Passport Without A Visa: Open Source Software Licensing and Trademarks

Tiki Dare[a], Harvey Anderson[b]

(a) Legal Director, Sun Microsystems, USA; (b) General Counsel, Mozilla, USA.

With an introduction by Amanda Brock[c]

(c) General Counsel, Canonical Group.

Abstract

Open source software has defied its sceptics and become a big business. Governments at the national, state and local level across the globe are requiring open source in their projects. Almost all major commercial software vendors use or distribute code under some open source license. As a user, it's hard to go a day on the web without interacting with some open source code which has replaced many server side legacy products. Worldwide more than 350 million consumers use open source software products and thousands of enterprises use open source code1.The best known open source brand is still probably the Linux operating system, but as open source projects and companies proliferate, the importance of brands to differentiate these offerings is on the rise. Trademarks, the legal rights that form the foundation for brand identity, will necessarily play a larger role in the open source world.

Keywords

Law; Free and Open Source Software; Trademarks

Info

This item is part of the **Articles** section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

Introduction by Amanda Brock

The following article, by Mozilla's Harvey Anderson and Tiki Dare, Sun's trade mark counsel, is a clear and accessible overview of the position of trade marks in FOSS. It was written from a US perspective, but the principles set out in the article apply equally in Europe, and the general legal position is similar.

In Europe there is an option to register either or both of a country specific or a community trade mark. For example, the editorial committee of this publication has recently applied for a community trade mark or CTM. This mark will give protection against infringement throughout all territories in the EEA. One downside is that in examining the mark application, the trade mark office (OHIM) in Alicante does not run checks on country specific pre-existing trade marks and so obtaining a CTM registration does not guarantee that there are no pre-existing marks registered in

a country's domestic trade mark registry. This means that during the process of registration, the mark will be registered even though there are conflicting national marks, unless the owner of the conflicting mark notices that an application has been made[1]. There is also the potential for owners of conflicting marks to challenge the mark in a territory for a period after registration, which is why applicants are generally advised to undertake a search themselves prior to applying for the mark.

A European (or domestic) mark can be the launchpad for an international registration under the Madrid Protocol, which allows a European or domestic application as the starting point for an application for marks in the other signatory countries. Marks are granted in specific classes, each class relating to a kind of usage, for example class 9 (which includes computers and related hardware, firmware and software) and class 42 (which includes computer-related related services), and are granted for renewable 10 year periods.

There is a degree of harmonisation of trade mark law in Europe. Although both registered and unregistered marks may be protected (unregistered marks to a lesser degree), harmonisation applies mainly to registered trade marks, with individual territories applying differing domestic law to unregistered marks.

Usage of trade marks must, as the article explains, be consistent - and this requirement exists in the UK and Europe as well as in the US. The rather amusing Penguin biscuit cases[2] were a great example of this. At the time, the Penguin chocolate biscuit was well-known in the UK. Each biscuit was packaged in a wrapper decorated with a lighthearted image of a penguin. When a supermarket brought out a copycat "Puffin" biscuit, the Penguin manufacturers felt that they had a case for trade mark infringement and passing off. However, Penguin's claim suffered when it was discovered that the marks which Penguin had registered, which included a number of illustrations for use on the biscuits, had not been used for some time and that the images which had been used were not in line with the registrations. The Penguin ultimately won its claim for unregistered trade mark infringement against the Puffin (but lost in its registered trade mark claim).

Anderson and Dare's article suggests a split in branding between enterprise and community versions of open source projects. However, marks in brands with a strong community contribution may not necessarily split in this way. Ubuntu, the operating system distributed by Canonical, does not have differentiated enterprise/community versions for important philosophical reasons. This stands in contrast to the dual-branded distributions, like Red Hat[3] and its community version Fedora[4]. This is easily resolved, in Canonical's case, by the brand allowing a non-commercial use of the protected mark by the community, and by making a clear distinction in the brand's trade mark policy between the freely licensed not-for-profit or non-commercial usage which is granted for free in a general licence to the community (subject to compliance of the non-commercial user with the trade mark policy) and commercial usage. Commercial usage may not only be subject to the rules of a trade mark policy, but also subject to the terms of a commercial licence available at the brand owner's discretion. In other words, there is no guarantee that the trade mark owner will grant a commercial licence to the trade mark.

