports of FOSS, copyleft gives the GPL teeth, ensuring that improvements built on FOSS remain free and forcing users who benefit from GPL-covered software to contribute improvements back to the FOSS community.[6] For commercial software vendors, copyleft is a danger that threatens to strip software of its

[2] FOSS refers to software that is both free and open source. The GPL is one such license. Though the technical definitions of open source software and free software differ slightly—*compare The Free Software Definition*, GNU OPERATING SYSTEMS, http://www.gnu.org/philosophy/free-sw.html (last visited Feb. 5, 2012) (describing free software) *with The Open Source Definition*, OPEN SOURCE INITIATIVE, http://www.opensource.org/docs/osd (last visited Feb. 5, 2012) (describing open source software)—the two terms describe nearly the same category of software. *See* Joseph Feller et al., *Introduction* to PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE, xi, xvii (Joseph Fuller et al. eds.) (2005), *available at* http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=10477&mode=toc (noting that the definitions of free software and open source software "are substantively identical"); Richard Stallman, *Why Open Source Misses the Point of Free Software*, GNU OPERATING SYSTEMS (Sept. 20, 2011, 8:15 AM), http://www.gnu.org/philosophy/open-source-misses-the-point.html ("The two terms describe almost the same category of software"); Posting of Eliah Kagan to *Is There a Difference Between Free Software and Open Source Software*, ASK UBUNTU (Nov. 14, 2011, 22:36), http://askubuntu.com/questions/78958/is-there-a-difference-between-free-software-and-open-source-software ("The terms free software and open source software do mean different things, though the categories of software they refer to are almost exactly the same."). The primary distinction between the terms is a difference in philosophies behind the two movements. The free software movement believes that the freedoms to modify, reproduce, distribute, and run software are inherently important and should be respected. That is, this movement believes such freedoms are an end in themselves. The open source movement, on the other hand, sees these freedoms as a means to achieving a superior product. The open source movement focuses on open source software as a technically superior alternative to proprietary software. This Article will use the philosophically neutral term FOSS to refer to both free and open source software.

[3] *See* 17 U.S.C. §§ 106 (2006).

[4] Richard Stallman, *Why Software Should Not Have Owners*, GNU OPERATING SYSTEM (Sept. 20, 2011, 8:15 AM), http://www.gnu.org/philosophy/why-free.html ("You deserve to be able to cooperate openly and freely with other people who use software. You deserve to be able to learn how the software works, and to teach your students with it. You deserve to be able to hire your favorite programmer to fix it when it breaks. You deserve free software.").

[5] *What is Copyleft?*, GNU OPERATING SYSTEM, http://www.gnu.org/copyleft/ (last visited Feb. 7, 2012) ("Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom.").

[6] *What is Free Software?*, GNU OPERATING SYSTEM, http://www.gnu.org/philosophy/free-sw.html (last visited Feb. 7, 2012). At the very least, those who make modifications to FOSS and distribute those modifications must contribute improvements back to the community.

proprietary nature and, correspondingly, much of its value. Microsoft, for example, would be crippled if users could freely copy, distribute, and modify Windows. Microsoft CEO Steve Ballmer articulated the fear held by many commercial software companies when he called Linux, a popular FOSS operating system, "a cancer that attaches itself in an intellectual property sense to everything it touches."[7]

Copyleft is triggered by distribution and creating a derivative work; therefore, companies must be careful of engaging in these two acts. Unfortunately, what constitutes a distribution or a derivative work under the GPL is uncertain terrain. Derivative works have garnered a fair amount of attention in the academic world; however, distribution has not been thoroughly explored.[8] Courts have yet to weigh in on either issue. The analysis is further complicated by the fact that software in general, and FOSS in particular, maps imperfectly onto traditional copyright law. The result is scant understanding in the tremendously popular area of FOSS—with potentially devastating consequences for companies using FOSS in developing proprietary software.

This Article attempts to fill the void by offering an overview of copyright law and applying it to determine which situations trigger the GPL's copyleft. It advocates using copyright law's limited publication doctrine to determine what constitutes a distribution under the GPL, a suggestion that has been briefly mentioned by commentators but never explored in detail.

Part II introduces the GPL. Part III provides background on software under copyright law. Part IV examines distribution, analyzing distribution in the language of the GPL itself, in copyright law, and finally offering conclusions as to what triggers copyleft. In particular, Part IV looks at unexplored questions such as whether cloud computing and outsourcing trigger copyleft. Part V analyzes derivative works in the language of the license and under copyright law before discussing which uses of GPL-covered software are considered derivative works. Part VI offers recommendations for companies to avoid triggering copyleft and concludes.

II. THE GPL

Richard Stallman started the free software movement while working as a computer programmer at the Massachusetts Institute of Technology.[9] Stallman fixed a printer jam problem in the computer lab by toying with the printers'

[7] Thomas C. Greene, *Ballmer: Linux is a Cancer, Contaminates All Other Software with Hippie GPL Rubbish*, REGISTER (June 2, 2001 6:19 PM), http://www.theregister.co.uk/2001/06/02/ballmer_linux_is_a_cancer/ (internal quotation marks omitted).

[8] *See, e.g.*, Mitchell L. Stoltz, Note, *The Penguin Paradox: How the Scope of Derivative Works in Copyright Affects the Effectiveness of the Gnu GPL*, 85 B.U. L. REV. 1439, 1476 (2005) (exploring the policy implications of a permissive stance towards derivative works under the GPL).

[9] SAM WILLIAMS, FREE AS IN FREEDOM: RICHARD STALLMAN'S CRUSADE FOR FREE SOFTWARE 1–2 (2002), *available at* http://oreilly.com/openbook/freedom/ch01.html. Source code is computer programming language that specifies actions a computer should perform. It stands in contrast to object code, which is binary language (comprised of 0s and 1s) that a computer can read and execute. A computer translates source code to object code prior to performing actions.

source code.[10] However, when he attempted the same fix on a Xerox printer, Xerox refused to provide the source code.[11] In response, Stallman founded the Free Software Foundation ("FSF"), a non-profit foundation dedicated to promoting free software, and developed the GPL.[12]

The GPL is the most popular FOSS software license in the world today, with the majority of all FOSS licensed under the GPL.[13] As of February 2012, 42% of all open source projects were licensed under GPLv2, while 6% were licensed under GPLv3.[14] The popular programs Linux and MySQL are licensed under the GPL.[15] Linux is a computing behemoth, an operating system that powers many of the servers on the Internet[16] and serves as the basis for Android, the fastest-growing mobile platform in the world today.[17] MySQL is the world's most popular FOSS database, with over 65,000 downloads a day.[18]

The GPL guarantees four freedoms:

> The freedom to run the program, for any purpose (freedom 0).
>
> The freedom to study how the program works, and change it to make it do what you wish (freedom 1). Access to the source code is a precondition for this.
>
> The freedom to redistribute copies so you can help your neighbor (freedom 2).
>
> The freedom to distribute copies of your modified versions to others (freedom 3). By doing this you can give the whole community a chance to benefit from your changes. Access to the source code is a precondition for this.[19]

The GPL, and FOSS software as a whole, contrasts with proprietary software such as Microsoft Word, which does not provide users with source code and prohibits users from copying, reproducing, modifying, and distributing the software.[20]

A. The GPL's Copyleft Provision

The GPL employs an infectious provision known as copyleft. Copyleft

[10] *Id.*

[11] *Id.*

[12] John Tsai, Note, *For Better or Worse: Introducing the GNU General Public License Version 3*, 23 BERKELEY TECH. L.J. 547, 550 (2008).

[13] *Top 20 Most Commonly Used Licenses in Open Source Projects*, BLACK DUCK OPEN SOURCE RESOURCE CENTER, http://www.blackducksoftware.com/oss/licenses/ (lasted visited Feb.7, 2012).

[14] *Id.*

[15] *General Public Software, The GPL/GNU*, DIGITAL EUROPEAN RIGHTS, http://eurorights.org/general-public-licensed-software-ae%E2%80%9C-gpl-gnu.html (last visited June 1, 2011).

[16] Sam Dean, *Linux is Growing Fast on Servers, and Red Hat Benefits*, OSTATIC (Apr. 29, 2011), http://ostatic.com/blog/linux-is-growing-fast-on-servers-and-red-hat-benefits.

[17] Trent Nouveau, *Google: Android "Fastest-Growing" Mobile Platform in the World*, TG DAILY (Feb. 15, 2011), http://www.tgdaily.com/mobility-features/54134-google-android-fastest-growing-mobile-platform-in-the-world.

[18] *Market Share*, MYSQL, http://www.mysql.com/why-mysql/marketshare/ (last visited June 1, 2011).

[19] *What is Free Software?*, *supra* note 6.

[20] *See* 17 U.S.C. § 106 (2006) (setting forth the exclusive rights for owners of a copyrighted work).

requires that all programs based on GPL-covered works and distributed to others be subject to the same basic four freedoms as the initial GPL-covered work and be accompanied by a license giving notice of these freedoms.[21] The GPL uses copyright law to enforce its provisions.

Copyright law prohibits anyone other than the copyright owner from exercising the exclusive rights of copying, reproduction, adaptation, and distribution.[22] Infringers face the prospect of lawsuits resulting in damages and/or injunctions, in addition to litigation expenses.[23] The GPL allows users to exercise these exclusive rights for GPL-licensed works, thus offering a safe harbor from the threat of copyright litigation. However, users must follow all provisions in the GPL in order to take advantage of the safe harbor.[24] The copyleft provision requires that any derivative works must be subject to the same freedoms of the original work and covered by the GPL.[25] Thus, the user of a GPL-covered work has three choices: (1) refrain from exercising the exclusive rights; (2) utilize the license to exercise the exclusive rights, including following the copyleft provision; or (3) exercise the exclusive rights without following all provisions of the GPL and face potential litigation.[26]

The copyleft provision is limited to derivative works distributed to the public. The GPL applies only to works licensed under the GPL and all derivative works;[27] thus a user who does not create a derivative work does not have to comply with copyleft. Furthermore, copyleft is triggered only when a covered or derivative work is distributed:[28] individuals can freely modify, copy, and run a covered or derivative work without providing the source code if they never distribute it.[29] Since distribution and derivation trigger the GPL's copyleft, the breadth of the twin concepts directly determines the effectiveness of the GPL. This Article explores these concepts in detail.

[21] *What is Copyleft?*, *supra* note 5.

[22] *See* 17 U.S.C. § 106 (listing exclusive rights for copyright owners).

[23] *Id.* §§ 502, 504, 505.

[24] *See What is Copyleft?, supra* note 5.

[25] *Id.*

[26] *See* GNU GENERAL PUBLIC LICENSE: VERSION 3 § 9 (June 29, 2007), *available at* http://www.gnu.org/licenses/gpl.txt [hereinafter GPLv3] ("[N]othing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License." *See also* Lothar Determann, *Dangerous Liaisons—Software Combinations as Derivative Works? Distribution, Installation, and Execution of Linked Programs Under Copyright Law, Commercial Licenses, and the GPL*, 21 BERKELEY TECH. L.J. 1421, 1481–82 (2006) ("Anybody who distributes software outside the scope of the applicable license agreements lacks a valid authorization required by the Copyright Act 231 and thus commits copyright infringement."). It is unclear whether litigation for violations of the GPL would arise under contract law or copyright law. Even under a contract theory, the GPL would work against a background of copyright law, allowing parties to privately contract in order to exercise exclusive rights to copyrighted works. *See infra* Part B for a discussion of the contract versus license debate.

[27] *What is Copyleft?, supra* note 5.

[28] *Id.*

[29] *See Frequently Asked Questions about the GNU Licenses*, GNU OPERATING SYSTEMS, http://www.gnu.org/licenses/gpl-faq.html (last visited Jan. 20, 2012) ("You are free to make modifications and use them privately, without ever releasing them . . . . But if you release the modified version to the public in some way, the GPL requires you to make the modified source code available to the program's users, under the GPL.").

B. Contract or License?

Two differing schools of thought underlie the GPL's infectious provision. Under the license theory, the GPL is a grant of permission that allows users to exercise the exclusive rights otherwise reserved to copyright owners. Anyone who violates the license cannot take advantage of its permissions and may be liable for copyright infringement. The alternative theory holds that the GPL should be interpreted as a contract.

Case law on the issue has been scant. In the first two cases dealing with violations of the GPL, the parties argued for a contract theory. In *Progress Software v. MySQL*, MySQL asked for a preliminary injunction to stop Progress Software from distributing MySQL and using the MySQL trademark.[30] MySQL argued, among other things, that Progress Software breached a contract when they distributed MySQL with proprietary software but refused to release source code for the proprietary software, thus violating the terms of the GPL.[31] The court accepted the contract theory without discussion, but denied the preliminary injunction because MySQL failed to show irreparable harm or likelihood of success on the merits.[32] In *MontaVista Software, Inc. v. Lineo, Inc.*, the parties alleged violations of the GPL, arguing for breach of contract in their complaints.[33] The case settled before trial.[34]

A few years later, the Federal Circuit examined the Artistic License in the seminal case of *Jacobsen v. Katzer* and determined that the plaintiff had a right to bring a suit of copyright infringement against the defendant.[35] The court used California contract law principles to determine that the restrictions imposed by the Artistic License were conditions to a license, rather than covenants to a contract.[36] The decision lends support to the license theory. However, because the court examined the Artistic License rather than the GPL and based its determinations on an analysis of California law, whether the GPL is a license or a contract still remains to be seen and may change depending on what state's law is applied.

The Free Software Foundation (FSF) maintains that the GPL is a bare license,[37] while commentators diverge on which theory they support.[38]

[30] Progress Software Corp. v. MySQL AB, 195 F. Supp. 2d 328, 329 (D. Mass. 2002). Progress sued MySQL for breach of contract, among other claims. *Id.*

[31] Plaintiff's Reply in Further Opposition to MySQL AB's Motion for a Preliminary Injunction at 1 Progress Software Corp. v. MySQL AB, 195 F. Supp. 2d 328, 329 (D. Mass. 2002) (No. 1:01-cv-11031) ("MySQL . . . is pursuing a contractual right. In abandoning its copyright infringement theory, MySQL abandons copyright presumptions and remedies . . . .").

[32] *Progress Softwar*e, 195 F. Supp. 2d at 329.

[33] Douglas A. Hass, *A Gentlemen's Agreement Assessing the Gnu General Public License and Its Adaptation to Linux*, 6 CHI.-KENT J. INTELL. PROP. 213, 230 (2007) (citing First Amended Complaint of Plaintiff, MontaVista Software, Inc. v. Lineo, Inc. No. 2-02 CV-00309J (D. Utah July 23, 2002).

[34] *Id*. at 230 n.169.

[35] Jacobsen v. Katzer, 535 F.3d 1373, 1382–83 (Fed. Cir. 2008).

[36] *Id.* at 1381–82. *See also* Marie Schiebert, Jacobsen v. Ketzer: *Precedent for Open Source Licensing Disputes*, INTERNETLAWBITS (Jan. 17, 2010), http://www.internetlawbits.com/2010/01/jacobsen-v-katzer-precedent-for-open-source-licensing-disputes/.

