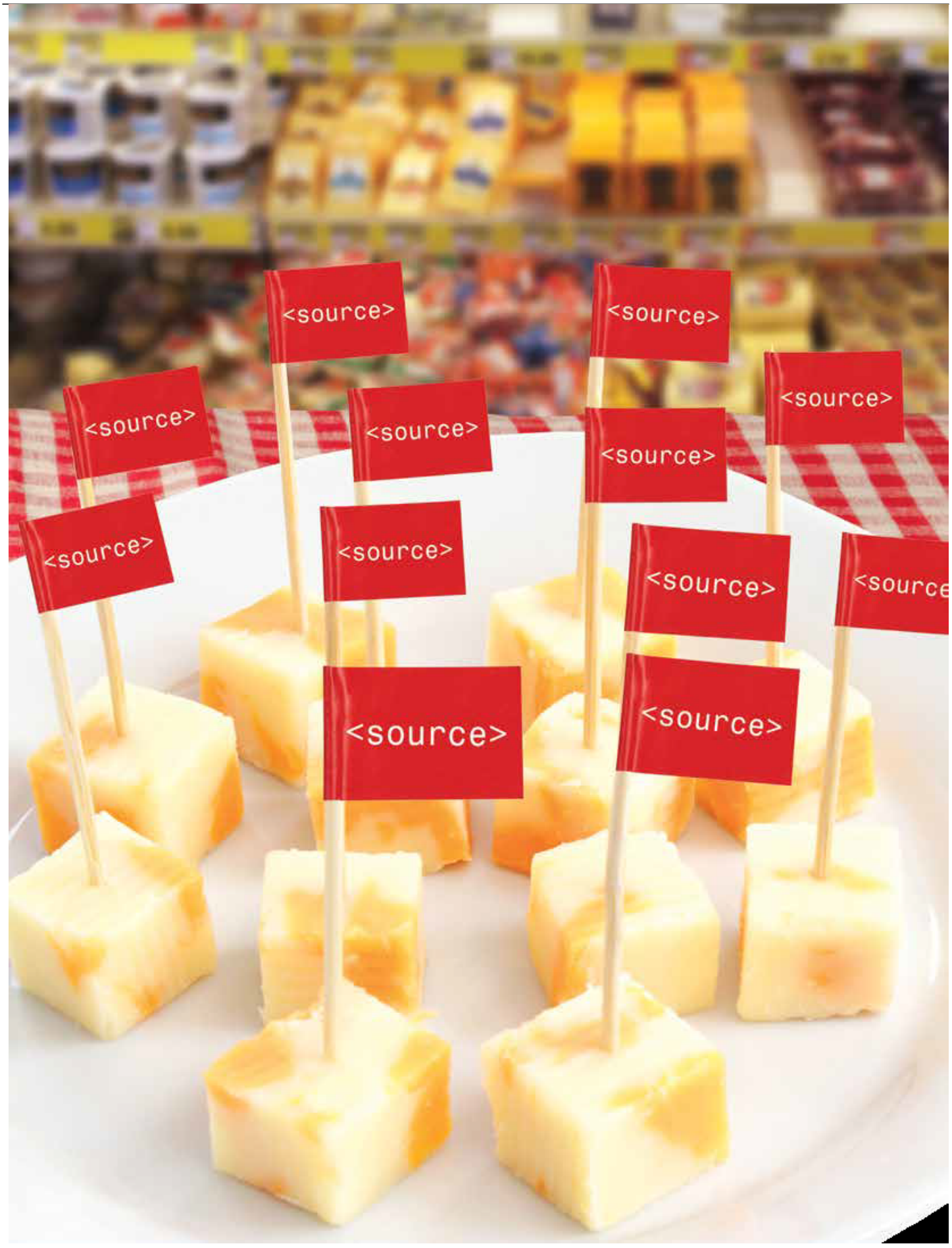
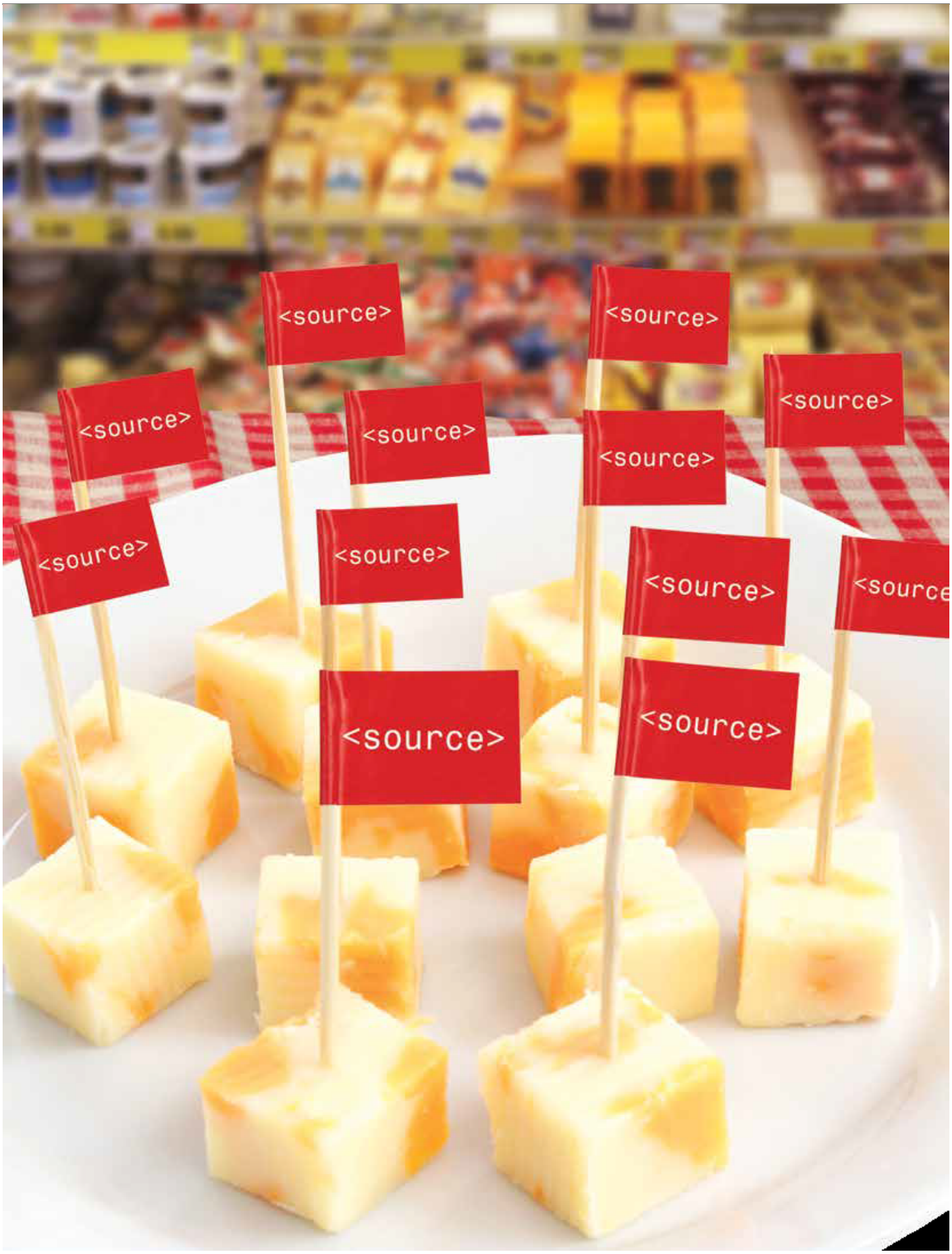
Minimizing Copyright Risks When Using Open Source Software