For useful information on trade marks registrations in Europe or to check if a trade mark registration exists in Europe, see the OHIM web site[5].

---

1   Though commercial 'watch' services which will monitor potentially-conflicting marks do exist
2   United Biscuits (UK) Limited v Asda Stores Limited (Chancery Division, 18th March, 1997) Robert Walker J
3   See Red Hat, http://www.redhat.com/
4   See Fedora Project, http://fedoraproject.org/
5   http://oami.europa.eu/

✳

Introduction

Before we dive further into our topic, a few definitions and some historical background are in order. Although it comes from earlier roots, the free software movement got its start in the early 1980s. This movement had the goal of breaking the traditional business mould of proprietary (and often expensive) software. Modifying software requires access to its source code (as opposed to the non-modifiable, executable binary code, which is also sometimes called object code). Because proponents of free software developed licenses that would allow unrestricted sharing of the source code, the term "open source" was coined. Another term often used with certain kinds of open source licenses is "copyleft," which is a licensing concept developed by the Free Software Foundation in a popular open source license, the GNU Public License (GPL). It is defined on the GNU.org website: "Copyleft is a general method for making a program or other work free, and requiring all modified and extended versions of the program to be free as well. Copyleft says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it."

Today 'F/OSS' is an inclusive term generally synonymous with both free software and open source software which describe similar development models, but with slightly differing cultures and philosophies. 'Free Software' focuses on the philosophical freedoms it gives to users, and "open source" – a superset of Free Software – focuses on the perceived strengths of its peer-to-peer development model achieved by making the source code open to foster improvement, modification, and use. Although there may still be significant philosophical distinctions between the two views, 'F/OSS' can generally be used to refer to both and for the purposes of this paper we will refer to the Free Software and Open Source Software communities as 'F/OSS'.

Licensing plays a critical role in the open source community as it is the operative tool to convey rights and redistribution conditions. The F/OSS licenses have focused primarily on copyright and patent rights, which directly protect the underlying "bits" of code - the software itself. Addressing the third pillar of intellectual property rights – trademarks[6] – at all in F/OSS licenses is a relatively recent trend, and none of the open source licenses grant trademark rights. It was logical to start with copyright and patents, because trademark law protects the name and logos or other branding elements that are applied to the underlying code, but not the code itself. As open source software has become widely adopted among consumers and now generates significant revenues for some companies, the need to understand trademark law and to develop licensing and other conventions for managing trademarks is increasingly evident in the community.

At the time of this writing, a non-profit group called the Open Source Initiative (OSI) lists 65 active open source licenses that it has approved. OSI approval is one pathway to acceptance of a license, and the code distributed according to its terms, by a large proportion of the F/OSS community. Of these approved licenses, 19 are completely silent on trademarks. Another 19 prohibit use of names or trademarks in endorsements, advertising or publicity. Twenty six explicitly exclude a grant of trademark rights, and a few more prohibit specific uses of a name or mark. In addition to the license text, open source publishers commonly include statements separate from the license indicating that trademark rights are not provided. In some cases, developers may also include the trademark and logo files in different directories with alternative headers to convey that the open source license terms do not apply. All of these efforts are focused at excluding

---

6   The fourth pillar, trade secret protection, is largely irrelevant in open source development.

trademark usage by others. Only one of the open source licenses mentions that a trademark license is available separately from the project owner.

One of the most prominent and long-awaited recent developments in open source licensing was the publication of GPL version 3 in 2007. The GPLv3 license states:

> "Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms: ...Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks...."[7]

The GPLv3 reference to a clarifying statement on trademarks recognizes the growing importance of explicit trademark terms, and also reflects acceptance by a significant proportion of the community that trademarks are not an inextricably linked part of the software they have licensed.

Taking a step further, a recently OSI-approved license, the Common Public Attribution License, includes no trademark license or permission, but does require the licensee to acknowledge the trademark owner's rights:

> "You acknowledge that all trademarks, service marks and/or trade names contained within the Attribution Information distributed with the Covered Code are the exclusive property of their owners and may only be used with the permission of their owners, or under circumstances otherwise permitted by law or as expressly set out in this License."

Acknowledgment of ownership is nearly always included in a trademark license.