[37] Peter Brown, Free Software Foundation's Executive Director, has stated, "The GPL is a software license, it is not a contract. It gives permissions from the copyright holder." Matt Lee, *The GPL Tested in U.S.*

Alternatively, many academics maintain that if a FOSS license does not impose any obligations on the user but only allows the user to exercise the exclusive rights under copyright law, it will be construed as a license.[39] If additional obligations are imposed on the user, courts will adopt a contract theory. This reasoning has led some commentators to conclude that GPLv2 should be analyzed as a license, while GPLv3 should be analyzed as a contract because it requires users to do additional affirmative actions such as provide a license on essential patents to downstream users.[40]

Applying a contract theory differs in several respects from applying a license theory. Generally, copyright owners prefer the uniformity, minimal requirements, and stronger remedies of a license theory. Contracts are governed by state law, which can vary widely state to state.[41] Copyright is governed under federal law and has a higher degree of uniformity, both nationally and internationally.[42] A contract requires offer, acceptance, and consideration, whereas a license does not.[43]

It is unclear whether offer and acceptance are present when a user downloads software licensed under the GPL. A user who downloads software covered by the GPL typically receives the GPL in a text file during the download, along with a notice at the beginning of the source code indicating that the program is licensed under the GPL.[44] This is similar to shrink-wrap licenses used by commercial software companies. Shrink-wrap licenses are

Courts, Wallace v. FSF, FREE SOFTWARE FOUNDATION (Mar. 22, 2006), http://www.fsf.org/news/wallace-vs-fsf.

[38] *Compare* LAWRENCE ROSEN, OPEN SOURCE LICENSING: SOFTWARE FREEDOM AND INTELLECTUAL PROPERTY LAW 138 (2005) ("[T]his GPL reliance entirely on copyright law for license enforcement is legally sound."), *and* DAVID MCGOWAN, LEGAL ASPECTS OF FREE AND OPEN SOURCE SOFTWARE, PERSPECTIVES ON FREE AND OPEN SOURCE SOFTWARE 368 (Joseph Feller et al. eds., 2007), *available at* http://mitpress.mit.edu/books/chapters/0262562278chap19.pdf ("Because [the GPL] demands no bargain, one could argue that the GPL cannot form a 'real' contract."), *and* Michael F. Morgan, *The Cathedral and the Bizarre: An Examination of the "Viral" Aspects of the GPL*, 27 J. MARSHALL J. COMPUTER & INFO. L. 349, 391 ("[I]t will be assumed that the GPL is a bare copyright license and the analysis of the scope of the viral provisions of the GPL will be done on that basis."), *and* Greg R. Vetter, *Infectious Open Source Software: Spreading Incentives or Promoting Resistance?*, 36 RUTGERS L.J. 53, 84–85 (2004) ("The GPL specifically acknowledges that its licensing power derives from copyright in the software . . . . Accordingly, it does not necessarily need to rely on contract to hold licensees to the conditions underlying the permission."), *and* Stoltz, *supra* note 8, at 1447 ("[T]he GPL can derive its legal force exclusively from the Copyright Act, with no resort to contract law."), *and* Pamela Jones, *The GPL is a License, Not a Contract*, LWN.NET (Dec. 3, 2003), http://lwn.net/Articles/61292/ ("Because the GPL does not require any promises in return from licensees, it does not need contract enforcement in order to work"), *with* Robert A. Hillman and Maureen A. O'Rourke, *Rethinking Consideration in the Electronic Age*, 61 HASTINGS L.J. 311, 335 (2009) ("Open source licenses are contracts."), *and* Sapna Kumar, *Enforcing the Gnu GPL*, 2006 U. ILL. J.L. TECH. & POL'Y 1, 35 (2006) (claiming that the GPL is a failed contract lacking consideration and therefore should be enforced under a theory of promissory estoppel), *and* Jason B. Wacha, *Is the GPL Enforceable?*, 21 SANTA CLARA COMPUTER & HIGH TECH. L.J. 451, 456 (2005) ("It is likely that a court, in the U.S. or abroad, would recognize the GPL as a contract.").

[39] *See, e.g.*, ROSEN, *supra* note 38, at 138.

[40] Hillman & O'Rourke, *supra* note 38, at 326.

[41] Jones, *supra* note 38.

[42] *See id*.

[43] Hersh R. Reddy, Jacobsen v. Katzer: *The Federal Circuit Weighs in on the Enforceability of Free and Open Source Software Licenses*, 24 BERKELEY TECH. L.J. 299, 307 (2009).

[44] *How to Use GNU Licenses for Your Own Software*, GNU OPERATING SYSTEM, http://www.gnu.org/licenses/gpl-howto.html (last visited Jan. 22, 2012).

composed of a notice on the outside of the software package and an actual license within the package that dictates the terms of use.[45] Those two components, along with a user's right to return the software if he or she disagrees with the terms, were sufficient for the Seventh Circuit to find offer and acceptance in *ProCD v. Zeidenberg*.[46] Although FOSS licensing is similar to shrink-wrap licenses, it remains to be seen whether courts will uphold offer and acceptance under the GPL. Furthermore, while the user receives valuable software and source code, it is unclear whether the licensor receives any consideration in return.

In terms of remedies, the typical relief a plaintiff can receive from breach of contract is damages.[47] Specific performance may arise, but only if the plaintiff shows that he or she will suffer irreparable harm, which can be difficult.[48] On the other hand, violation of the license results in copyright infringement, which typically leads to the presumption of irreparable harm needed for an injunction.[49] Therefore, plaintiffs can more easily obtain an injunction preventing a defendant from committing further violations under a license theory. Copyright infringement can also be a basis for awarding statutory damages.[50]

Unfortunately, even the most extensive remedies typically offered for both breach of contract and copyright infringement do not cover the robust relief outlined by the GPL—namely that proprietary software becomes infected and loses its proprietary nature.[51] In fact, this disjunction has led some commentators to posit that the GPL's infectious terms cannot be upheld.[52] The typical remedies for both breach of contract and copyright infringement are damages.[53] An injunction may be granted in some instances. Specific performance dictating that an individual release proprietary software as FOSS may lie outside the scope of these typical remedies. If this theory is true, the GPL would lose much of its efficacy as its viral terms could not be enforced in a court of law. However, other commentators believe that Section 103 of the Copyright Act results in consequences that are almost as severe as enforcement of the viral provision.[54] Section 103 states:

> The subject matter of copyright as specified by section 102 includes compilations and derivative works, but protection for a work

[45] Reddy, *supra* note 43, at 309.

[46] ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1452 (7th Cir. 1996).

[47] RESTATEMENT (SECOND) OF CONTRACTS § 346 (1981).

[48] *Id.* § 357.

[49] Jacobsen v. Katzer, 535 F.3d 1373, 1378 (Fed. Cir. 2008) ("In cases involving copyright claims, where a copyright holder has shown likelihood of success on the merits of a copyright infringement claim, the Ninth Circuit has held that irreparable harm is presumed.").

[50] 17 U.S.C. § 504(a) (2006).

[51] *See* Jones, *supra* note 38.

[52] *Id.* ("Similarly, when you hear that the GPL is viral and can force proprietary code to become GPL, which a couple of lawyers have been saying, you'll know that isn't true. If you steal GPL code, you can expect an enforcement action. But this action can only be enforcement of a license, not a contract, and a forced release under the GPL can't be imposed on you under copyright law.").

[53] *See supra* notes 47–50 and accompanying text.

[54] *See* Jack E. Brown, *"Analytical Dissection" of Copyrighted Computer Software – Complicating the Simple and Confounding the Complex*, 25 ARIZ. ST. L.J. 801, 830–39 (1993).

> employing preexisting material in which copyright subsists does not extend to any part of the work in which such material has been used unlawfully.[55]

Thus, if an infringer uses GPL-covered software in a derivative work but does not release that work under the GPL, he or she has used that material unlawfully and does not have a valid copyright in that work. Commentators have argued that any portions of the derivative work that unlawfully incorporate a GPL-covered work could potentially fall into the public domain, with the result that users with access to either the source or binary versions of the software can freely use, copy, distribute, and modify the software with impunity.[56] This is a potentially crippling problem for commercial software businesses. Another possible scenario is that the original copyright holder of the GPL-covered work now owns the copyright in the derivative work, due to his or her exclusive right to make derivative works and the user's forfeiture.[57] This would be equally devastating to proprietary software vendors, as the original copyright holder would likely license the software under the GPL in such a scenario, bringing about the very result originally envisioned by the license.

These two possibilities would arise in the most extreme scenario. Since the Copyright Act states that copyright protection "does not extend to any part of the work in which such material has been used unlawfully,"[58] an infringer may be able to remove the infringing portions of code from his or her software (particularly if it is a discrete portion), rewrite that portion, and maintain a copyright in the entire piece. Most likely, a court would deny copyright protection to the whole work only in cases where the infringing work pervades the entire derivative work.[59] Regardless of whether a contract or license theory is used, a close examination of the Copyright Act and case law interpreting it is important. Under a license theory, copyright law would be the only source of authority, and a copyright owner's wrongful interpretation of law would have no authority. For example, if the GPL's Frequently Asked Questions ("FAQ"), which are written by the Free Software Foundation, assert that something is a derivative work while case law does not, a license theory would follow the case law and determine that such a use does not infringe the exclusive right to

[55] 17 U.S.C. § 103(a).

[56] *See, e.g.*, *What is Copyleft?*, *supra* note 6. *Cf.* Anderson v. Stallone, 11 U.S.P.Q.2d (BNA) 1161 (1989) (holding that plaintiff could not gain copyright protection for any part of his unauthorized derivative work but not explicitly stating that the unauthorized work fell into the public domain as a result).

[57] This is, admittedly, an unlikely outcome.

[58] 17 U.S.C. § 103(a).

[58] 17 U.S.C. § 103(a).

[59] *See* 1 MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 3.06 (Matthew Bender rev. ed 2011) ("The case that most thoroughly ventilates this doctrine is *Pickett v. Prince*, in which a fan made a guitar in the shape of the symbol that serves as the name for The Artist Formerly Known As Prince . . . . Given that the underlying work "pervaded" the derivative work, the latter could not sustain independent copyright protection." (citing Pickett v. Prince, 207 F.3d 402 (7th Cir. 2000))); Mark A. Lemley, *The Economics of Improvement in Intellectual Property Law*, 75 TEX. L. REV. 989, 1022 (1997) ("In other cases, however, where the material is inextricably intertwined, the improver loses the entire value of her contribution. This is the case with regard to movies or plays based on a book. It is also likely to be the case with many improvements in the field of computer software.").

prepare derivative works. Under a contract theory, because the intent of the parties are significant, interpretations such as the FAQ likely have more weight. However, because the GPL relies heavily on copyright law to define certain terms, copyright law remains important. For purposes of this paper, I will assume a license theory rather than a contract theory.

III. SOFTWARE UNDER COPYRIGHT LAW

Despite the relevance of copyright law to an analysis of the GPL, software in general does not fit cleanly into the conceptual framework laid out by the Copyright Act. Computer programs must be copied and installed onto a user's computer before use and may need to be adapted in order to be used on an individual's computer. Under traditional copyright law, every user of a computer program would necessarily infringe the exclusive right to copy. Section 117 of the Copyright Act clarifies that taking such actions does not constitute copyright infringement.[60] Second, copyright law protects expressive rather than functional elements. Since software is utilitarian rather than artistic, it is difficult to fit into a copyright framework. Courts have accordingly adopted filtration tests to examine copyright infringement in the software context. These tests filter out strictly functional elements, allowing courts to compare only the remaining non-functional elements for substantial similarity to determine whether infringement has occurred.[61]

Software also depends on compatibility—with other programs, operating systems, device drivers, and the like. In fact, "customers will often choose a software product not simply because it is technically superior, but because it is compatible with software they already own."[62] If courts adopt an expansive definition of derivative work (for example, one that encompasses add-on modules or compatible interfaces), they may deter the creation of compatible software, or diminish competition by making it economically viable for only the owners of copyrighted software to create add-ons or compatible programs, to the detriment of customers and the public as a whole.[63]

The various problems involved with applying copyright law to software have caused courts to liken the problem to "fit[ting] the proverbial square peg in a round hole."[64] Nevertheless, because distribution and derivative works are terms borrowed directly from copyright law—and because the GPL relies on copyright law for copyleft—copyright law remains relevant to this analysis.

[60] 17 U.S.C. § 117.

[61] *See* Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 706–12 (2d Cir. 1992) for a description of the Abstraction-Filtration-Comparison Test. See *infra* Part 2 for an in-depth discussion of filtration tests.

[62] Stoltz, *supra* note 8, at 1461.

[63] *See id.* at 1458.

[64] *Altai*, 982 F.2d at 712. *See also* Sega Enterprises Ltd. v. Accolade, Inc., 977 F.2d 1510, 1527 (9th Cir. 1992) ("We must avoid the temptation of trying to force 'the proverbial square peg in[to] a round hole.'" (quoting Computer Assocs. Int'l v. Altai, 982 F.2d 693, 712 (2d Cir. 1992))).

IV. Distribution

A. Distribution in the Language of the GPL

Both GPLv1 and GPLv2 use the term "distribute," but neither license defines the term.[65] According to the FSF, GPLv1 and GPLv2 do not define "distribute" because the authors assumed courts would adopt the definition of "distribute" from U.S. copyright law.[66] GPLv3 abandons the term "distribute" altogether in favor of the terms "propagate" and "convey."[67] Under GPLv3:

> To 'propagate' a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.[68]
>
> To 'convey' a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.[69]

Commentators diverge as to whether GPLv3's adoption of the term "convey" makes U.S. copyright law irrelevant. On the one hand, FSF asserts that "convey" in version 3 "more or less" means the same as version 2's "distribute."[70] The FSF changed "distribute" to "propagate" in GPLv3 after learning that jurisdictions other than the United States also use the term "distribute" in their copyright laws, which led to inconsistent rulings.[71] They adopted the term "convey" to ensure uniformity.[72] This approach suggests that U.S. copyright law still informs the definition of "convey" in GPLv3—that courts simply should look *only* to U.S. copyright law and not law of other jurisdictions.

Other commentators believe that abandoning the term "distribute" means that copyright law's definitions of distribution have no bearing on the GPL. According to Eben Moglen, Chairman of the Software Freedom Law Center,

[65] *See, e.g.*, GNU General Public License: Version 2 (June 1991), *available at* http://www.gnu.org/licenses/gpl-2.0.html [hereinafter GPLv2].

[66] Brett Smith, *A Quick Guide to GPLv3*, GNU Operating Systems, http://www.gnu.org/licenses/quick-guide-gplv3.html (last visited Feb. 20, 2012) ("The license never says what distribution is, because the term was borrowed from United States copyright law. We expected that judges would look there for the definition.").

[67] *See generally* GPLv3, *supra* note 26.

[68] *Id.* § 0.

[69] *Id.*

[70] *Frequently Asked Questions About the GNU Licenses*, *supra* note 29.