Litigation and Dispute Resolution

Technology, Privacy, and eCommerce


<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>


<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>
<source>

CHEAT SHEET

- ***Not so open after all.*** Open source software developers do retain control over their code through copyright law.
- ***Create a policy.*** To ensure a consistent treatment of open source software, establish a uniform company policy, and train your software engineers in the policy and approved open source licenses.
- ***Beware third parties.*** Perform a careful review of any software that your company acquires from third parties, and include an open source policy and compliance review of software as part of your due diligence for mergers and acquisitions.
- ***Be critical.*** Carefully review the open source license, being especially wary of infringement indemnification provisions.

“Quick question,” an IT employee says as he sticks his head in your office. “We’re working around the clock on a new database software product, and it’ll launch in 30 days. The software has open source components — they’re bug-free and cheap, and they’re what we want to use. Is that okay?” Your response, as in-house counsel, should be neither quick nor simple. Facing a looming deadline, determined developers and the apparent innocence of this request, you need to approach the situation with a certain degree of caution. To ensure the best interests of your corporation, your response should be based on an awareness of the benefits and legal risks associated with the use of open source software.

Open source software is being incorporated into commercial software products at increasing rates. Mainstream examples of open source software include the Linux operating system, the Firefox Web browser and the Apache Web server. Although open source software is associated with a sense of public-domain freedom, open source software is in fact licensed with a variety of restrictions. Violating these restrictions could result in copyright liability for your company, and could even put the proprietary status of your company’s software at risk. Releasing your own company’s software under an open source license can also invite certain benefits and risks, depending on which type of license you select. Familiarity with open source software and its copyright implications will help you protect your company from liability and enable it to make more informed decisions.