The trademark problem that arises in F/OSS is that anyone can modify, release, and distribute the code under the F/OSS license and, despite the exclusionary language in some licenses, there's an expectation that the project name – often the brand – can be used by the developers. What does this mean in terms of trademark law? Trademarks identify origin, and origin operates as a proxy for a level of quality that users expect. In this sense, quality could be excellent or poor, but consistent with the user's expectation. So in this context, can anyone modify the code and then use the trademark on the modified code? Is the source or origin of the code still the same? Is it licensed? How do developers show endorsement or relationship to the project? Ultimately, are consumers obtaining the protections and indicators of the source they deserve? This presents interesting challenges for trademarks and F/OSS projects. It also suggests that new notions of trademark law may be required to reflect dramatically different creation practices for goods that were not foreseen when the body of U.S. trademark law was last overhauled in 1946 by the Lanham Act.[8] Looking more closely at the legal structure around trademarks makes it clear why this is true.

Trademark 101

A trademark is most often a name or logo (but can also be a sound, color, smell, design or other device) that identifies the source of a product or service. You can immediately see the conundrum – what does it mean to identify source (for clarity, we'll say origin) of software in the open source world? The point of using an open source license is to allow the underlying software to be modified and redistributed, possibly through many generations of modification and distribution. If

---

7   The GNU Project, "The GNU General Public License" version 3, at s7 (e). Available at:
     http://www.gnu.org/licenses/gpl.html
8   See http://en.wikipedia.org/wiki/Lanham_Act#History

you start with a piece of software named Pure, after one modification, is the resulting software still Pure? More over, would users still consider it "Pure" - no doubt it would depend on the modifications.

Trademarks as IP differ from copyright and patent rights in important ways. Both copyright and patent provide the author/inventor with a monopoly (or right to exclude others in the case of patents) for a limited number of years, for the purpose of recouping the investment of time and resources in developing the work or invention. This policy is founded in Article III, Section 8 of the U.S. Constitution. By contrast, trademarks can last forever, as long as the relevant customers recognize the mark as identifying the source of products (including non-commercial items, like open source software) or services. Instead of repaying an inventor's "sweat of the brow," the principle underlying trademark law is consumer protection. The goal is to prevent customers from confusion about the origin of products or services.

By identifying source (or origin), trademarks tell you with whom you are dealing. They also symbolize specific qualities a product or service will have. Trademarks are the legal rights underlying brands, and these qualities form the user's brand experience. For example, when you see the name Cheerios, you (and toddlers and their parents all over the world) expect a particular oat-based, O-shaped cereal, with consistent flavor, freshness and crunch every time. You may or may not recall the name General Mills, the company that is the source of the cereal, but you know that one entity should be working to ensure you have a consistent brand experience. And if you don't get the flavor, freshness or O-shaped cereal you are expecting, General Mills will do something about it, possibly through coupons, a refund or a replacement box of cereal. That's the brand promise that General Mills makes to its customers and the trust relationship built between them, symbolized by the Cheerios name and logo.[9]

Source matters in the open source realm too. In open source products the unique features (both included and excluded), functionality, interfaces, security, architecture, and performance collectively create an identifiable user experience that consumers associate with a source or project. This expectation is the "quality" consumers use to inform their selection decisions and is of paramount importance. On a popular community website, Bill Burke, then JBoss' chief architect, listed trademarks as one of the most important considerations for any open-source business. Why? Because being the source of code arguably matters more than source code in an open-source business. The code is easily replicated, as it is open, but the trust associated with source (or origin) is not replicable. Trademarks are all about source. Who is the source of a given product or service? Even if the source isn"t known, a trademark represents to the user that the goods or services come from the same source, and in the case of F/OSS projects, a collection of like minded developers.

Trademark Licensing and Quality Control

The relationship between trademarks and quality is reflected in the law around trademark licensing. The trademark owner must maintain a meaningful opportunity to control the quality of any product bearing the trademark, even if the trademark owner licenses others to design, manufacture or modify the product and affix the trademark owner's brand. The legal test is not whether the quality is high or low – it is whether the trademark owner exercises control to ensure that quality is consistent.