[71] See *id.* ("During the course of enforcing GPLv2, we learned that some jurisdictions used the word 'distribute' in their own copyright laws, but gave it different meanings. We invented a new term to make our intent clear and avoid any problems that could be caused by these differences.").

[72] *See id.*

"This strategy, of using a new term of art, 'to propagate software,' not reflected in any particular copyright statute, reflects an overall drafting decision to attempt to cut the language of the license loose from any particular system's copyright law."[73]

Regardless, because the term "distribute" comes from copyright law and because copyleft works against a background of copyright law, a look at distribution under copyright law is appropriate.

B. Distribution Under Copyright Law

The Copyright Act gives a copyright owner the exclusive right "to distribute copies or phonorecords of the copyrighted work to the public by sale or other transfer of ownership, or by rental, lease, or lending."[74] However, the Copyright Act does not define "distribute," which has led to much confusion in the courts, as detailed below. But the Copyright Act does define publication. Publication is:

> [T]he distribution of copies or phonorecords of a work to the public by sale or other transfer of ownership, or by rental, lease, or lending. The offering to distribute copies or phonorecords to a group of persons for purposes of further distribution, public performance, or public display, constitutes publication. A public performance or display of a work does not of itself constitute publication.[75]

Despite the lack of definition in the Copyright Act, a few basic principles underlie the concept of distribution. At the most elementary level, selling a physical copy of an infringing work constitutes distribution.[76] Software usage has somewhat complicated this analysis because a physical copy is not necessary for others to obtain infringing material. Courts have clarified that transferring an electronic copy constitutes distribution, despite the fact that case law has traditionally envisioned physical copies and despite the fact that an owner need not part with a copy in order to distribute it.[77]

1. The Making Available Debate

The "making available to the public" debate asks whether making a copyrighted work available to the public constitutes distribution, even when there is no evidence that a member of the public actually obtains a copy. In

[73] Eben Moglen, General Counsel, FSF, Chairman, SFLC, Opening Session of First International GPLv3 Conference (Jan. 16, 2006) (transcript *available at* http://www.ifso.ie/documents/gplv3-launch-2006-01-16.html). *See also* Tsai, *supra* note 12, at 564 n.111 ("[The change in terminology] means that lawyers can no longer rely on U.S. case law and statutes to provide the meanings of the license terms.").

[74] 17 U.S.C. § 106(3) (2006).

[75] 17 U.S.C. § 101 (2006).

[76] Microsoft Corp. v. Lopez, No. 08-1743, 2009 WL 959219, at *1, *3 (W.D. Wash. Apr. 7, 2009) (holding that defendant infringed on the distribution right by selling counterfeit software through websites like Craigslist).

[77] *See, e.g.*, London-Sire Records v. Doe, 542 F. Supp. 2d. 153, 170 (D. Mass. 2008) ("[T]he Court concludes that § 106(3) confers on copyright owners the right to control purely electronic distributions of their work.").

other words, the debate questions whether distribution is the same as publication: whether "offering to distribute copies or phonorecords to a group of persons for purposes of further distribution, public performance, or public display," which constitutes publication, also constitutes distribution. This question gained particular relevance in the context of peer-to-peer networks such as Napster, where plaintiffs often had no record that a music file was actually transferred. There is currently a circuit split on the question.

Some courts find that distribution encompasses offering to distribute copies or phonorecords for purposes of further distribution. In *Hotaling v. Church of Jesus Christ of Latter-Day Saints*, a church unlawfully copied plaintiff's copyrighted work to distribute to its branch libraries.[78] The court held that a library distributes a published work when it has the work in its collection and makes it available to the public, even if there is no record that anyone actually viewed or checked out the illegal copy.[79]

In *A&M Record, Inc.* v. *Napster, Inc.*, record companies brought suit against defendant users of the file-sharing program Napster. [80] Napster created a centralized index of all the files in the shared folders.[81] Users did not upload files to Napster's server, but merely uploaded the name of the file.[82] Actual transfers occurred only when other users decided to obtain files.[83] The court held that "Napster users who upload file names to the search index for others to copy violate plaintiffs' distribution rights."[84]

Another compelling argument that offering to distribute constitutes distribution comes from Congress's 2005 amendment to Section 506 of the Copyright Act, which governs criminal copyright infringement. Congress added a new basis for criminal liability for "the distribution of a work being prepared for commercial distribution, by making it available on a computer network accessible to members of the public, if such person knew or should have known that the work was intended for commercial distribution."[85] This suggests Congressional intent to penalize users of peer-to-peer networks as copyright infringers.

Other courts and commentators find that distribution requires actual dissemination. Merely offering to distribute a copyrighted work, without an actual transfer or dissemination of a copyrighted work, does not infringe on the exclusive right to distribute. According to Nimmer, "[T]he right of distribution apparently is not infringed by the mere offer to distribute to members of the public."[86]

[78] Hotaling v. Church of Jesus Christ of Latter-Day Saints, 118 F.3d 199, 201–02 (4th Cir. 1997).

[79] *Id.* at 203.

[80] A&M Record, Inc. v. Napster, Inc., 239 F.3d 1004, 1010–11 (9th Cir. 2001).

[81] *Id.* at 1012.

[82] *Id.*

[83] *Id.*

[84] *Id.* at 1014.

[85] Family Entertainment and Copyright Act of 2005, Pub. L. No. 109-9, 119 Stat. 218 (codified as amended at 17 U.S.C. § 506 (a)(1)(c)).

[86] Elektra Entm't Group, Inc. v. Barker, 551 F. Supp. 2d 234, 243 (S.D.N.Y. 2008) (quoting MELVILLE B. NIMMER & DAVID NIMMER, NIMMER ON COPYRIGHT § 8.11[A] n.2 (2005)).

The court in *Capitol Records, Inc.* v. *Thomas* came to a similar conclusion.[87] Record companies sued defendants for illegally downloading and distributing music files using Kazaa.[88] The court held that offering to distribute a copyrighted work does not constitute distribution, for a number of reasons.[89] First, the plain language of the Copyright Act gives the owner the exclusive right to distribute by sale, transfer of ownership, rental, lease, or lending—it does not address offering to do these acts.[90] Second, the plain meaning of "distribute" entails a transfer.[91] Third, in the Copyright Act, publication is a distinct term that triggers its own set of consequences, such as requiring a deposit in the Copyright Office.[92] Finally, Congress used the two terms differently by defining publication as either distribution of copies or phonorecords to the public or offering to distribute for purposes of further distribution.[93] The court concluded that distribution is a subset of publication.[94]

In *National Car Rental System, Inc.* v. *Computer Associates International, Inc.*, the court concluded that a license agreement was not preempted by copyright law because it did not implicate any of the fundamental rights—including distribution—under the Copyright Act.[95] The license agreement was different than the right of distribution under the Copyright Act because distribution requires "an actual dissemination of either copies or phonorecords."[96] Finally, in *London-Sire Records, Inc.* v. *Doe*, the court ruled that dissemination and distribution are not identical:

> [S]uppose an author has a copy of her (as yet unpublished) novel. If she sells that copy to a member of the public, it constitutes both distribution and publication. If she merely offers to sell it to the same member of the public, that is neither a distribution nor a publication. And if the author offers to sell the manuscript to a publishing house 'for purposes of further distribution,' but does not actually do so, that is a publication but not a distribution.[97]

A few conclusions can be drawn from this complex case law. At the very least, transferring ownership of a copy by sale, rental, lease or lending constitutes distribution. Making a work available to the public, such as placing a copy of a copyrighted work on a server, available for download, constitutes distribution if multiple individuals actually download it. If only a single individual downloads the work, under *Ford Motor Co. v. Summit Motor Products, Inc.*, this act would constitute distribution to the public,[98] although

[87] Capitol Records, Inc. v. Thomas, 579 F. Supp. 2d 1210, 1227 (D. Minn. 2008).

[88] *Id.* at 1212–13.

[89] *Id.* at 1227.

[90] *Id.* at 1216.

[91] *Id.* at 1217.

[92] *Id.* at 1220.

[93] *Id.*

[94] *Id.*

[95] Nat'l Car Rental Sys., Inc. v. Computer Assocs. Int'l, Inc., 991 F.2d 426, 430–35 (8th Cir. 1993).

[96] *Id.* at 434 (internal quotation marks omitted).

[97] London-Sire Records, Inc. v. Doe, 542 F. Supp. 2d 153, 169 (D. Mass. 2008).

[98] *See* Ford Motor Co. v. Summit Motor Prods., Inc., 930 F.2d 277, 299 (3d Cir. 1991) (holding that

academics may argue that dissemination of a single copy is not sufficient to constitute "to the public." It is debatable whether making a copy available to the public, with no evidence of anyone downloading or otherwise obtaining a copy of the work, constitutes distribution.

2. *Limited Publication*

Another wrinkle in distribution case law involves the limited publication doctrine. The right of distribution entails distribution to the public; thus, "[a] limited publication (i.e., a distribution made to a limited group for a limited purpose and not made to the public at large) should not infringe this right."[99]

Courts developed the limited publication doctrine to determine when an author lost his common-law copyright to works by publishing them. Prior to the Copyright Act of 1976, copyright vested with authors of original works at creation under common law copyright.[100] If an author published the work, he lost his common-law copyright.[101] However, if he published a work *with proper notice*, he triggered protection under federal copyright law.[102] If he published a work without proper notice, he lost both common law and federal copyright protection, and the work was placed into the public domain.[103] Courts differentiated between limited publication and general publication in order to mitigate the harshness of this rule.[104] A work that had only limited publication was not subject to forfeiture if the author did not affix a notice to the work or the notice did not comply with copyright standards.[105]

The current Copyright Act does not mention limited publication; however, courts and commentators continue to assert its relevance. According to Professor Nimmer:

> The statutory definition of "publication" does, however, require a "distribution of copies or phonorecords of a work *to the public* . . . ." Moreover, the House Report in explaining what is meant by "the public" in this context states that it refers to persons under no explicit or implicit restrictions with respect to disclosure of its [i.e., the work's] contents." This appears to suggest a continuation of the

distribution to a single person can constitute distribution to the public).

[99] NIMMER & NIMMER, *supra* note 59, § 8.11[C][1][b] n. 139. *See also* Intown Enterprises, Inc. v. Barnes, 721 F. Supp. 1263, 1265 (N.D. Ga. 1989) ("This definition [of publication] does not encompass 'limited' publications.").

[100] The Copyright Act of 1976 abolished common law copyright and vested federal copyright of original works with authors upon fixation in a tangible medium, without a requirement for notice. *See* 17 U.S.C. § 102(a); Jalbert v. Grautski, 554 F. Supp. 2d 57, 69 n.15 (D. Mass. 2008) ("[T]he Copyright Act of 1976 abolished common law copyright as a general matter . . . .").

[101] Thomas F. Cotter, *Toward a Functional Definition of Publication in Copyright Law*, 92 MINN. L. REV. 1724, 1728-29 (2008).

[102] *Id*. at 1729.

[103] *Id.* Congress has continuously relaxed notice requirements since then.

[104] American Vitagraph, Inc. v. Levy, 659 F.2d 1023, 1026–27 (9th Cir. 1981) ("[C]ase law has created a distinction between general and limited publication, holding that only the former operates to divest common law copyright and subject a work to the federal statutory scheme.").

[105] Cotter, *supra* note 101, at 1730.

> doctrine of limited publication under the current Act.[106]

Similarly, recent courts have applied the limited publication doctrine to not only determine if a copyright owner has been divested of his or her common law copyright, but also to determine if a work has infringed the exclusive rights of publication and distribution.[107]

The Ninth Circuit set forth the seminal test for limited publication in *White* v. *Kimmel*.[108] Stewart Edward White wrote and personally published a manuscript, printing over a hundred copies of it.[109] He mailed copies to friends with a letter that stated:

> I wish you to read it, to use it as you like, and pass it on to others, and for as long a time as you can. If you get through with it, you might return it to me to hand to someone else. Otherwise, you are at liberty to keep it.[110]

Pursuant to the letter, White's secretary handed out other copies to her friends and acquaintances.[111] White further permitted four different fans of the manuscript to print copies of the manuscript, distribute those copies, and charge for the cost of reproduction, although some fans were cautioned to use extreme care in selecting recipients.[112]

The court found that this amounted to a general publication, rather than a limited one.[113] The court determined:

> [A] limited publication which communicates the contents of a manuscript to a definitely selected group and for a limited purpose, and without the right of diffusion, reproduction, distribution or sale, is considered a 'limited publication,' which does not result in loss of the author's common-law right to his manuscript; but . . . the circulation must be restricted both as to persons and purpose, or it can not be called a private or limited publication.[114]

White's manuscript was not a limited publication because he did not give the manuscript to a limited group of individuals.[115] Rather, he passed on copies to anyone who requested one and allowed others to distribute the work.[116] Furthermore, White's letter evidenced that recipients of the work were free to pass it on to others.[117]

[106] NIMMER & NIMMER, *supra* note 59, § 4.13[B] (alteration in original).

[107] *See* Jalbert v. Grautski, 554 F. Supp. 2d 57, 68–69 (D. Mass. 2008) (applying the limited publication doctrine in discussing whether defendant violated the exclusive right to distribute to the public). *See also* RPM Mgmt., Inc. v. Apple, Inc., 943 F. Supp. 837, 842-43 (S.D. Ohio 1996) (discussing the doctrine of limited publication in determining whether plaintiff is entitled to statutory damages).

[108] *See* White v. Kimmell, 193 F.2d 744, 746–47 (9th Cir. 1952).

[109] *Id*. at 745.

[110] *Id*.

[111] *Id*.

[112] *Id*. at 746.

[113] *Id*. at 747.

[114] *Id*. at 746–747.

[115] *Id*. at 747 ("[W]e are unable to see in this picture any definitely selected individuals or any limited, ascertained group or class to whom the communication was restricted.").

[116] *Id*. at 745–46.

[117] *Id*. at 747.

In sum, in order to constitute a limited publication, the work must be distributed to (1) a select group, (2) for a limited purpose, and (3) with restrictions on the recipients' ability to distribute, reproduce, sell and diffuse the copyrighted works, none of which White satisfied. Cases following *White* have found that delivering architectural plans for bidding, disseminating samples of a work for promotion, sharing copies of a work with trade members for criticism or review, and providing copies of manuscript to an author's friends are limited rather than general publications.[118]

The limited publication doctrine has never been used by courts to analyze distribution under the GPL. At least one commentator has recognized the applicability of the limited publication doctrine to the GPL and has used limited publication in discussing when the use of open source software in the cloud constitutes a distribution under the GPL.[119] The doctrine is not only limited to cloud computing; in fact, many of the scenarios in which copyleft may be implicated involve a distribution to a limited group for a limited purpose. For example, limited publication may be useful in determining whether outsourcing the development or modification of proprietary software constitutes a distribution or whether transferring software to subsidiaries constitutes a distribution.