Although open source software also carries risks of patent liability, this article focuses on the unique interplay between open source software and copyright law and suggests best practices to limit your company’s potential copyright liability.

What is open source software?

The first key to using open source software wisely is understanding what it is — and what it is not. *Open source software* in this article encompasses software with openly shared *source code*, which anyone can use, copy, modify and distribute. When consumers buy software, they receive the *object code*, which is readable only by machines. The underlying *source code* is readable by humans, and thus sharing it allows third-party developers to study and modify the code.

You may see varying definitions and terminology referring to open source software. OSS, for example, is a common abbreviation for *open source software*. *Free software* generally refers to software that anyone can freely license to use, copy, study and modify in any way. *Free and open source software*, or *FOSS*, qualifies as both free and open source. This article uses the term open source software to refer to all software with source code that is openly shared and that anyone can freely license to use, copy, modify and distribute.

Open source software fosters widespread collaboration, allowing users to modify and debug its code quickly, cheaply and creatively. In return for sharing the source code, open source developers receive the benefit of essentially crowd sourced assistance from third-party developers. Attracted by its reliability and cost-effectiveness, many companies now incorporate open source code into their proprietary software, appreciating that it has been vetted and debugged by an open source community.

If your company develops software, it is important to understand the implications of using or making open source software and the best practices for limiting your company's liability from a copyright perspective.

Common open source licenses

Rather than abandoning copyright protection in favor of dedicating code to the public domain, open source software developers retain a certain degree of control over their code through copyright law. Developers release the code under an open source copyright license, which places restrictions on anyone using or modifying the code. The terms of the open source license are included in the code itself. Open source licenses often restrict the use, modification and redistribution of the code.

There are a variety of different types of open source licenses, each with its own permissions and restrictions. One of the most widely used open source licenses is the Free Software Foundation's ("FSF") GNU General Public License (GPL), which was originally designed for use with GNU operating system programs. The GPL permits users to copy, distribute and modify its software. The GPL's key restriction is that any distributed or modified software containing GPL code must also be released under the GPL. Thus, proprietary code cannot incorporate GPL code, as it would then need to be released under the GPL subject to its permissive terms. A related FSF license, the GNU Lesser General Public License (LGPL), differs from the GPL in that it explicitly allows users to link to LGPL code from proprietary software, under conditions stated in the license, without requiring that the linking software be licensed under the LGPL.

More permissive software licenses have fewer requirements for the distribution of derivative software. The Berkeley Software Distribution (BSD) license, developed at the University of California, Berkeley, allows anyone to modify the code and release derivative software commercially. Similarly, the broad MIT License, published by the Massachusetts Institute of Technology, requires only that its short copyright and permission notice remain included with software incorporating the licensed material. The Artistic License, released by the Perl Foundation, allows modification and commercial distribution subject to certain requirements, including the retention of the code's copyright notices, attribution to the original authors and a record tracking modifications to the original code.

Certain open source licenses allow licensors to add supplementary provisions to the license. For example, open source licensors sometimes add supplementary terms requiring payment for the use of their code in a derivative commercial work.

Copyright protection of open source software

Courts have questioned whether licensors can pursue copyright infringement claims against licensees who breached an open source software license. Under traditional copyright law in the United States,* if a copyright license is limited in scope and the licensee acts outside of this scope, the licensor can pursue remedies under both contract and copyright law. But if a licensor grants a nonexclusive license that is not limited in scope, then the licensor waives its right to pursue copyright remedies and must rely only on contract law for any breach of the license. To determine whether a copyright license is sufficiently limited in scope, courts examine whether the license terms serve as actual conditions or merely as covenants, independent limitations that do not limit the scope of the license. If the license is sufficiently limited in scope, then copyright law arms licensors with the ability to bring suits in federal court and obtain preliminary and permanent injunctions more easily. Copyright law also provides licensors with statutory damages rather than potentially unquantifiable actual damages. Copyright owners can also use the Digital Millennium Copyright Act (DMCA) as a fast, inexpensive method of removing access to infringing material online. An open source software licensor would thus seek to show that its license is sufficiently limited in scope so that any breach can be remedied under copyright law.

* This article focuses on analyzing open source licenses under US copyright law.

Jacobsen v. Katzer is the leading decision on whether copyright claims can arise from breaches of open source licenses. In that case, the district court had found that the plaintiff's use of the Artistic License, an open source software license, granted the public a nonexclusive license to use, distribute, and copy the work. The district court held that this license was "intentionally broad" and without sufficient limits to give rise to copyright claims.