---

9   Although Cheerios started as a US brand, it is now available worldwide. In China, General Mills distributes the cereal through a joint venture and we understand the formulation is sweeter than the US original to please local palates. Nevertheless, it is General Mills of Golden Valley Minnesota that is responsible for the quality worldwide.

A 2002 case involving California wine shows how the absence of control – called "naked licensing" – plays out in the US courts. Barcamerica Int'l USA Trust v. Tyfield Importers et al., 2002 WL 850825 (9th Cir. 2002). Barcamerica held a US registration for the mark DA VINCI for wine. An Italian wine maker, Cantine Da Vinci, sought to cancel Barcamerica's US mark, alleging that Barcamerica no longer used the DA VINCI mark. Barcamerica alleged that it was using the mark through its licensee, the Renaissance wine company. Under the terms of the license, Renaissance's use of the DA VINCI mark would "inure" to the benefit of Barcamerica - meaning that Renaissance's use would be legally equivalent to use by Barcamerica.

The court held the license was no longer valid, however, because Barcamerica was not exercising quality control. Two factors weighed against Barcamerica: (1) Barcamerica did not directly test or sample the wine; and (2) the individual winemaker employed by Renaissance to make the DA VINCI brand wine, Karl Werner, had passed away, so that Barcamerica could no longer rely on his skill and reputation to guarantee the consistent quality of the wine. As a result, the court found that the license was no longer valid, Renaissance's use of the DA VINCI mark could not be attributed to Barcamerica, and because Barcamerica was no longer using the mark itself or through a licensee under a valid license, it had abandoned its mark.

The Trademark Model for Standards Organizations Doesn't Fit Open Source Projects

Trademark law and licensing principles usually work well for standards bodies, such as The Open Group, which licenses the UNIX brand, or the Blu-Ray Disc Association, which licenses the Blu-Ray technology brand. Standards bodies form because some members of an industry perceive a common need to establish (1) a technology standard, (2) a means of communicating which products implement the technology (often through a trademark or logo), and (3) a set of tests or other criteria for determining whether the technology standard is implemented properly (quality control).

Most F/OSS projects do not follow the standards model, so trademark licensing requirements are not a natural fit. For example, many open source developers are interested in solving a particular problem - how can I create this functionality in a smaller footprint? How can I add better security? How can I make this more scalable? Choosing an open source license and administering a community take some time away from the development effort, but the resulting contributions and bug fixes make the software better faster, furthering the original goal. But that may not extend into the need to standardize. The project may fill a particular niche in its industry without the need to harmonize with parallel or complementary efforts. The developers may not want to spend time developing compliance tests, and there may be no market need for testing or a compliance brand. The developers of these open source projects may be content to give their software an "etymologically interesting" name.[10] The requirements of trademark law may feel bolted on and unnecessary.

On the surface, it may seem that the very nature of a F/OSS project precludes actual control in the most literal sense. F/OSS projects do create voluntary and mutually agreed protocols for developing code released under their respective marks. These methods, although they do not come from a single point of control in the conventional trademark law sense, do constitute quality

---

10   We are indebted to Simon Phipps, Chief Open Source Officer of Sun Microsystems, for this phrase and concept. A classic example of this is the name of the "Apache" open source project, which derives from an early description, "a patchy server."

control in the real world and could, at least in theory, be formalized into a trademark license requirement.

Another key reason traditional trademark licensing is not more widely practiced in the open source community in contrast to a standards body is the absence of a central authority. The organizational structure of a F/OSS project can fall anywhere on the spectrum from a completely informal arrangement among a few individuals to a publicly traded corporation, with most falling in between. It is common for larger, sustaining projects to have a formal charter and governing board. Smaller projects may not have a formalized legal entity (such as a corporation or foundation) that serves as the owner of the intellectual property in the project. This raises two challenges on the trademark front. First, most jurisdictions do not allow joint ownership of registered trademarks, and the legal owner of the mark must be the entity to license it. Second, an organizational control point or (distributed mechanism) is needed to establish the appropriate level of quality that is consistent with the core values and goals of the project, as well as appropriate quality control mechanisms.