3. Software Over a Network

Copyright law also suggests that providing services over a network, without transfer of a copy or making a copy available, does not constitute distribution. This is relevant for application service providers ("ASPs"), also known as software as a service providers ("SaaS" providers), that provide software and other applications to customers over a network. Computing giant Google is an example of an ASP that provides software like Gmail and Google Docs to users without actually transferring a copy of the software. Under copyright law, providing software as a service to a user over a network, without transferring a copy of that software, probably does not constitute a distribution, because copyright law requires the actual transfer of a copy for distribution.[120] According to Nimmer, "Infringement of this right requires an actual dissemination of either copies or phonorecords."[121] In the debate over whether making a work available to the public constitutes distribution,[122] a number of courts have held that an actual transfer of a file is needed to violate the public distribution right. The most relevant case was *National Car System v. Computer Associates*, where Plaintiff Computer Associates licensed

[118] *See* NIMMER & NIMMER, *supra* note 59, § 4.13[A][3] (reviewing cases). *See also* Unix Sys. Labs. v. Berkeley Software Design, No. 92-1667, 1993 U.S. Dist. LEXIS 19505, at *13 (D.N.J. Mar. 3, 1993) (reviewing cases).

[119] *See* Mark Radcliffe, *Open Source in the Cloud: New Legal Questions*, LAW & LIFE: SILICON VALLEY (July 30, 2010), http://lawandlifesiliconvalley.com/blog/?m=201007 ("Ironically, cloud computing requires us to reach back to the 'limited publication' doctrine from the 1909 Act to find legal guidance for how to interpret the scope of the distribution right.").

[120] *See, e.g.*, Capitol Records, Inc. v. Thomas, 579 F. Supp. 2d 1210, 1227 (D. Minn. 2008).

[121] NIMMER & NIMMER, *supra* note 59, § 8.11[C][1][a].

[122] *See infra* Part 1 for an in-depth discussion of this debate.

computer software to National Car Services for the limited purpose of processing the defendant's own data.[123] National Car Services subsequently used the software to process data of third parties, violating the terms of the license agreement.[124] The court held that the contractual breach was not preempted by federal copyright law because the restriction on use was not equivalent to any of the exclusive rights in copyright, including distribution.[125] The court held that "even with respect to computer software, the distribution right is only the right to distribute *copies* of the work."[126]

On the other side of the making available debate, courts have held that making a copy available to the public, even without the actual transfer of a copy, constitutes distribution.[127] However, these cases can be distinguished from ASPs in that ASPs never make a copy of the software available. Users can only interact with the ASP over a network in order to *use* the software—obtaining a copy is never an option. The emerging case law involving streaming copyrighted works over the Internet further bolsters this argument. Plaintiffs who have sued defendants for streaming copyrighted works have argued violations of the public display and reproduction rights, rather than the distribution right.[128] This is presumably because distribution classically entails transfer of a copy, which is not present in streaming.

One commentator has distinguished between cases in which ASPs do not transfer any copy of software and cases in which ASPs transfer some portions of software, such as user interfaces or Java-applets, that are then copied and stored in the RAM of a user's computer.[129] Courts have held that copying occurs when the information necessary to make a copy onto a user's RAM is transferred, even if no actual material copy is transferred.[130] Thus, the argument proposes that a copyrighted portion of software copied onto a user's RAM is sufficient to find distribution, even for ASPs.[131] Such a situation would be rare. The generally accepted rule under copyright law is that providing software over a network, without any transfer of a copy, does not constitute distribution under the GPL.

[123] Nat'l Car Rental Sys., Inc. v. Computer Assoc. Int'l, Inc., 991 F.2d 426, 427 (8th Cir. 1993).

[124] *Id*. at 428.

[125] *Id*. at 433.

[126] *Id*. at 434.

[127] *See infra* Part 1 for a discussion of these cases.

[128] Cartoon Network LP, LLLP v. CSC Holdings, Inc., 536 F.3d 121, 124 (2d Cir. 2008) ("[Plaintiffs] alleged that Cablevision's proposed operation of the RS-DVR would directly infringe their exclusive rights to both reproduce and publicly perform their copyrighted works.). *See also* Joshua Keesan, Note, *Let It Be? The Challenges of Using Old Definitions for Online Music Practices*, 23 BERKELEY TECH. L.J. 353, 361 ("Thus, as early as 2000, HFA [the Harry Fox Agency] has argued that every streaming transmission is a reproduction as well as a public performance due to RAM storage.").

[129] Michael P. Widmer, *Application Service Providing, Copyright, and Licensing*, 25 J. MARSHALL J. COMPUTER & INFO. L. 79, 98 (2007).

[130] *Id*. at 97 (citing Playboy Enterprises Inc. v. Webbworld, Inc., 991 F. Supp. 543, 551 (N.D. Tex. 1997); Central Point Software v. Nugent, 103 F. Supp. 1057 (E.D. Tex. 1995); Playboy Enterprises v. Frena, 839 F. Supp. 1552 (M.D. Fla. 1993)).

[131] *Id*. at 97–98.

C. What is a Distribution for Purposes of the GPL?

Although GPLv1 and GPLv2 fail to define "distribute," the Frequently Asked Questions (FAQ) offer some insight into what the FSF believes the term to mean.[132] Exactly how much weight to give the FAQ is uncertain. If the courts adopt a contract theory rather than a license theory, courts would probably give the FAQ some weight. This is particularly true if the FSF owns a copyright in the work and brings suit. On the other hand, a court may not adopt a contract interpretation given that the FSF has continuously stated that the GPL is a license that should not extend beyond the scope of copyright law.[133] However, if the FSF is not a party, it is unclear how much weight to give the FAQ—particularly if the plaintiff copyright owner's opinion diverges from that of the FSF. Such a scenario is likely given the sheer number of copyright owners in GPL-covered works, with hundreds of people owning a copyright in Linux alone.[134] Furthermore, if an individual licensed a work under the GPL under one understanding of the license, and the FSF updates the FAQ with another understanding of the license, the FAQ probably should not be granted very much weight. Finally, if the court adopts a license theory, and the FAQ diverge from what has been established in copyright case law, the courts will follow case law. This is particularly true given that the FSF believes the GPL does not extend beyond the scope of copyright law.[135] Thus, it is important to examine the FAQ with a grain of salt in analyzing terms of the license.

This section synthesizes the FAQ with copyright law and commentators' understandings in order to determine what constitutes a distribution.

1. Acts That are Not Distributions

Using GPL-covered software for one's own use does not constitute a distribution. Furthermore, using software within a single company does not constitute distribution.[136] This position is consistent with the FAQ,

[132] GPLv2 has accompanying Frequently Asked Questions; GPLv1 does not. *Frequently Asked Questions about Version 2 of the GNU GPL*, GNU OPERATING SYSTEMS, http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#InternalDistribution (last visited Feb. 20, 2012).

[133] VAN LINDBERG, INTELLECTUAL PROPERTY AND OPEN SOURCE: A PRACTICAL GUIDE TO PROTECTING CODE 237 (Andy Oram ed., 2008) ("The FSF has repeatedly stated, however, that they believe in copyright minimalism and that the GPL should not be interpreted to extend beyond the reach of copyright.").

[134] *Copyright Considerations with LKMs*, LINUX DOCUMENTATION PROJECT, http://tldp.org/HOWTO/Module-HOWTO/copyright.html (last visited Feb. 20, 2012) ("Who owns the copyright on Linux? Lots of people. Nearly everyone who has contributed code to Linux reserved his own copyright."); Jay Michaelson & Christopher Holst, *Closed-Source Loadable Kernel Modules Violate the GPL*, WASABI SYSTEMS, 6 (Feb. 21, 2006), *available at* http://www.wasabisystems.com/pdf/LoadableKernelModules_Wasabi.pdf ("There are hundreds of individuals and organizations who have a copyright interest in the body of code contained in the Linux kernel.").

[135] LINDBERG, *supra* note 133, at 237.

[136] *Frequently Asked Questions about the GNU Licenses*, *supra* note 29 (Q: "If I give a copy of a GPLv3-covered program to a coworker at my company, have I "conveyed" the copy to him?" A: "As long as you're both using the software in your work at the company, rather than personally, then the answer is no. The copies belong to the company, not to you or the coworker. This copying is propagation, not conveying, because the company is not making copies available to others."); *Frequently Asked Questions about Version 2*

commentators' understandings,[137] and case law.

The GPL contains an ASP loophole, or network use provision, providing that vendors who provide software as a service to users over a network, without transfer of a copy, do not engage in distribution.[138] The ASP loophole has come under fire for allowing some of the most aggressive users of FOSS, who also double as computing giants that provide software services to billions of users, to resist contributing back to the FOSS community.[139] Amazon and Google, for example, built their programs on FOSS, but yet are not required to abide by the very free terms they profited from.[140] To remedy this problem, the FSF created the Affero General License, which closes the loophole by clarifying that hosted software is distributed and triggers the GPL's copyleft.[141] However, while the Affero license solves the ASP loophole in theory, it does not do so in practice because major programs like Linux and MySQL are not licensed under the Affero license.[142]

The ASP loophole is present in all versions of the GPL. Although neither GPLv1 nor GPLv2 affirmatively state the existence of the ASP loophole in the license itself, its presence has been inferred from the fact that no copy of the software is transferred or made available to the public when a company provides software as a service.[143] Furthermore, GPLv2's FAQ refers to the ASP loophole—the FAQ acknowledge that a company running a modified version of a GPL'ed program on a web site does not need to release the code.[144] In GPLv3, the Free Software Foundation explicitly included the loophole in the license itself, not just in the FAQ.[145] The license states, "Mere interaction with a user through a computer network, with no transfer of a copy,

of the GNU GPL, *supra* note 132 (Q: "Is making and using multiple copies within one organization or company 'distribution'?" A: "No, in that case the organization is just making the copies for itself. As a consequence, a company or other organization can develop a modified version and install that version through its own facilities, without giving the staff permission to release that modified version to outsiders.").

[137] *See* Martin von Haller Groenbaek, Distribution in Open Source, Presentation at ITech Law European Conference, Brussels, Belg. (Nov. 5, 2009) (slideshow available at http://www.slideshare.net/vonhaller/distribution-in-open-source) (discussing the meaning of propagating and convey works); Philip H. Albert, *GPL Rules: Lessons from the Sandbox*, LINUXINSIDER (Dec. 24, 2004), http://www.linuxinsider.com/story/38728.html?wlc=1277203695.

[138] *AGPL: Open Source Licensing in a Networked Age*, REDMONK, http://redmonk.com/sogrady/2009/04/15/open-source-licensing-in-a-networked-age/ (last visited Feb. 7, 2012).

[139] *Id.*

[140] *See* Albert, *supra* note 137 (stating that MySQL was rethinking its commitment to the GPL). *See also* Mark Radcliffe, *Licensing Shift: Move Toward GPLv3 but away from GPL Based Licenses*, LAW & LIFE: SILICON VALLEY (Feb. 7, 2009), http://lawandlifesiliconvalley.com/blog/?p=256 (discussing how Linux decided to remain with GPLv2 instead of changing to GPLv3).

[141] GNU AFFERO GENERAL PUBLIC LICENSE, (Nov. 19, 2007), *available at* http://www.gnu.org/licenses/agpl-3.0.html.

[142] *Id.*

[143] In fact, in GPLv2's FAQ, the Free Software Foundation disapproved of this loophole as contrary to the spirit of the GPL and signaled its intent to potentially close the loophole in version 3. *See Frequently Asked Questions about Version 2 of the GNU GPL*, *supra* note 132 ("[P]utting the program on a server machine for the public to talk to is hardly 'private' use, so it would be legitimate to require release of the source code in that special case. We are thinking about doing something like this in GPL version 3, but we don't have precise wording in mind yet.").

[144] *Id.*

[145] GPLv3, *supra* note 26, § 0 (defining "convey").

is not conveying."[146]

The ASP loophole is generally consistent with case law in holding that software provided over a network does not constitute distribution.[147] However, it is broader than necessary. As discussed previously, courts would likely hold that the use of software over a network which results in copyrighted portions of software being stored onto a user's RAM *does* infringe the distribution right.[148] The GPL, however, does not parse so finely. GPLv3 explicitly and unconditionally grants users permission to freely provide software as a service, with no trigger of copyleft.[149] As a result, ASPs can freely modify GPL-covered software, provide the modified software as a service, and refuse to share the source code or license the new work under the GPL without fearing a lawsuit. Even though copyright holders would have a cause of action under traditional copyright law in the rare case that network use results in portions of software being stored in a user's RAM, the unconditional grant of permission prevents copyright holders from suing for either breach of contract or copyright infringement. There may be different consequences for GPLv2, however, as the ASP loophole is only inferred and not written in the license.[150] If a copyright holder wanted to bring suit against an ASP who offers software that is copied onto a user's RAM, he would be able to do so. Courts would likely find that this constitutes a distribution, and the GPL does not provide otherwise. However, because the ASP loophole has been generally accepted by the FOSS community, the chances of a copyright holder bringing suit in such a scenario are slight. Generally, ASPs can rest assured that providing software over a network, without transferring a copy, does not constitute distribution.

2. *Acts That are Distributions*

Transferring a GPL-covered work to other organizations or individuals is distribution. In particular, giving someone a copy of GPL-covered software, either via a download or via mail-order for physical mediums such as CDs, constitutes distribution. This interpretation is consistent with the GPLv2's and the GPLv3's Frequently Asked Questions,[151] commentators' understandings,[152] and case law.[153]

[146] *Id.*

[147] *See infra* Part 3 for a discussion of the relevant case law.

[148] *See supra* Part IV.B.3 (discussing the implications of Nat'l Car Rental Sys., Inc. v. Computer Assoc. Int'l, Inc., 991 F.2d 426, 427–33 (8th Cir. 1993)).

[149] GPLv3, *supra* note 26, § 13.

[150] *Frequently Asked Questions about version 2 of the GPL, supra* note 132.

[151] *See Frequently Asked Questions about the GNU Licenses*, *supra* note 29; *Frequently Asked Questions about version 2 of the GPL*, *supra* note 132.

[152] *See, e.g.*, Albert, *supra* note 137 (noting that distribution to end users outside an organization is distribution); Groenbaek, *supra* note 137 (noting that transferring a copy from one person to another is distribution under GPLv2).

[153] *See, e.g.*, London-Sire Records v. Doe, 542 F. Supp. 2d 153, 170 (D. Mass. 2008) (holding that transferring electronic copies violates the distribution right); NIMMER & NIMMER, *supra* note 59, § 8.11[C][1][a] (noting that distributing copies or phonorecords violates the distribution right). This assumes that the limited publication doctrine does not come into play.

3. Open Questions for Distribution under the GPL

Whether providing software to a cloud computing vendor, outsourcing services, and making a work available to the public constitute distributions under the GPL remain open questions.

a. Does Cloud Computing Trigger Copyleft?