The Federal Circuit Court of Appeals, however, vacated and remanded the district court's opinion and held that the plaintiff's open source license was a valid, sufficiently limited copyright license. Although the plaintiff shared this code free of charge, the Federal Circuit found that the license had conditions necessary to protect the plaintiff's economic rights. The license's required copyright notices directed downstream users to the plaintiff's website and the license's required modification tracking enabled the plaintiff to benefit from downstream users. Here, the court held that the plaintiff could assert copyright claims against the defendant, who had disregarded these conditions while modifying and distributing the code, thus exceeding the scope of the license. This case affirmed that licensors could seek copyright remedies for breaches of open source software licenses.

Open source licensors may also face challenges to whether certain elements of their software are eligible for copyright protection. In *Oracle America, Inc. v. Google Inc.*, Oracle sued Google for copyright infringement of 37 Java application programming interface ("API") packages of open source code used within Google's Android mobile operating system. The district court had held that API programs are not copyrightable because they are equivalent to a "system [or] method of operation" and thus ineligible for protection under the Copyright Act. On appeal, the Federal Circuit held that each of Oracle's API programs was sufficiently original and non-trivial to obtain copyright protection. The court noted that an author could have written these programs in a number of different ways. Google filed a Petition for Certiorari to the US Supreme Court on the question of whether Oracle's open source Java API packages are copyrightable, but the petition was denied in June 2015.

Given that the Federal Circuit has armed open source licensors with copyright remedies, and affirmed that types of software including API packages are copyrightable, the risk of breaching an

open source software license cannot be taken lightly. Work with your company's developers to evaluate whether to use or release open source software and to ensure compliance with all applicable open source software licenses. Below are a number of best practices to consider when your developers would like to incorporate or make open source software.

Develop an open source policy

Establish a uniform company policy on open source software to curtail case-by-case decisions and create consistency for your organization. The policy will depend on your company's goals and business objectives, including whether it sells commercial software. All employees, in particular software developers, should be aware of the policy and understand whom to contact when considering whether to use or make open source software. Software engineers are often unaware of the serious implications of incorporating open source software, and training on the topic should be incorporated into mandatory employee training.

Require that each open source license undergo legal review for the specific context of its proposed use. To promote efficiency, you could create a list of approved open source software licenses. In general, the review and approval must be based on a consideration of whether your company releases software under open source licenses, and if so, its preferred type of open source license.

Start-up software companies may be tempted to delay creating an open source policy until the company becomes more established, but such a delay could be costly. When considering whether to acquire a start-up, large companies are increasingly wary of the risks associated with open source code. Buyers look for open source software policies with consistent compliance efforts, and may search the seller's software to determine whether it incorporates open source code.

Audit existing software for open source terms

If your company has not maintained a consistent policy toward open source software in the past, conduct an audit of its software. An audit will help you determine whether your company's software, particularly if commercially released, contains any open source code that requires compliance and monitoring. Depending on the volume of software being searched, your company may consider using technology that scans for open source code, such as the audit tools offered by Black Duck, Palamida, or FOSSology.

Although potentially a considerable amount of work, performing an audit now can avoid costly litigation later. Several interrelated lawsuits recently brought to light the risks of incorporating GPL code into a company's proprietary software. Amidst a lawsuit arising from a software license dispute, the defendant discovered that the plaintiff's software improperly incorporated code authored by XimpleWare Corp. and licensed under the GPL. The defendant counterclaimed that the plaintiff's incorporation of GPL code subjected the plaintiff's software to the GPL, which the plaintiff had breached. Meanwhile, XimpleWare learned of the violation and filed copyright and patent infringement lawsuits against the plaintiff, the defendant and a number of the plaintiff's other software buyers. Here, the plaintiff's incorporation of GPL code into its software had a snowball effect — first threatening the plaintiff's position in its initial software license dispute, and then compelling the plaintiff to defend itself from patent and copyright infringement lawsuits, which were settled in February 2015. To avoid this type of costly litigation, an audit may reveal your company's unknowing incorporation of open source code before any lawsuits are filed.

Manage third parties

Perform a careful review of any software that your company acquires from third parties, and include an open source policy and compliance review of software as part of your due diligence for mergers and acquisitions. If your company uses vendors or other contractors to help develop its software, include your open source software policy in your contracts with them and require that they obtain your consent for the incorporation of open source software in their work.

Gather relevant facts on the proposed use

When an employee approaches you about potentially using open source code, ask questions to uncover the full context of the proposed use. Ask how the code will be used. Ask whether the code will be used only internally or offered for commercial sale. Most open source software license terms apply only when the licensee modifies or distributes the code, and would not apply to internal use of the software by your company.