Implied Licenses

One of the primal questions about trademarks in F/OSS projects is, absent a clause excluding a trademark grant, "Do the open source software licenses imply a trademark license?"[11] A concern underlying this question is whether the hypothetically implied trademark license would be viewed as a "naked license" that would in turn cause the owner to lose its rights in the mark. Given the large proportion of OSI-approved licenses that are either silent on trademarks, or prohibit only endorsement, advertising or other specific behaviors, and the number of software offerings that may be distributed under these licenses, the impact of an implied license would be far-reaching. With the caution that this has not been tested by the courts, the answer should be a clear "no". First, open source licenses are source code licenses that permit copying, modification and redistribution of the software code based on copyright law. Some also grant a license to or promise non-assertion of patent rights, and some are "copyleft". All of these features relate to the work itself. Although rights to the use, modification and redistribution of the code are granted under the F/OSS licenses, trademark rights are not provided inherently and often are expressly excluded as a point of clarification.

The trademark problem that arises in the F/OSS setting is that anyone can modify, release, and distribute the code under the F/OSS license. Thus, the origin of the modified code is no longer consistent or known. Consequently, implying a trademark license to a work that is modified by someone other than the original developers does not make sense.

Implying a trademark license would also conflict with the main purpose of trademark law: consumer protection. In open source development, the customer could be a developer who plans to make more modifications or an end user that will deploy the software. Both need an easy way to distinguish whether the software is coming from the original contributors or if it has been modified by someone else.

The US courts have generally resisted opportunities to imply a trademark license. They will look for proof that permission was given to use the mark and for an exercise of reasonable quality control.[12] A trademark owner, absent a licensing arrangement, would rarely have any opportunity

---

11   The courts have also had few opportunities to review open source software licenses. Recently in the U.S., the August 2008 Jacobsen v. Katzer decision by the Court of Appeals for the Federal Circuit upheld enforcement of the copyright restrictions of an open source license.
12   Trademark Licensing, Neil J. Wilkof & Daniel Burkitt, Section 11-23 p. 252.

to strictly control the quality of software being modified and distributed under an open source license.

Passport without a Visa - Challenges in the Community Context

An open source license is like a passport without a visa. The software can move freely under the copyright license to the source code (the passport), but the trademarks are subject to more limitations and may not be able to cross some borders without additional licensing (the visa). The analogy is apt in many ways. To an ardent traveler, the need for a visa is an inconvenient restriction and unwelcome formality. Although the internet (fueled in part by open source software) has made the world smaller, trademark law is still firmly rooted in a territorial past.[13]

As an attorney counseling on open source software issues, in addition to knowing these basics of trademark law and how these legal principles apply in the open source realm, you will also want to understand the open source community your client company may be engaging, and some of the concerns the community may have about trademarks and licensing.

First is a lack of clarity about when a trademark license is needed, and what use of the trademark is permissible without one. A license is needed for use of a trademark in a company name, product name or service name, or whenever a name or logo[14] is "affixed" to a product or service. In practice, this means that in the absence of a trademark license, a source code licensee should not use the name of the source code base as the name of her or his own software distributions, or use any logos associated with the source code F/OSS project teams may inadvertently create some confusion in their community by including the name and especially splash-screen or logo files in the code base they make available under a source license.

This is how that potential confusion may play out. When multiple developers legitimately exercise their rights under the F/OSS license, each making their own changes to the code base as they deem appropriate, and each using the same project name for the release, what then does the name convey to the user? It may certainly convey that the code derives from a collection of contributions that form a project, however, any of those developers may have included code and/or features that are inconsistent with the project values, code that introduces security vulnerabilities, or that adopts an architecture that conflicts with the norms of the project. In this case, use of the project name ceases to operate as a trademark because the consumer no longer knows the derivation of the product, nor can the consumer reasonably expect the brand to continue to operate as a symbol of consistent quality because the trust is gone.

A second concern is quality control. The quality control requirement is one of the most likely to create disagreement because it is has no parallel in the more familiar copyright or patent law. Mozilla and Debian disagreed over what code would be included in the "Firefox" branded web browser (in trademark terms the quality control standard). Debian wanted to make unrestricted modifications to the Firefox code base and then continue to distribute under the Firefox name and logo. Mozilla objected to unrestricted modifications of the code without community review in the branded release. Debian ultimately distributed a new browser based on the Firefox code base, but

---

13 Approximately 184 different jurisdictions offer trademark registration, and with a few exceptions, including the U.S., where rights are based on first use rather than first registration, rights are only protected in countries where a mark is registered.