Cloud computing is the consumption of software, services, and hardware over a network.[154] Software as a service is only one component of cloud computing—cloud computing also encompasses infrastructure as a service or IaaS (providing hardware, operating system, servers, disk storage, etc. over the internet) and platform as a service or PaaS (providing infrastructure as a service to developers to build or extend applications).[155] Cloud computing offers a variety of benefits for companies: efficiency, low cost, low barriers to entry, customization, and immediate access to a variety of applications.[156] Users can avoid the capital influx needed to build their own computing systems, and companies can pay on a per-use basis, allowing them to forgo the costs of idle servers and employees during slow periods.[157] Furthermore, companies can utilize the services of skilled third-party experts for resource-intensive tasks, freeing up labor and money to take on other essential tasks.[158] Cloud computing has become a huge phenomenon, with companies rushing to the cloud to take advantage of its dramatic savings.[159]

Cloud computing vendors offering software as a service to end users do not activate copyleft due to the ASP loophole, as previously discussed.[160] However, other situations arise which may activate copyleft. For example, suppose a company decides to host its software on a cloud computing vendor's servers. Does using cloud computing to host proprietary software (regardless of whether the ultimate purpose is to make the software accessible to end users, use it within the company, etc.) constitute a distribution to the vendor, thus implicating copyleft?

Under the limited publication doctrine, probably not. The transfer is made to a limited group—the cloud computer vendor's employees—for the limited purpose of providing facilities in which to run the works. Presumably, the cloud computer vendor is restricted from freely distributing, reproducing,

[154] STEVE BENNETT, MANS BHULLER, & ROBERT COVINGTON, ORACLE WHITE PAPER IN ENTERPRISE ARCHITECTURE – ARCHITECTURAL STRATEGIES FOR CLOUD COMPUTING 4 (Aug. 2009), *available at* http://www.techrepublic.com/ whitepapers/architectural-strategies-for-cloud-computing/1188405.

[155] Mike Gray, *Cloud Computing: Demystifying IaaS, PaaS and SaaS*, ZDNET (Oct. 21, 2010, 8:28 AM), http://www.zdnet.com/news/cloud-computing-demystifying-iaas-paas-and-saas/477238.

[156] Cindy Waxer, *The Benefits of Cloud Computing*, WEB HOSTING UNLEASHED, http://www.webhostingunleashed.com/features/cloud-computing-benefits/ (last visited Feb. 7, 2012).

[157] *Id*.

[158] *Id*.

[159] Dawn M. Foster, *Is the Current Economy Good for Open Source Software?*, INTEL SOFTWARE BLOGS (July 16, 2009, 9:08 AM), http://software.intel.com/en-us/blogs/2009/07/16/is-the-current-economy-good-for-open-source-software/ ("[W]ith the current economy, cost is becoming a primary driver leading more companies to explore open source solutions.").

[160] *See supra* Part 1 for a discussion of ASPs.

and selling the proprietary works; thus, a court would likely find that this limited publication does not amount to a distribution.[161] If a company fails to place restrictions on the cloud computing vendor's ability to access, use, and distribute the proprietary software (admittedly, an unlikely scenario), courts are more likely to find a distribution.

The FAQ loosely trace the limited publication doctrine with a safe harbor, which we will refer to as the "cloud computing/outsourcing safe harbor." The safe harbor reads:

> You may convey covered works to others for the sole purpose of . . . provid[ing] you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.[162]

The safe harbor thus enunciates three criteria set forth by the *White* court—a select group, the limited purpose of running those works in third-party facilities and restrictions on the third party's ability to copy the copyrighted material.[163] The safe harbor diverges from the *White* test in only prohibiting recipients from making copies of copyrighted material, and not restricting recipients' rights to "diffusion, reproduction, distribution, or sale" of copyrighted material, as outlined by *White*.[164] Thus, a best practice would probably be to include all the aforementioned restrictions in an agreement with a cloud computing vendor. Under those circumstances, providing proprietary software to a cloud computing vendor should not constitute distribution. The few commentators who have addressed the question have reached the same conclusion, though some have adopted different reasoning.[165]

b. Does Outsourcing Trigger Copyleft?

It is unclear whether outsourcing triggers copyleft, although the answer is likely no. The question arises in the following context: a company wants to modify GPL-covered work for its own use or wants to use a contractor to develop, update, or modify software derived from GPL-covered software. The company wants to keep the software proprietary and will not distribute it (it

[161] See *supra* Part IV.B.2. for a discussion of judicial application of the limited publication doctrine to determine if copyright has been infringed.

[162] GPLv3, *supra* note 26, § 2.

[163] *See* White v. Kimmell, 193 F.2d 744, 746–47 (9th Cir. 1952).

[164] See *id*.

[165] *See* Posting of Joseph H to *Does Placing GPL Licensed Software on a Server Qualify as "Distribution" if the End User Never Sees It?*, STACK OVERFLOW (Sept. 4, 2010 12:51), http://stackoverflow.com/questions/3640415/does-placing-gpl-licensed-software-on-server-qualify-as-distribution-if-end-use (citing the GPL's safe harbor in concluding that hosting software on a third party's servers does not constitute distribution); Posting of Bramp to *id.* (Sept. 4, 2010, 0:56) (concluding that distribution does not occur if a company does not distribute binaries). *Cf.* Radcliffe, *supra* note 119 (applying the limited publication doctrine to cloud computing but not reaching a conclusion).

will either provide the software as a service to users or only use it within the company). The question is whether the initial transfer of the software from the client to the contractor and any ensuing transmissions of software during the design process (if, for example, the client and contractor make changes and send the modified software back and forth) constitute a distribution.[166] If yes, the client has inadvertently granted the contractor the exclusive right to modify, distribute, and reproduce the software. Any contractual attempts to limit such rights would be contrary to the GPL and could not be enforced.[167]

i. Outsourcing Under the Language of the GPL

The various versions of the GPL and their accompany Frequently Asked Questions are unclear about whether outsourcing—providing software to contractors for development, modification, or updating—triggers copyleft. GPLv2 does not address outsourcing in the license itself.[168] GPLv3 includes the "cloud computing/outsourcing safe harbor," suggesting that outsourcing does not implicate the GPLv3's copyleft.[169] Since the outside company simply provides "facilities for running" the software or "make[s] modifications" to the proprietary software, provided that the conditions regarding direction, control, and contract terms are met, such an arrangement should not trigger copyleft.[170]

However, there remains some confusion stemming from language in GPLv3's FAQ. GPLv3's FAQ states, "[W]hen the organization transfers copies to other organizations or individuals, that is distribution. In particular, providing copies to contractors for use off-site is distribution."[171] This language seems to potentially contradict the outsourcing safe harbor in some scenarios. Consider the following situation: a company provides a contractor with proprietary software based on GPLv3-covered software. The contractor makes modifications to the program off-site, under all of the conditions listed in the outsourcing safe harbor—exclusively on the company's behalf, under the company's direction and control, on terms that prohibit them from making any copies of copyrighted material outside their relationship with the company. Would that constitute a distribution? According to the outsourcing safe harbor, it would not, but according to the FAQ, it would. Courts interpreting this scenario would likely give more weight to the language of the actual license rather than the FAQ and say no distribution occurred.

[166] Note that the transfer of software from the contractor to the client does not present a problem. Though this may constitute a distribution, the result is that the client has the right to modify and redistribute the work if it wants. If the client chooses not to redistribute the software, it can keep the software proprietary.

[167] *See* GPLv2, *supra* note 65, § 6 ("Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein."). *See also* GLPv3, *supra* note 26, § 10 ("You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License.").

[168] *See generally* GPLv2, *supra* note 65.

[169] *See* GLPv3, *supra* note 26, § 2.

[170] *See id.* §§ 2, 5 (discussing modifications and outsourcing).

[171] Note that this language would not be pertinent to cloud computing, as cloud computing vendors do not "use" software, but merely host it. *See Frequently Asked Questions about the GNU Licenses*, *supra* note 29.

One possible distinction between the conflicting language may be that the outsourcing safe harbor refers to individuals "running or modifying" the software, while the GPLv3 FAQ refers to a contractor "using" the software.[172] Perhaps a contractor who "uses" the software may be doing so for his own purposes and not the company's purposes, although the language of the FAQ does not distinguish between personal use and company use.[173] Another distinction lies in whether a contractor is off-site or on-site.[174] It is unclear why the geographical location of the contractor would determine whether a transfer is a distribution. One could argue that an off-site contractor is less like an employee of the company and more like a separate entity, and thus the company is not using the software internally. However, one could also easily imagine a scenario where a contractor is very much like an employee but works off-site, or where an independent contractor works at the company but otherwise maintains his independence. Off-site versus on-site should not be determinative to whether a transfer is a distribution.

Suppose that the same scenario featured above again presents itself but this time, the software in question is licensed under GPLv2. The analysis becomes a bit more complicated because the outsourcing safe harbor is absent in GPLv2. In fact, the outsourcing safe harbor was added to GPLv3 as a result of companies and IT contractors expressing confusion and concern over whether outsourcing triggers copyleft.[175] Further confusion arises from the fact that GPLv2's FAQ, like GPLv3's FAQ, state that "providing copies to contractors for use off-site is distribution."[176]

The FAQ does contain some relevant language indicating that outsourcing is permitted. GPLv2's FAQ states:

> Does the GPL allow me to develop a modified version under a nondisclosure agreement?
>
> Yes. For instance, you can accept a contract to develop changes and agree not to release your changes until the client says ok. This is permitted because in this case no GPL-covered code is being distributed under an NDA.
>
> You can also release your changes to the client under the GPL, but agree not to release them to anyone else unless the client says ok. In this case, too, no GPL-covered code is being distributed under an NDA, or under any additional restrictions.
>
> The GPL would give the client the right to redistribute your version.

[172] *See id.* ("[W]hen the organization transfers copies to other organizations or individuals, that is distribution. In particular, providing copies to contractors for use off-site is distribution.").

[173] *Id.*

[174] *See id.* (stating that "providing copies to contractors for use off-site is distribution").

[175] Ciaran Farrell, *Outsourcing Development with the GPLv2 and the GPLv3*, FOSS BAZAAR, https://fossbazaar.org/content/outsourcing-development-gplv2-and-gplv3 (last visited Feb. 20, 2012).

[176] *See Frequently Asked Questions about the GNU Licenses*, *supra* note 29.

> In this scenario, the client will probably choose not to exercise that right, but does have the right.[177]

At a minimum, GPLv2 allows the above scenario. This language clarifies that a contractor who transfers work based on the program to a client engages in distribution. However, GPLv2 does not directly address whether the initial transfer of GPL-covered software or work based on the program from the client to the contractor should constitute a distribution as well, thus triggering copyleft and giving the contractor all of the rights granted by the GPL.

The language cited above seems to contemplate a scenario in which a contractor independently obtains a GPL-covered work—not a proprietary modified version. Since he did not obtain the work from the client, the client has not engaged in a distribution and has not triggered copyleft. Whoever gave the contractor the GPL-covered work has triggered copyleft and thus cannot claim the work as proprietary; however, this has no bearing on the contractor and the client. The contractor could then agree to develop changes for the client under a non-disclosure agreement. The non-disclosure agreement does not violate the GPL in such a scenario because it was not a part of the initial distribution—the original distribution of the GPL-covered work from the distributor to the contractor complied with the terms of the GPL. The contractor could then release the modified work to the client, thus distributing it and granting the client the right to redistribute and modify the work. However, as the GPL notes, the client will probably not want to do so.[178]

If the contractor obtains a GPL-covered work from the client, it is unclear whether he could then agree to modify it under a disclosure agreement. The answer would hinge on whether the non-disclosure agreement accompanied the initial distribution, as the GPL prohibits non-disclosure agreements accompanying distributions.[179] In such a scenario, the non-disclosure agreement would probably not accompany the distribution, as it accompanies an agreement for modification and is not a condition of accepting the initial distribution.

The language in GPLv2's FAQ does not contemplate a scenario in which the client uses a contractor to further modify a derivative work based on GPL-covered software. That is, suppose the client has developed proprietary software based on GPL-covered software. If the client uses a contractor to further modify or update that proprietary software, the initial transfer of the proprietary software to the contractor would likely constitute a distribution, since GPLv2 does not provide otherwise. This would trigger copyleft, as distribution and a derivative work are both present.[180] The result is less than ideal for the client—the client cannot accompany the distribution with a non-disclosure agreement; the contractor can freely reproduce, distribute, and

[177] *Id.*

[178] *Id.*

[179] *See id.* ("Does the GPL allow me to distribute copies under a nondisclosure agreement? No. The GPL says that anyone who receives a copy from you has the right to redistribute copies, modified or not. You are not allowed to distribute the work on any more restrictive basis.").

[180] *See supra* Part II.A (discussing the triggers for copyleft).

modify the software; and the proprietary software may lose much of its value. This is the exact scenario contemplated and addressed by the outsourcing safe harbor in GPLv3; however, the absence of it in GPLv2 creates many problems for clients wishing to utilize outsourcing.[181] The easiest solution would be to amend the GPL itself to clarify that companies using contractors for development purposes do not trigger the GPL's copyleft by transferring the software to those contractors. This should be clarified in the language of the license itself rather than the FAQ.

ii. Outsourcing Under Applicable Copyright Law

Surprisingly, outsourcing under applicable copyright law is much simpler analysis. Under the limited publication doctrine, outsourcing would not constitute a distribution. This analysis is very similar to that of cloud computing. Outsourcing involves a transfer of software to a limited group (the contractors), for the limited use of modifying or updating the software, and presumably with restrictions on the contractors' abilities to copy, sell, and distribute the software.[182] Again, if any of these components are not met—for example, if the contractor is not restricted from selling or copying the software—then courts may not find a limited publication.[183] Thus, outsourcing under copyright law would probably be a fact-specific inquiry.

iii. Conclusion for Outsourcing

The GPL and its FAQ are unclear about whether outsourcing is a distribution. In general, copyright holders bringing suit under GPLv2 have a much stronger argument that outsourcing constitutes distribution than those bringing suit under GPLv3; thus, companies should attempt to base proprietary software on GPLv3-covered works if possible. The GPLv3 gives users unconditional permission to outsource without triggering copyleft. Because GPLv2 does not have this language and relies on the definition of distribution adopted by copyright law, it is unclear whether outsourcing triggers copyright law under GPLv2. Presumably, asking a contractor to make modifications on-site rather than off-site would allow one to skirt the GPL's copyleft, although there does not seem to be much of a reason for that distinction. Copyright law offers a more elegant analysis, dictating that outsourcing does not involve a distribution provided that the transfer of software to the contractor follows the principles delineated in *White* for a limited publication. This analysis, however, is contingent upon courts applying the limited publication doctrine to the GPL, which they may or may not do. In general, however, the FOSS community appears to accept that outsourcing should not trigger copyleft, as evidenced by the FSF adding a safe harbor to GPLv3 in order to allay companies' concerns. Thus, companies may be able to rely on individuals

[181] *See* Farrell, *supra* note 175.

[182] *See* Farrell, *supra* note 175 (discussing outsourcing).

[183] *See supra* Part IV.B.2 (discussing the limited publication doctrine).

refraining from bringing suit, even if they cannot rely on clearly articulated law.

c. Does Making a Work Available to the Public—Without the Transfer of a Copy—Trigger Copyleft?