You should also ask whether the code will be integrated into the software before the object code is created, or whether the open source object code will be built separately and then connected to your software through a link. If linked, ask whether the code will be integrated through a *static* or *dynamic* link, as the type of integration may affect the software's status under certain types of licenses. Computer languages that *statically* link to a licensor's code essentially incorporate the code into your company's software in one intermingled work after compiling into object code, thus likely creating a derivative work of the licensor's code. *Dynamically* linking to code allows access to the code without incorporating it into one assembled work. Although opinions differ and there appears to be no case law on the topic, there are good arguments that *dynamic* linking does not give rise to a derivative work.

Given that any derivative work of software licensed under the GPL must also be licensed under the GPL, losing its proprietary status, it is advisable to avoid linking to GPL code in any commercial software. In fact, the FSF, which publishes the GPL, states that either *static* or *dynamic* linking to GPL code creates a derivative work subject to the GPL. Not all commentators agree with this assessment, however, and the question remains open in the courts. At the very least, your company should be careful not to link *statically* to GPL code, and understand the risk that even a *dynamic* link to GPL code may subject your software to the GPL as a derivative work. For the LGPL, the FSF states that your proprietary software can *dynamically* link to LGPL code in a library already on the user's computer without creating a derivative work.

Perform due diligence on the source of the software

Research the source of the code to determine its quality and reliability. If the software is available for download online, review the licensor's website. Given that any downloadable file on the Internet could contain a virus, malware or other security flaws, review any information available about the source before the download. Developers vary widely in size, resources and available support and quality controls. Whereas popular programs are actively monitored and constantly improving, smaller programs may be less developed, although you may find that they have active contributor communities online.

If you are concerned about the quality of an open source code provider, research whether it offers supplemental warranties or other assurances of quality. Your company can also consider purchasing

insurance that specifically manages the risks inherent in using open source code.

Popular open source software licenses

This brief synopsis highlights some — but not all — features of today's most popular open source software licenses.

GNU GENERAL PUBLIC LICENSE (GPL)

- Popular versions: 2.0; 3.0
- Any distributed or modified software incorporating GPL code must also be released under the GPL.

GNU AFFERO GENERAL PUBLIC LICENSE (AGPL)

- Popular versions: 1.0; 3.0 (2.0 is merely a transitional license)
- Based on GPL, but with different, more stringent, requirements.
- AGPLv3 addresses use of software over a computer network, requiring that all source code be made available to any network user of the work.

GNU LESSER GENERAL PUBLIC LICENSE (LGPL)

- Popular versions: 2.1, 3.0
- Proprietary software can link to LPGL code under certain circumstances without being released under the LGPL.

MIT LICENSE

- Broad license requiring only that its short copyright and permission notice remain included with software incorporating its code.

APACHE LICENSE

- Popular version: 2.0
- Grants users perpetual, irrevocable, worldwide rights for no fee or royalty.
- Redistribution of code requires attribution credit and maintaining the license.

BERKELEY SOFTWARE DEVELOPMENT (BSD) LICENSE

- Popular versions: 1-Clause ("Simplified" or "FreeBSD"); 3-Clause ("New," "Modified" or "Revised")
- Permissive license with fewer restrictions on distribution than the GNU licenses.
- Anyone can modify BSD code and release derivative software commercially with proper attribution.

MOZILLA PUBLIC LICENSE

- Popular version: 2.0
- Allows developers to intermingle its code with proprietary software.

ARTISTIC LICENSE

- The standard implementation of the Perl computer language.
- Allows modification and commercial distribution subject to certain requirements, including the retention of the code's copyright notices, attribution to the original authors, and a record tracking modifications to the original code.

ECLIPSE PUBLIC LICENSE (EPL)

- Business-friendly license that allows linked derivative works to select their own licenses.

Analyze the license

Invest the time to review the license, which can often be found on the same website on which the program is downloaded. Determine whether this license allows commercial use, and if so, whether such use incurs special requirements such as attribution or a fee-based version of the code. Be particularly wary of infringement indemnification provisions that require users who use the code in a commercial product to indemnify all other contributors to the code from legal claims, including intellectual property infringement actions. Also be alert to the fact that licenses may have multiple versions with different terms that may change your analysis and that these licenses are often subject to change without notice.

Manage compliance

Once you understand the terms of the license, put procedures in place to ensure compliance. Discuss the license with your company's developers, who may not realize that violating the terms of an open source license could subject your company to an injunction and damages. Explain that incorporation of GPL code, particularly through a static link but potentially even under a dynamic link, could render your company's software a derivative work subject to the GPL itself. The GPL allows anyone to copy, distribute and modify the code, which could frustrate your company's goals for the product and destroy its value.

Document your findings and analysis

As you research an open source software license, document all of your findings and analysis meticulously. Record the website on which the code was found and the exact name and version number of the license, as open source software developers frequently post updated versions of their software. Some licenses, such as the Artistic License, require tracking and reporting of all modifications made to the code. For this type of license, ensure that there is a procedure in place to comply with this requirement.