If a company makes a proprietary work available (for example, available for download over a web server) and no user actually obtains a copy, it is unclear whether making the work available would constitute a distribution. Courts are divided over whether making a work available constitutes distribution.[184]

GPLv2 does not address the making available debate. GPLv3, in contrast, includes "making available to the public" in its definition of "propagate."[185] GPLv3's FAQ clarifies that "convey" includes making available and suggests that making available constitutes distribution:

> One example of 'making available to the public' is putting the software on a public web or FTP server. After you do this, some time may pass before anybody actually obtains the software from you—but because it could happen right away, you need to fulfill the GPL's obligations right away as well. Hence, we defined conveying to include this activity.[186]

To the extent this interpretation is contrary to the controlling precedent in a certain jurisdiction, courts probably will not find distribution on the basis of making a proprietary work available. However, because the law is inconsistent in this area, companies wanting to keep software proprietary may want to err on the side of caution and offer software as a service rather than make it available for download. In general, because the case law is confused in this area,[187] whether making available constitutes distribution remains an open question and may vary by jurisdiction.

V. Derivative Works

A. Derivative Works Under the GPL

A derivative work is the second trigger for the GPL's copyleft. The GPL applies to covered works (works released under the GPL), and derivative works, which are, in the language of the GPL, works based on the program.[188]

[184] *See supra* Part 1 for an in-depth discussion of the making available debate.

[185] *See* GPLv3, *supra* note 26, § 0 ("To 'propagate' a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.").

[186] *Frequently Asked Questions about the GNU Licenses*, *supra* note 29.

[187] *See supra* Part IV.B.1 (discussing cases).

[188] *Compare* GPLv3, *supra* note 26, § 5 ("You may convey a work based on the Program, or the modifications to produce it from the Program, . . . provided that . . . [y]ou must license the entire work, as a whole, under this License to anyone who comes into possession of a copy.") *with* GPLv2, *supra* note 65, § 2

Under GPLv1, a "'work based on the Program' means either the Program or any work containing the Program or a portion of it, either verbatim or with modifications."[189]

Both GPLv3 and GPLv2 expressly turn to copyright law to define derivative works. GPLv2 states that "a 'work based on the Program' means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language."[190] Under GPLv3, "To 'modify' a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a 'modified version' of the earlier work or a work 'based on' the earlier work."[191]

Since the GPL specifically incorporates definitions established by copyright law, a look at what constitutes a derivative work for software under copyright law is in order.

B. Standard Copyright Law

1. Derivative Works Under Copyright Law

The Copyright Act defines "derivative work" as:

> [A] work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, *or any other form in which a work may be recast, transformed, or adapted*. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a "derivative work."[192]

In copyright law, derivative works arise in two contexts: infringement and copyrightability. In other words, one can ask of a derivative work, "Does it infringe?" and "Can it be copyrighted?" First, the Copyright Act gives a copyright owner the exclusive right to create derivative works.[193] Anyone besides the owner who creates a derivative work without a license to do so violates the owner's copyright.[194] The exclusive right to create derivative works is a powerful and expansive one, as it limits the abilities of others to create works based on earlier works. For example, a copyright owner who

("You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also [license the work based on the program under GPLv2].").

[189] GNU GENERAL PUBLIC LICENSE: VERSION 1 § 0 (Feb. 1989), *available at* http://www.gnu.org/licenses/gpl-1.0.html [hereinafter GPLv1].

[190] GPLv2, *supra* note 65, § 0.

[191] GPLv3, *supra* note 26, § 0.

[192] 17 U.S.C. § 101 (2006).

[193] 17 U.S.C. § 103(a).

[194] *See, e.g.*, Pickett v. Prince, 207 F.3d 402, 404–05 (7th Cir. 2000).

writes a novel has the exclusive right to write a sequel to that novel, adapt that novel into a movie, or translate that novel into a different language.

Derivative works also arise in the context of copyrightability. An author who creates a derivative work can obtain a copyright in the original aspects of that work only—not the preexisting material which was created by someone else.[195] For purposes of our analysis, we will focus on the infringement aspect of derivative works rather than the copyrightability of derivative works.

Courts first examined derivative works in the software context in *Midway Manufacturing. Co. v. Artic International, Inc.*[196] Artic International sold circuit boards that sped up Midway's video game.[197] The court held that these circuit boards infringed on Midway's right to create derivative works.[198] The court discussed the issue very briefly, focusing mainly on the fact that a sped-up video game has value and a copyright owner should be entitled to monopolize the market for such a work.[199]

In *Lewis Galoob Toys, Inc. v. Nintendo of America, Inc.*,[200] the Ninth Circuit examined whether the Game Genie—a device that plugged into the Nintendo game console and allowed a user to increase the number of lives for a player's character, speed up the game, and float above obstacles—infringed on Nintendo's right to create derivative works.[201] The court answered in the negative because "[a] derivative work must incorporate a protected work in some concrete or permanent 'form.'"[202] The Game Genie worked by blocking data values sent by the game cartridge to the game console and replacing them with other values.[203] It was incapable of creating audiovisual displays on its own, and did not incorporate the protected work, but "merely enhance[ed]" it—like a kaleidoscope.[204] The court distinguished the case from *Midway* because Midway's circuit board copied and replaced the defendant's circuit board, whereas the Game Genie could not "duplicate or recast[]" Nintendo's displays.[205] Furthermore, as in *Midway*, the court employed a market effects test, noting that the Game Genie did not lessen market demand for Nintendo's games, whereas Arctic International's circuit board replaced Midway's chip and thus lessened the market for Midway's product.[206]

Six years later in *MicroStar v. FormGen, Inc.*, the court again examined derivative works in the context of videogames.[207] FormGen created a video

[195] *See* Stewart v. Abend, 495 U.S. 207, 223 (1990) ("The aspects of a derivative work added by the derivative author are that author's property, but the element drawn from the pre-existing work remains on grant from the owner of the pre-existing work.").

[196] Midway Mfg. Co. v. Artic Intern., Inc., 704 F.2d 1009, 1009 (7th Cir. 1983).

[197] *Id*. at 1010–11.

[198] *Id*. at 1013.

[199] *Id*. at 1014.

[200] Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc., 964 F.2d 965 (9th Cir. 1992).

[201] *Id*. at 967.

[202] *Id.*

[203] *Id.*

[204] *Id.* at 968–69.

[205] *Id.* at 969.

[206] *Id.*

[207] MicroStar v. FormGen Inc., 154 F.3d 1107, 1107 (9th Cir. 1998).

game where users could create new "levels" by selecting and arranging different objects, scenery, and characters found in FormGen's source art library.[208] MicroStar downloaded several hundred levels that users had uploaded to the Internet, compiled them into CDs, and sold them for profit.[209] The "levels" on the CD were actually MAP files that instructed the game to take certain images from the source art library and arrange them in a specific way to create the level.[210] The Ninth Circuit issued a preliminary injunction against MicroStar, holding that it likely infringed on FormGen's exclusive right to create derivative works.[211] The court applied a substantial similarity test, reasoning that because the levels generated by MicroStar's CD incorporated images that came from FormGen's art library, the two audiovisual works would likely be substantially similar.[212] The court further determined that MicroStar's infringing content existed in a concrete or physical form—one of the prerequisites for being a derivative work—because the MAP files themselves described in great detail what each of the levels should look like.[213] The court distinguished the case from *Lewis Galoob*, where the audiovisual displays created by combining Nintendo with the Game Genie were neither permanent nor concrete, disappearing when the game finished.[214] Importantly for purposes of this article, MicroStar argued that since the MAP files only referenced the source library's images, but did not actually incorporate them, MicroStar did not infringe.[215] The court dismissed this argument because the MAP files infringed the story behind the videogame itself, and the MAP files were essentially sequels to the story.[216]

2. *Infringement of Software under Copyright Law*

Another informative line of cases traces infringement of software outside of the derivative works context. Although these cases do not directly examine derivative works, they are helpful to determining the scope of a derivative work under the GPL. Infringement cases are relevant in two ways. First, a plaintiff must show that the defendant's work infringed plaintiff's copyrighted material in order to show a violation of an exclusive right to create derivative works.[217] In other words, "a work will be considered a derivative work only if it would be considered an infringing work if the material that it has derived from a pre-existing work had been taken without the consent of a copyright proprietor of such pre-existing work."[218]

[208] *Id.* at 1109.

[209] *Id.*

[210] *Id.* at 1110.

[211] *Id.* at 1114.

[212] *Id.* at 1112.

[213] *Id.* at 1111–12.

[214] *Id.* at 1111.

[215] *Id.* at 1112. This argument is relevant for the discussion of dynamic linking. *See infra* Part 3.

[216] *Id.*

[217] *See* Litchfield v. Spielberg, 736 F.2d 1352, 1357 (9th Cir. 1984) (stating that a derivative work must substantially incorporate protected material from the preexisting work).

[218] *MicroStar*, 154 F.3d at 1112 (quoting Mirage Editions v. Albuquerque A.R.T. Co., 856 F.2d 1341,

Second, infringement creates the need for the GPL, which raises a question of copyleft, which in turn requires an analysis of derivative works. The conceptual framework is as follows: if a work infringes on the exclusive rights of a copyright owner, the creator of that work needs a defense such as the GPL to escape liability. If a work infringes on a GPL-covered work (thus necessitating reliance on the GPL) and is also distributed, we must then look to case law to determine whether it is a derivative work and triggers the GPL's copyleft. Infringement is the first step. If the work is not infringing for any reason—because it is not substantially similar to a GPL-covered work, because there is fair use, or because the work only copied non-copyrightable aspects of the GPL-covered work such as public domain material—then the owner of that work does not face liability. Hence, they do not need the safe harbor offered by the GPL and there is no need to determine whether there is a derivative work that triggers copyleft—the GPL is not involved in this scenario.

The most influential test for software infringement was set out by the Second Circuit in *Computer Associates v. Altai*.[219] Computer Associates created a component called ADAPTER that enabled the same code to run on different operating systems.[220] An employee for Altai copied code directly from ADAPTER when Altai created its own component to ensure that its programs ran compatibly with multiple operating systems.[221] Altai realized this, removed the code, and rewrote the code using programmers who were unfamiliar with the original code.[222] The court held that both literal (source and object code) and non-literal elements of software are protected by copyright.[223] The court formulated the abstraction, filtration, and comparison test to determine whether the code infringed on plaintiff's copyright.[224] Under the test, the court dissected the software into individual levels of abstraction, filtered out those parts that were not copyrightable, and then compared the remaining elements for substantial similarity.[225]

The court abstracted the program into (1) object code; (2) source code; (3) parameter lists; (4) services required; and (5) general outline.[226] The court determined that the object code and source code were not similar at all because virtually no lines of code were identical.[227] As for the parameter lists, after subtracting out elements dictated by function and elements from the public domain, only a few lists and macros were similar—not enough to warrant infringement.[228] The services were similar but this structure "was dictated by the nature of other programs with which it was designed to interact" and

1343 (9th Cir. 1988)).

[219] Computer Assocs. Int'l v. Altai, Inc., 982 F.2d 693, 706 (2d Cir. 1992).

[220] *Id.* at 698.

[221] *Id.* at 699–700.

[222] *Id.* at 700.

[223] *Id.* at 702.

[224] *Id.* at 706–12.

[225] *Id.*

[226] *Id.* at 714.

[227] *Id.*

[228] *Id.* at 714–15.

therefore not protected.[229] The organizational charts were similar, but were so "simple and obvious" that they were dictated by *scenes a faire*.[230] The court held that there was no copyright violation.[231] The Abstraction-Filtration-Comparison Test has been followed by the Fifth,[232] Sixth,[233] and Tenth Circuits.[234]

Apple Computer, Inc. v. Microsoft Corp. set out the next test for infringement of software.[235] Apple created Lisa and Macintosh, two computers with graphical user interfaces featuring a "desktop metaphor with windows, icons and pull-down menus which can be manipulated on the screen."[236] Apple and Microsoft agreed to a license where Microsoft could use and sublicense derivative works of Windows 1.0.[237] Microsoft and its licensee, Hewlett-Packard, released new versions derived from Windows 1.0.[238] Apple sued Microsoft and Hewlett-Packard, alleging that the versions were too similar to the Macintosh operating system.[239] The court developed the analytic dissection test as a result.[240]

Under the test, plaintiff must list out the elements that are infringing.[241] Unlike the abstraction, filtration, and comparison test, a court applying the analytic dissection test only looks at the elements the plaintiff specifies.[242] The court analytically dissects those parts that are not protected by copyright (essentially the same as filtration) or that are allowed by licensing.[243] For the protected parts that remain, the court determines whether they deserve "broad" or "thin" protection.[244] Thin protection is accorded to those elements that can only be expressed in a few ways.[245] For these elements, "illicit copying could occur only if the works as a whole are virtually identical."[246] The standard for broad protection is the traditional standard of substantial similarity.[247]

The court held that many aspects of the program—such as the desktop metaphor, graphical user interface, windows, icons, and menus—were

[229] *Id.* at 715.

[230] *Id*. *Scenes a faire* refers to materials whose inclusion is caused by external factors. *Id*.

[231] *Id*.

[232] *See* Gen. Universal Sys. v. Lee, 379 F.3d 131, 142 (5th Cir. 2004) ("To assess a claim of software infringement, we have generally endorsed the 'abstraction-filtration-comparison' test first outlined by the Second Circuit in *Altai* and refined by the Tenth Circuit in *Gates Rubber Co. v. Bando Chemical Industries, Ltd.*")

[233] *See* Kohus v. Mariol, 328 F.3d 848, 855 (6th Cir. 2003) (adopting the Abstraction, Filtration, and Comparison test for use outside the software context).

[234] Gates Rubber Co. v. Bando Chem. Indus., 9 F.3d 823, 834 (10th Cir. Colo. 1993) ("In substantial part, we adopt the 'Abstraction-Filtration-Comparison' test . . . .").

[235] Apple Computer, Inc. v. Microsoft Corp., 35 F.3d 1435, 1443 (9th Cir. 1994).

[236] *Id.* at 1438.

[237] *Id.*

[238] *Id.*

[239] *Id.*

[240] *Id.* at 1443.

[241] *Id.*

[242] *Id.*

[243] *Id.*

[244] *Id.* at 1446–47.

[245] *Id.*

[246] *Id.* at 1447.

[247] *Id.* at 1442.

unprotectable ideas.[248] Many of the other infringing elements listed by Apple were covered by the license.[249] Since almost all similarities were not protectable as ideas or covered under the license, Apple could only win if the works as a whole were virtually identical.[250] Apple declined to oppose Microsoft's motion for summary judgment for lack of virtual identity and the district court entered judgment in favor of Microsoft.[251] The court of appeals affirmed the legal standard of virtual identity and declined to address the merits because Apple did not appeal on those grounds.[252]

The last test for software infringement arose in *Lotus Development Corp. v. Borland Int'l., Inc*.[253] The First Circuit found no infringement when Borland incorporated a virtually identical menu hierarchy to that of Lotus in its spreadsheet computer program.[254] The court found that the menu hierarchy was a method of operation because it provided the means by which users could control and operate the program; therefore, it could not be protected under copyright.[255]

C. What is a Derivative Work for Purposes of the GPL?

The breadth of a derivative work under the GPL is an unsettled question and a hot area of debate for commentators. Courts have yet to define what constitutes a derivative work under the GPL. However, reasoning by analogy to copyright case law, evaluating opinions of commentators, and analyzing the GPL's Frequently Asked Questions offers insight into determining the scope of a derivative work under the GPL.