If you receive a large volume of open source code requests, create a searchable repository of these requests to ensure consistency in their approval. Developers can then search this repository to see whether earlier requests for the same or similar code was accepted or rejected.

Releasing software under an open source license

If your company is considering releasing software under an open source license, review the different types of available licenses to determine which license will best fit your software. The GPL, for example, provides several strict permissions and restrictions, and will apply to the entire code and any derivative works. A more moderate open source license allows use of the code as part of a larger proprietary software product. Several open source licenses cover issues related to patent licensing and litigation, as well as legal jurisdiction for any disputes that may arise.

When it comes to open source software, the more diligence and policy shaping you do at the outset, the better protection you will provide for your company. In this area, forewarned is definitely forearmed.

Further Reading

The name of the GNU operating system, a Unix-compatible system, “is a recursive acronym meaning GNU’s Not Unix.” Free Software Foundation, About the GNU Operating System, (last updated Apr. 12, 2014).

Jacobsen v. Katzer, 535 F.3d 1373 (Fed. Cir. 2008).

Jacobsen v. Katzer, No. C 06-01905, 2007 WL 2358628, at *7 (N.D. Cal. Aug. 17, 2007).

Oracle Am., Inc. v. Google Inc., 750 F.3d 1339 (Fed. Cir. 2014); cert. denied (June 29, 2015).

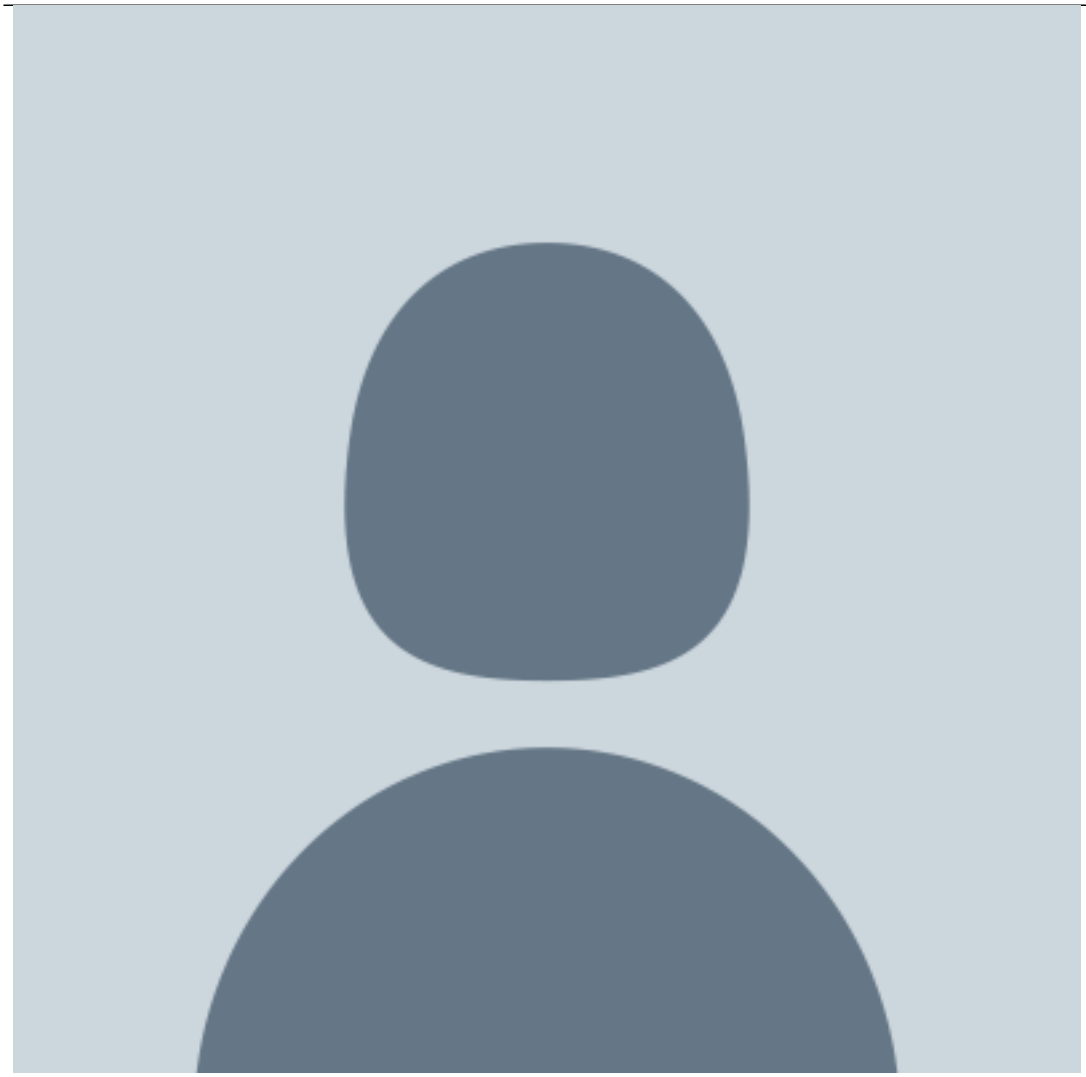
See 17 U.S.C. § 102(b).

XimpleWare Corp. v. Versata Software, Inc. et al., No. 13-cv-05160-SI, 2014 WL 6681163 (N.D. Cal. Nov. 25, 2014).

Free Software Foundation, Frequently Asked Questions about the GNU Licenses, (“Linking a GPL covered work statically or dynamically with other modules is making a combined work based on the GPL covered work. Thus, the terms and conditions of the GNU General Public License cover the whole combination.”) (last updated June 2, 2015).

Lothar Determann, Dangerous Liaisons—Software Combinations As Derivative Works? Distribution, Installation, And Execution Of Linked Programs Under Copyright Law, Commercial Licenses, And The GPL, BERKELEY TECH. L. J. Vol. 21:4, at 1465, 1496 (Sept. 2006).

Lauren Bruzzone

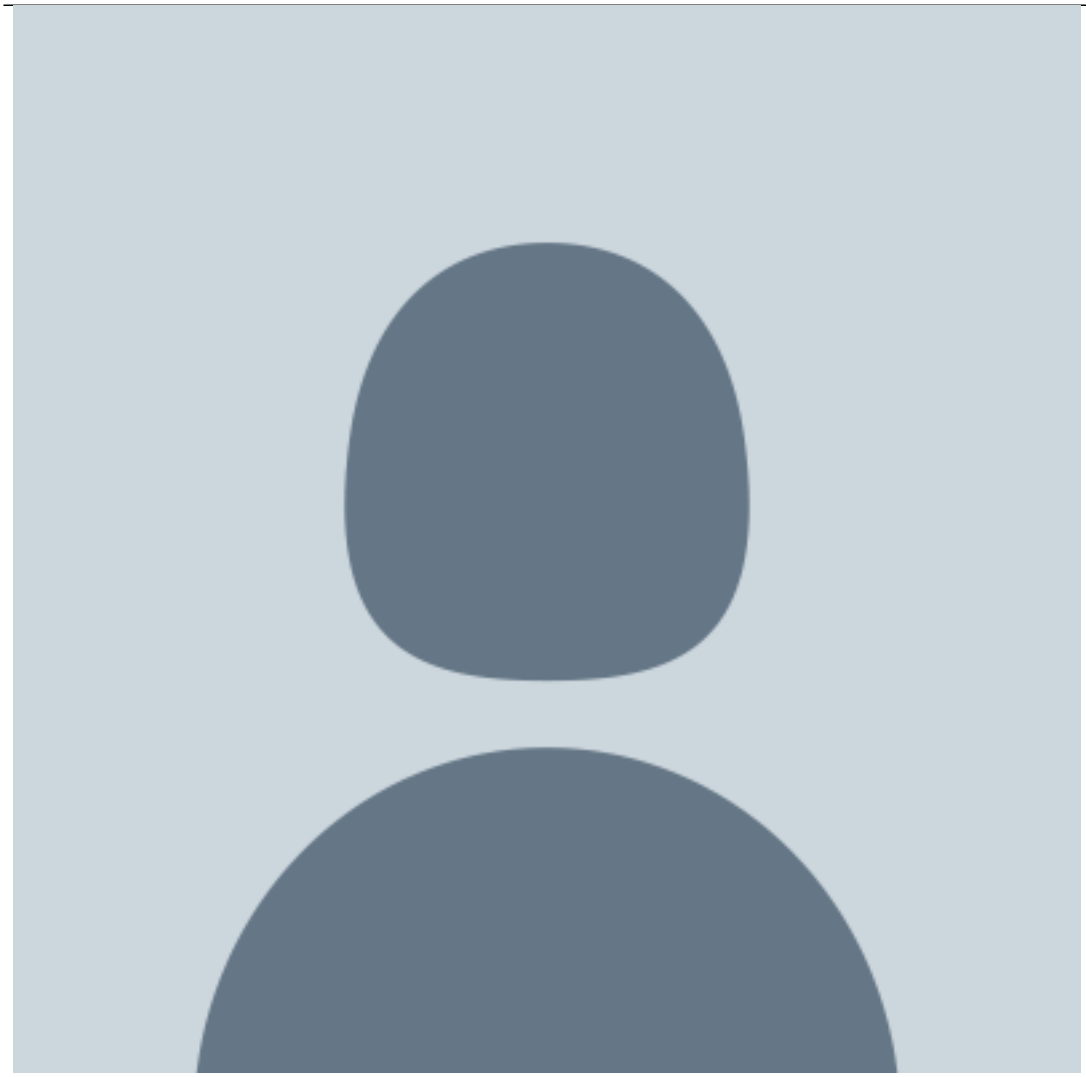


Senior Group IP Counsel

IBM

Lauren Bruzzone is responsible for IP matters for global business and transaction services, cloud, commerce, sales and distribution, and global finance business units. She has experience negotiating alliances, joint development agreements, and acquisitions.