1. Copying GPL-covered Source or Object Code

The most basic case of creating a derivative work under the GPL is altering or deleting a few lines of code from a GPL-covered work. Under *Computer Associates*, if the original work and the modification are substantially similar in both source code and object code, then a court would likely find infringement.[256] Furthermore, the modified work contains the copyrighted work in a permanent form—the actual lines of copyrighted code are present. *Lewis Galoob* indicates that this is sufficient to be a derivative work.[257]

Directly copying source code or object code from a GPL-covered work

[248] *Id.* at 1443–44.

[249] *Id.* at 1446 ("By virtue of the licensing agreement, Microsoft and HP were entitled to use the vast majority of features that Apple claims were copied.").

[250] *Id.* at 1447.

[251] *Id.* at 1438.

[252] *Id.* at 1439.

[253] Lotus Development Corp. v. Borland Int'l., Inc., 49 F.3d 807, 819 (1st Cir. 1995), *aff'd*, 516 U.S. 233 (1996).

[254] *Id*. at 819.

[255] *Id*. at 815.

[256] *See* Computer Assocs. Int'l, Inc. v. Altai, Inc., 982 F.2d 693, 702 (2d Cir. 1992).

[257] *See* Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc., 964 F.2d 965, 967 (9th Cir. 1992).

into a new program is trickier. Commentators generally believe such action creates a derivative work.[258] However, the question is more nuanced. It is well-established that *scenes a faire*, ideas, methods of operation, elements that can only be expressed in one way due to merger, and elements in the public domain are not protected by copyright.[259] If a programmer copies part of a GPL-covered work that is not protected by copyright, say because it is dictated by external factors, his work would not infringe on the GPL-covered work's copyright. Thus, he does not have to utilize the safe harbor of the GPL, and his work would not be considered a derivative work for purposes of the GPL.

Furthermore, if a programmer copies just a few lines of code from a GPL-covered work and adds a substantial amount of new code to it to create a substantially different program, courts may be more likely to find fair use given that the amount and substantiality of the use is minimal and the work is transformative.[260] Finally, courts have held that copying a minimal amount of source code will not create a derivative work if the two programs serve different functions.[261] This accords neatly with the analysis in *Computer Associates*—after abstracting the program into several levels, even if there is minimal overlap of source or object code, the court will not find infringement if higher levels of abstraction such as overall function, parameter lists, and services required are not similar.

However, these scenarios would be the exceptions, not the rule. Generally speaking, copying and pasting a few lines of GPL-covered code into a larger program—provided that the code is copyrightable and there is no fair use defense—will result in a derivative work for purposes of the GPL.

2. *Static Linking*

Static linking and dynamic linking describe two ways in which programs can interact. Static linking means embedding a subcomponent's code into a larger program—literally, copying and pasting lines of code from one program

[258] *See* Ron Phillips, *Deadly Combinations: A Framework for Analyzing the GPL's Viral Effect*, 25 J. MARSHALL J. COMPUTER & INFO. L. 487, 492 (2008); Jerry Epplin, *Using GPL Software in Embedded Applications*, ZDNET (Mar. 5, 2001, 12:00 AM), http://www.zdnet.com/news/using-gpl-software-in-embedded-applications/296812 (asserting that modifying Linux source code to create a new module classifies as creating a derivative); *Derivative Works*, SOFTWARE PLURALISM, http://www.law.washington.edu/lta/swp/law/derivative.html (last visited Jan. 10, 2012) (deleting or altering a few lines of source code from a GPL covered work "is probably the canonical case envisioned by drafters of the GPL").

[259] *See* 17 U.S.C. § 102(b) (2006) ("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."); *Computer Assocs*., 982 F.2d at 707–11 (ruling that expressions merging with ideas, elements dictated by *scenes a faire*, and elements in the public domain are uncopyrightable).

[260] *See* Determann, *supra* note 26, at 1430 ("If the creator of a new work takes very little of an existing work, taking only non-protectable content such as ideas or facts, or changing the original so much that the new work differs substantially from the existing work, the new creation is simply a new work of authorship and not a derivative of the existing work.").

[261] *See* Vault Corp. v. Quaid Software, Ltd., 847 F.2d 255, 267–68 (5th Cir. 1988) (copying thirty characters out of fifty pages of source code does not create a derivative work when the two programs serve different functions).

to another.[262] This is akin to copying chapters out of Lewis Carroll's *Through the Looking Glass* into another book.[263] Dynamic linking means referencing object code that separately exists.[264] This would be akin to mentioning Alice and expecting readers to consult the other work. In dynamic linking, this referencing takes place during runtime (runtime, in basic terms, is when the computer is running the programs in question).[265] Since the subcomponent is referenced by the larger program and runs separately from the larger program in dynamic linking, several programs can dynamically link to one subcomponent at once.[266]

In general, static linking creates a derivative work. Most commentators have come to this conclusion,[267] and the FSF agrees. [268] This interpretation accords with the copyright framework set out by courts. Since object code from the GPL-covered work is literally copied and pasted into the new work, the analysis would be very similar to that explained in the previous section. The new work would incorporate the previous work in a form sufficiently "concrete or permanent" for *Galoob*. The works would likely be found substantially similar, particularly with identical portions of source and object code. Furthermore, static linking incorporates and replaces the underlying work much like the chip in *Midway*, in contrast to dynamic linking, which does not physically incorporate the underlying work. This accords with the distinction set forth in *Galoob*:

> Artic's chip substantially copied and *replaced* the chip that was originally distributed by Midway. Purchasers of Artic's chip also benefited economically by offering the altered game for use by the general public. The Game Genie does not physically incorporate a portion of a copyrighted work, nor does it supplant demand for a component of that work.[269]

The rule that static linking generally creates a derivative work would still be subject to the exceptions mentioned in the previous section. If the portions

[262] Tsai, *supra* note 12, at 557.

[263] *Id.* at 557–58.

[264] *Id.*

[265] Vetter, *supra* note 38, at 103 ("'Runtime' is, for our purposes, simply when the computer is running, and when the programs of interest are executing.").

[266] Tsai, *supra* note 12, at 558.

[267] *See e.g.*, Joseph A. Chern, Note, *Testing Open Source Waters: Derivative Works Under GLPv3*, 13 CHAP. L. REV. 137, 154 (2009) ("Software that uses static linking to a GPL library warrants minimal discussion that it is a derivative work under any definition . . . . Because the software now contains the GPL code 'or a portion of it, either verbatim or with modifications,' the software is now subject to the provisions of the license."); Stoltz, *supra* note 8, at 1450 ("If a programmer writes a new module and statically links it with an existing GPL-covered program, the result is almost certainly a derivative work of the existing GPL program."); Tsai, *supra* note 12, at 558 ("Static linking almost certainly falls within the scope of GPLv2's copyleft obligation."); Matt Asay, *The GPL: Understanding the License that Governs Linux*, NOVELL.COM (Jan. 16, 2004), http://www.novell.com/coolsolutions/feature/1532.html ("The crucial thing to remember is that a static link outputs a combined file from the compiled code and the libraries. This executable would almost certainly be considered a derivative work of the GPL code.").

[268] *Frequently Asked Questions about the GNU Licenses*, *supra* note 29 ("If the modules are included in the same executable file, they are definitely combined in one program."); *Frequently Asked Questions about Version 2 of the GPL*, *supra* note 132 (same).

[269] Lewis Galoob Toys, Inc. v. Nintendo of America, Inc., 964 F.3d 965, 969 (9th Cir. 1992).

copied are *de minimis* and the author adds substantially new material to the work such that it contrasts with the previous work, courts will be more likely to find no infringement or apply a fair use exception.

3. Dynamic Linking

a. Commentator's Opinions

Commentators diverge widely as to whether dynamic linking creates a derivative work. The inconsistency is so great that one commentator has lamented that "there appear to be no definitive answers to the question of what constitutes a derivative work under the GPL, not even from the holders of the license in question."[270] On one end of the spectrum, some commentators believe that a number of reasons create a blanket exception to the GPL's copyleft for dynamically linked subcomponents. First, since dynamic linking only references a work, it does not incorporate the protected work—hence, there is no copying.[271] Second, the work does not incorporate the protected work in any concrete or permanent form sufficient for *Galoob*.[272] Furthermore, since the two programs interoperate at arm's length, and code from one program is not copied into the other one, they should be viewed as "two separate bodies of code that talk to each other only as needed."[273] Finally, even if courts find a derivative work, some commentators believe that dynamic linking should constitute a fair use.[274] The fair use factors are: (1) the purpose and character of the use; (2) the nature of the copyrighted work; (3) the amount and substantiality of the portion taken; and (4) the effect of the use upon the potential market.[275] Under these factors, one commentator has argued (1) code subject to the GPL exists to promote efficiency and functionality; (2) the nature of work subject to the GPL is that it is published and freely distributed; (3) the amount copied is only what is necessary for the program to function; and (4) such use establishes legitimate competition.[276]

Other commentators believe that dynamic linking can create a derivative work in some instances. Jerry Epplin, Principal Software Engineer at Global Velocity, Inc., adopts an expansive notion of the GPL's copyleft, believing that both dynamic and static linking activate copyleft.[277] He has stated that linking to a GPL-licensed library infects the larger program so that the larger program becomes a derivative work of the library.[278] Linus Torvalds, creator of the

[270] Matt Asay, *A Funny Thing Happened on the Way to the Market: Linux, the General Public License, and a New Model for Software Innovation*, at 17 (April 2002), *available at* http://www.linuxfordevices.com/files/misc/asay-paper.pdf.

[271] Chern, *supra* note 267, at 156 ("When a work does not incorporate any protected expression, it cannot be a derivative.").

[272] *Id*. at 157.

[273] Asay, *supra* note 270.

[274] *See, e.g.*, Chern, *supra* note 267, at 158.

[275] 17 U.S.C. § 107(1)–(4) (2006).

[276] Chern, *supra* note 267, at 158.

[277] Epplin, *supra* note 258.

[278] *Id*.

Linux operating system, does not believe that the type of linking one uses dictates whether a work is derivative.[279] Rather, he argues that one should look to whether the two programs are "independent" works.[280] For example, if a program links to a specialized library and cannot run without that specific library, it is a derivative work. If it can be linked against other libraries, it is independent.[281] Richard Stallman has also asserted that the type of linking does not matter, noting that "[i]f there are two modules that are designed to be run linked together and it's clear from the design from one or the other that they are meant to be linked together[,] then we say they are treated as one program . . . ."[282]

The FSF takes an expansive reading of derivative works. The Frequently Asked Questions to the GPL maintains that if a programmer links a program subject to the GPL with his or her own code to build a proprietary program, he or she must make the program available under the GPL or a license compatible with the GPL.[283] Note that the language does not distinguish between dynamic and static linking.

The FAQs for the second and third versions of the GPL do not directly address the question of dynamic linking as a whole; however, they do examine dynamic linking in the specific contexts of libraries and plug-ins.

b. Dynamically Linking to Libraries

Dynamic and static linking are typically used with libraries, collections of code that perform processes generally used by different programs.[284] Libraries increase efficiency by containing standard code that different programs can link to, rather than requiring that each program write similar code for an identical purpose.[285] For example, a library coding for fonts could be used with various programs, including Microsoft Word, Powerpoint, and Gmail.

Two separate questions arise for use of libraries. First, if a proprietary work dynamically links to a GPL-covered library, does that GPL-covered library infect that larger work and strip it of its proprietary nature? Second, can a GPL-covered program link to a proprietary library? In other words, since the GPL requires that the distributor distribute source code for the program, which would not be possible for a proprietary library, is it impossible to combine a GPL-covered program with a proprietary library and abide by the terms of the GPL?

The FSF addresses both questions in the FAQ. For the first question, the

[279] E-mail from Linus Torvalds to Alexandre Oliva (Dec. 17, 2006, 09:59:51 PST), *available at* http://lkml.org/lkml/2006/12/17/79.

[280] *Id*.

[281] *Id*.

[282] Richard Stallman, Remarks on GPLv3 at the Meeting on the Future of Free Software (Mar. 18, 2006) (transcript available at http://fsfe.org/projects/gplv3/torino-rms-transcript.en.html).

[283] *Frequently Asked Questions about the GNU Licenses*, *supra* note 29; *Frequently Asked Questions about Version 2 of the GPL*, *supra* note 132.

[284] Tsai, *supra* note 12, at 557.

[285] *Id*.

FAQ to both the second and third versions clarify that a proprietary work linking to a GPL-covered library will infect the larger work.[286] The Free Software Foundation advises that programmers wishing to allow proprietary software to dynamically link to a free library without tainting the proprietary software use the GNU Lesser General Public License, a license specifically targeted to this problem and so named because of its less virulent form of copyleft.[287] The GNU Lesser General Public License specifies that libraries licensed under it can be used with proprietary programs without tainting the larger work.[288] In all other respects, it is the same as the GPL.

As for the second question, the GPL offers a system library exception stating that an individual distributing a GPL covered program that dynamically links to a proprietary library does not have to provide the source code for the proprietary library; provided, however, that the proprietary library meets the specifications set forth in the exception.[289] However, GPLv3 asserts that a distributor must provide source code for libraries that lie outside the system library exception—specifically, "dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work."[290] Since this text is directly in the license itself and not in the frequently asked questions, it has more weight. In fact, because it is contained directly in the license itself, it is difficult to assert that dynamic linking is automatically exempt from the GPL's copyleft.[291] However, the change both clarifies and confuses—it is unclear what intimate data communication or control flow means. The Free Software Foundation offers no clarification, and courts and commentators have failed to explore the problem.

Copyright law does not affirmatively answer whether dynamically linking to a library creates a derivative work. For purposes of this analysis, assume that we are dealing with a situation in which a proprietary work links to a GPL-covered library—question (1) examined above. The *MicroStar* court came closest to addressing this situation when it examined whether ForumGen's MAP files created a derivative work when they referenced a source art library.[292] ForumGen argued that they did not create a derivative work because the files "reference the source art library, but do not actually contain any art files themselves."[293] The court avoided answering whether merely referencing

[286] *See Frequently Asked Questions about the GNU Licenses, supra* note 29; *Frequently Asked Questions about Version 2 of the GPL, supra* note 132 (Q: "If a library is released under the GPL (not the LGPL), does that mean that any program which uses it has to be under the GPL?" A: "Yes, because the program as it is actually run includes the library.").

[287] Rowan Wilson, *The GNU Lesser General Public License Version 2.1:-An Overview,* OSS WATCH (Feb. 8, 2011), http://www.oss-watch.ac.uk/resources/lgpl.xml.

[288] *See Frequently Asked Questions about the GNU Licenses, supra* note 29.

[289] *See id.* (discussing the system library exception).

[290] GPLv3, supra note 26, § 0 (defining "Corresponding Source").

[291] Tsai, *supra* note 12, at 566 ("[U]sers, developers, and distributors can no longer advocate a categorical exclusion of dynamic linking from the scope of the copyleft obligation under GPLv3, as some did under the previous license.").

[292] MicroStar v. ForumGen, 154 F.3d 1107, 1114 (9th Cir. 1998).

[293] *Id.* at 1112.

the art files was sufficient to create a derivative work, instead finding a derivative work because ForumGen infringed on the story behind the videogame.[294] It is therefore unclear from *MicroStar* how a court would rule when a derivative work does not infringe on the underlying work's story and merely references the underlying work, as is the case with dynamic linking.

On the one hand, one could make a persuasive argument that dynamic linking does not create a derivative work. Under copyright law, merely referencing a library may not incorporate that library in a sufficiently concrete or permanent state for *Galoob*. Furthermore, the market effects test utilized by both the *Midway* and *Galoob* courts further suggests that dynamic linking does not create a derivative work. In *Galoob,* the court clarified that a work is more likely to infringe the exclusive right to create derivative works if it supplants purchasers' demand for the underlying works and emphasized that the market effects test can be more important than the concreteness/permanency test.[295] Under the market effects test, static linking would probably supplant demand for the underlying library since the library is copied directly into the program. Dynamic linking, in contrast, would not supplant demand for the underlying library. Rather, it would probably *increase* demand because users must obtain the library (either in conjunction with the larger program or separately) and combine the two in the runtime environment.

On the other hand, one could argue that dynamic linking can create derivative works in specific situations. For example, dynamic linking can incorporate header files, which may be sufficiently permanent and concrete for *Galoob*. Header files indicate to the main program which functions and names will be provided by the library.[296] Commentator Van Lindberg explains their function as follows:

> Assume that you have a set of directions (your source code) that include one instruction to "go to Jane's house." You cannot follow your directions from start to finish unless at some point somebody tells you where Jane's house is located. The header file is like a letter that says, "I will tell you the address of Jane's house right before you start actually following the directions. Until then, just accept that you will go to Jane's house during your trip."[297]

The header file must be completely incorporated into the program, which may be enough for a court to find a derivative work even in the case of dynamic linking.

On the other hand, one could argue that header files are not protectable under copyright because they are purely functional or are methods of operation under *Lotus*. Furthermore, header files are incorporated into the larger program only in some instances of dynamic linking. Dynamic linking can occur without header files, or alternatively, a GPL-covered library may

[294] *Id*.

[295] *See* Lewis Galoob Toys, Inc. v. Nintendo of Am., Inc., 964 F.2d 965, 969 (9th Cir. 1992).

[296] LINDBERG, *supra* note 133, at 233.

[297] *Id*.

incorporate header files from the main program.[298] This occurs when a program does not change to interoperate with various libraries; a programmer must write the library to conform to the larger program and correspondingly must adopt the larger program's header files.[299] In this scenario, the main program does not incorporate any portion of the GPL-covered library; rather, the GPL-covered library incorporates a portion of the main program. A derivative work should not be found.

Overall, one cannot conclusively state whether dynamic linking to a library creates a derivative work. Companies seeking to keep code proprietary can take a few precautionary steps. First, they should seek to use dynamic linking that does not incorporate header files or other interfaces from a GPL-covered library. Second, they should distribute the proprietary program alone, not in conjunction with the compatible GPL-covered library. The user can separately obtain the GPL-covered library and combine the two programs at runtime. This would not trigger copyleft because the user does not distribute the combination, but only uses it privately.[300]

c. Dynamically Linked Plug-ins

The FAQs for both versions two and three also clarify dynamic linking in the particular context of plug-ins. Plug-ins are add-ons that allow a user to extend the functionality of a program. For example, the plug-ins Adobe Acrobat and Flash extend Mozilla Firefox's functionality. Programmers creating proprietary plug-ins for GPL-covered programs must follow a confusing spectrum set out by the GPL's FAQ. On one end of the spectrum, if plug-ins communicate with the main program via fork and exec, the two programs are separate, and the plug-in can be proprietary.[301] On the other hand, a program that dynamically links plug-ins is considered a derivative work if the plug-in and program "make function calls to each other and share data structures."[302] Finally, a "borderline case" exists when "the communication between them is limited to invoking the 'main' function of the

[298] *Id*. at 234.

[299] *Id*.

[300] Commentators agree that such a practice should not invoke the GPL's copyleft. *See* LINDBERG, *supra* note 133, at 236 ("The GPL does not apply to binaries that are not distributed as a single program, and the GPL places no restrictions on the end user's ability to combine the GPL-licensed and non-GPL-licensed programs into a single binary at runtime."); Phillips, *supra* note 258, at 500–01 ("Proprietary software that incorporates GPL code as a dynamically linked library probably creates a legally derivative work, but likely will not trigger the GPL's copyleft terms if the library is not distributed with the proprietary software. The proprietary program by itself does not contain the object code from the GPL software and can easily be distributed without the dynamically linked library. The end user can get the library independently and install it on his or her computer, all the while complying with the GPL."); Chern, *supra* note 267, at 157 ("[A]ssume Program A is meant to be maintained as a proprietary word processing program, while Program B is a GPL-ed printing program . . . . User X of Program A . . . can independently acquire Program B and use it privately with no obligation. As a result, even if Program A and B are combined in memory when the computer is running both of them, there is no 'copyleft' obligation for this private use.").

[301] *Frequently Asked Questions about the GNU Licenses*, *supra* note 29; *Frequently Asked Questions about the GNU GPL v2.0*, *supra* note 132.

[302] *Frequently Asked Questions about the GNU Licenses*, *supra* note 29.

plug-in with some options and waiting for it to return."[303] Note that a GPL-covered plug-in probably could not infect the main program: a main program would not be considered a derivative work of the plug-in because a derivative work must be based on a "preexisting" work.[304] Here, the main program would come first.

While the FSF sets out a highly technical spectrum, copyright law would take a different approach to plug-ins, focusing "on the abstract inquiry of whether the challenged software is a derivative work rather than on the underlying technology."[305] Here again, the inquiry would focus on whether the main program incorporates the plug-in in a concrete or permanent form. If yes, the court would ask whether the incorporated portion was copyrightable, or if it was purely functional, dictated by external factors, dictated by efficiency, a method of operation, etc. If the incorporated portion is not copyrightable, no derivative work will be found. Thus, whether or not a plug-in is a derivative work would be a fact-specific inquiry that gives even fewer answers than the FSF's spectrum approach.

4. Loadable Kernel Modules

Derivative works also arise in the context of loadable kernel modules. The Linux kernel is a core operating system licensed under GPLv2. Linux kernel modules are add-on modules that extend the running of the Linux kernel. For example, kernel modules can add support for new hardware or file systems. Interestingly, the Linux license states: "This copyright does *not* cover user programs that use kernel services by normal system calls—this is merely considered normal use of the kernel, and does *not* fall under the heading of 'derived work.'"[306] The text precedes the preamble to the GPL.

Commentators are divided as to whether this language deviates from the GPL or whether it simply clarifies a gray area in the GPL.[307] Other commentators maintain that proprietary loadable kernel modules violate the GPL.[308] Linux developers and others in the Free Software Community believe

[303] *Id*.

[304] Greg Wilson, *The Chilling Effect of the GPL*, THIRD BIT (April 12, 2010), http://third-bit.com/blog/archives/3767.html.

[305] Chern, *supra* note 267, at 156.

[306] GNU GENERAL PUBLIC LICENSE: VERSION 2, LINUX KERNEL (June 1991), *available at* http://www.kernel.org/pub/linux/kernel/COPYING.

[307] Interview by Joe Brockmeier with Eben Moglen, President and Exec. Dir., Software Freedom Law Center (Aug. 11, 2005), *available at* http://lwn.net/Articles/147070/ ("If the kernel is pure GPL, then I think we would all agree that non-GPL, non-free loadable kernel modules represent GPL violations. Nonetheless, we all know that there are a large number of such modules and their existence is tolerated or even to some degree encouraged by the kernel maintainers, and I take that to mean that as an indication that there is some exception for those modules."); E-mail from Linus Torvalds, Project Coord. for Linux, to Paco Moya, (Dec. 17, 1995) ("No such exception exists. There's a clarification that user-space programs that use the standard system call interfaces aren't considered derived works, but even that isn't an 'exception'—it's just a statement of a border of what is clearly considered a 'derived work.'").

[308] Michaelson & Holst, *supra* note 134, at 1 (stating that proprietary kernel modules violate the GPL and the only thing stopping suit against creators of proprietary LKMs "is the decision of the license's primary enforcers to look the other way").

kernel modules to be derivative works. In fact, Linux developers have called for an end to proprietary software drivers, signing a position statement[309] and even threatening to ban them by 2008.[310]

Linus Torvalds maintains that the language does not create an exception to the GPL, but merely crystallizes the ready-known fact that programs using kernel services via normal system calls are not derivatives. He has adopted his own specific test to determine whether a work is derivative, focusing on the two dual concepts of specialization and intimacy.[311] Specialization involves whether a module or program was specialized for Linux and designed with Linux in mind. Intimacy deals with how distantly the program interacts with the Linux kernel.

According to Torvalds, user-space programs that use kernel services by normal system calls are both are distant enough from the kernel and standard enough that they are not derivative.[312] Kernel modules are distant from the kernel since they are a separable module. However, whether or not they are derivative still hinges on their degree of specialization: whether they are specialized to Linux and designed with Linux in mind, or whether they are standard and can be interchanged with several different operating systems. Non-modules, on the other hand, programs that "are so central to the kernel that you can't load them as a module" are "clearly derived works just by virtue of being very intimate"[313]—regardless of whether they are specialized or not. Correspondingly, kernel patches—lines of code that fix problems in the kernel itself, or programs that interact closely with core kernel code[314]—are intimate enough to be derivative. Along the same lines, a highly specialized work written with Linux in mind is derivative, regardless of how intimate it is. Programs that were not written with Linux in mind but work with Linux due to its similarity with other interfaces are gray areas. His interpretation has been adopted by others in the FOSS community.[315] Companies such as Nvidia have released proprietary device drivers for hardware built on the Linux operating system.[316] Linux has tolerated them.

Importantly, Linus Torvald's interpretation, based on the twin concepts of intimacy and specialization, have no basis in copyright law. Copyright law's analysis of derivative works focuses on substantial similarity, rather than intimacy and specialization.[317] Under copyright law, even if a work is

[309] Jeremy Andrews, *Position Statement on Linux Kernel Modules*, KERNEL TRAP (June 23, 2008, 9:27 AM), http://kerneltrap.org/Linux/Position_Statement_on_Linux_Kernel_Modules.

[310] Ryan Paul, *Linus Won't Ban Binary Kernel Modules*, ARS TECHNICA, (Dec. 15, 2006, 8:46 AM), http://arstechnica.com/business/news/2006/12/8428.ars.

[311] Michaelson & Holst, *supra* note 134, at 7 (quoting emails from Linus Torvald).

[312] *See id.* at 7–8.

[313] PICHAI RAGHAVAN ET AL., EMBEDDED LINUX SYSTEM DESIGN AND DEVELOPMENT 390 (Auerbach Publ'n ed., 2006) (quoting Email from Linus Torvalds, Project Coord. for Linux (Dec. 3, 2003, 16:10 PST)).

[314] *See* Andrews, *supra* note 309 (discussing kernel patches).

[315] *See* Epplin, *supra* note 258 (discussing Linux programming and how proprietary device drivers that use the normal driver/kernel interfaces are accepted).

[316] Paul, *supra* note 310.

[317] Morgan, *supra* note 38, at 490 ("[T]he terminology used by the FSF is very difficult to analyze from a copyright perspective since copyright law does not contemplate matters such as the intimacy of

developed with a specific copyrighted work "in mind," it will not be a derivative work if it does not incorporate copyrightable aspects of that copyrighted work in a concrete or permanent form.[318] Similarly, courts have not relied on interoperability to determine whether work is derivative.

Because Torvald's interpretation deviates from that of copyright law, the impact the statement has depends on whether it is considered part of the license or a mere clarification. If the statement is merely a clarification and not part of the license, assuming that the GPL is a license rather than a contract, to the extent that Torvald's interpretation differs from copyright law, such a use will not be protected. Put another way, if user programs interacting normally with the kernel are considered derivative works under copyright law, a purely copyright-driven regime would find that such a use infringes on the exclusive right to create derivative works. A user would then need to utilize the conditional permission offered by the GPL, and proprietary kernel modules would therefore not be allowed under the GPL. On the other hand, even under a pure license regime, a court may find it inequitable to penalize creators of proprietary kernel modules when the preamble to the license itself claims that such an action will not trigger copyleft.

On the other hand, one may take the position that the statement is part of the license itself and that Torvalds created a variation on the GPL that allows proprietary loadable kernel modules.[319] In this situation, even if loadable kernel modules are derivative works under copyright law, the license itself allows users to create LKMs without triggering the GPL's copyleft—thus, there is no relief for copyright holders.

5. *Aggregates*

Two programs distributed together on the same CD not interacting at all would not be considered derivative works under almost all interpretations, including that of the FSF.[320]

By analogy, software that exists in the same virtualized cloud computing environment but does not interact will probably not be considered derivative works. One work would not be substantially similar to the other and one would not incorporate the other in a sufficiently permanent medium for *Galoob*.

VI. CONCLUSION

FOSS, particularly FOSS released under the GPL, has had unprecedented influence since its creation. Constant innovation by the FOSS community has

communication or combining two parts into a larger program.").

[318] See Part V.B.1 (discussing cases).

[319] Vetter, *supra* note 38, at 166 ("If one interprets Torvalds' 'immunization for normal system calls' notice as a license permission, then one characterization is that Torvalds wrote his own license. He did so by copying the GPL, but then further specified a technical scenario where the GPL's infectious terms did not apply, resulting in the GPL-INC (GPL with immunization for normal calls).").

[320] *See* GPLv3, *supra* note 26, § 5 (defining an "aggregate").

enabled FOSS to become an integral part of today's computing world—and a pervasive and potentially threatening force for commercial software companies. Despite the GPL's popularity, much uncertainty shrouds the copyleft clause and its twin triggers of distribution and derivative works. This Article suggests applying the limited publication doctrine to distribution under the GPL in order to clarify the scope of the term and reduce traps for the unwary while remaining consistent with the principles outlined in the FAQ.

Commercial software companies can adopt a few precautionary measures to lower the risk of invoking the GPL's copyleft. Companies should run Black Duck software on their proprietary software to determine how much of it is based on FOSS. They should refrain from distributing any software that statically links to a GPL-covered program, as static linking almost certainly creates a derivative work. If statically-linked software is to be distributed, companies should remove the statically linked portions before distribution, either by rewriting new code to fill in those portions or by switching to dynamic linking. Companies should attempt to link to libraries licensed under the LGPL, if at all possible.

Companies should offer software as a service rather than available for download. Any transfers to third parties made for purposes of outsourcing or cloud computing should be carefully regulated. A contractor should not be permitted to copy, sell, lease, or distribute proprietary software. Furthermore, a contractor should be restricted from using proprietary software for any purpose other than within the scope of his relationship with the company. These precautionary steps may help curb excessive fears about the scope of copyleft and promote the efficient use of free and open source software.