Shiveh Roe

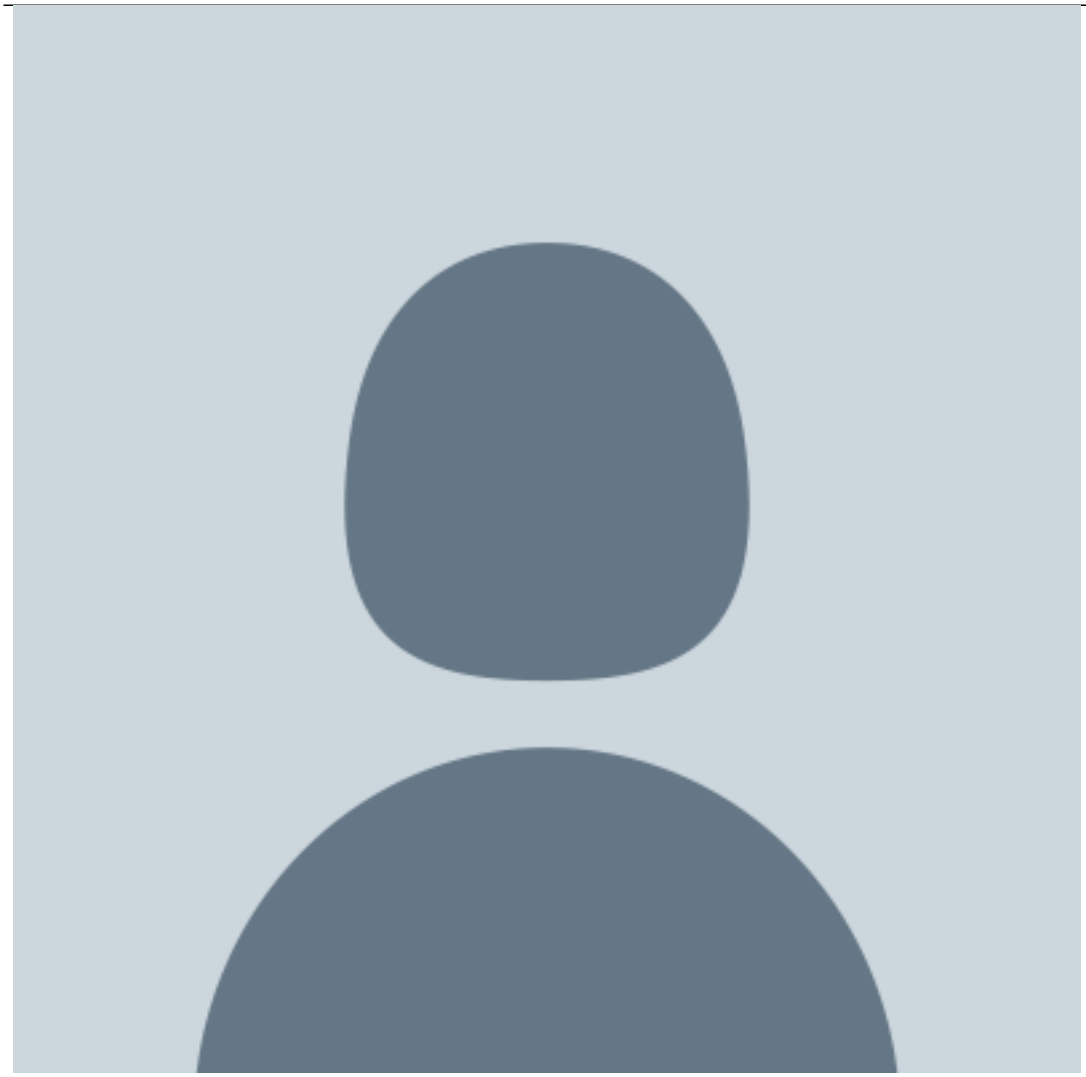


Associate

Kilpatrick Townsend

Shiveh Roe focuses her practice on trademark, copyright and domain name issues.

Joe Petersen



Partner

Kilpatrick Townsend

Joe Petersen focuses his practice on litigating copyright and trademark cases involving new technologies and business models.