14 Under US law, any use of a logo requires some form of a license or permission, while the European Directive offers more latitude.

branded it "Iceweasel".[15] This is often called forking[16] which results in two similar but distinct code bases.   The lack of compatibility between the two code bases can create real problems.

Because of the quality control requirement, and because F/OSS licenses permit free modification of the underlying software, trademarks are an important tool for guaranteeing compatibility.

Conversely, developers may modify the project code base in ways completely consistent with both the project values and the developer"s use case. Here, use of the brand conceptually should work as a trademark because it fundamentally serves the purpose of trademark law – consumer protection – but under conventional trademark jurisprudence it may fail the test for quality control.

Newcomer status for trademarks is a third major hurdle in protecting trademarks in the F/OSS arena. The Software Freedom Law Center[17] (SFLC) and other non-profit groups counsel in this area, but independent resources like these are few and not widely known.

Many F/OSS community members have copyright and patent licensing expertise (despite the ubiquitous disclaimer, "IANAL" for I Am Not A Lawyer in their communications), but trademark is often outside the comfort zone. Moreover, the contours of the intersection of trademark and copyright are not well defined even for skilled practitioners.

Fourth, community members or open source governing board members may feel a strong affinity for, or even outright legal ownership of, a brand that may be legally owned by another entity involved in the project. A range of traditional and creative approaches can be used to manage disagreements about ownership and licensing, but you may want to consider developing a trademark or logo for community usage, to provide an outlet for community sentiment.

A fifth challenge is the F/OSS culture of transparency. Our legal training and experience often predispose us to prefer confidential settings. Just as the development of the F/OSS software is public and collaborative, most F/OSS communities will have a strong preference to conduct traditionally confidential discussions about proposed trademark licensing terms and opportunities and appropriate quality control (and also trademark enforcement) in an "open kimono," public forum.

None of these challenges is insurmountable, nor is the lack of well-established F/OSS trademark licensing term and conditions. More members of the F/OSS community are actively discussing and working through trademark issues than ever before, and the collaborative nature of the community helps new knowledge and best practices travel quickly and globally.

Best Practices

The following best practices can address some of the challenges presented by trademark and open source development:

- Adopt both a brand name for official releases from the project and a separate community project name that may shows affiliation, but not constitute use of the original trademark.

---

15 As of December 18, 2008, the Wikipedia article is generally accurate, see http://en.wikipedia.org/wiki/Iceweasel

16 Forking occurs when a developer or group takes the source code of a project and starts a parallel, independent development project (of course, modifying the code is permitted under the open source license). If the second project, the fork, does not adopt a new name, forking can cause significant confusion about the origins of the respective projects.

17 SFLC offers a legal issues primer for open source projects on its website, including a section on trademarks.

Examples include Red Hat & Fedora, Google Chrome and Chromium.

- In your software distributions and repositories, consider distributing logos, and possibly other design elements, splash screens and icons, if any, in one or more separate files. (The licensing terms may need to be different.) This is easier and more efficient than including them and requiring your downstream developers to spot potential problems and strip them out later.
- Consider adding a section to any open source license identifying your trademarks and stating that no license is granted.
- If a new license section isn't appropriate, consider adding a trademark notice such as the OpenJDK Trademark Notice.
- Publish a set of trademark guidelines on your community site. The Software Freedom Law Center includes a proposed set in its Legal Primer.
- Consider logo programs that your community can use with minimal administration e.g., OpenSolaris fan buttons; or consider a mascot or logo that is designed for virtually unlimited community use (Sun open-sourced Duke using the BSD license).
- Consider the most transparent options for communicating with your community members about trademark use and misuse; involve community advocates and evangelists rather than using legal alone.
- We always recommend checking with independent counsel to make sure any trademarks, logos and other elements are legally available for your proposed use.

Conclusion

Because of their powerful role as source identifiers and their ability to guarantee compatibility, trademarks are a valuable asset for open source communities. The community is moving, albeit slowly, in a direction that we hope will lead to the successful development of standard trademark licensing terms and conditions, and possibly even an OSI-approved trademark license. A standard license along with more informational resources around trademarks would have clear benefits to all of the diverse stakeholders in the open source community.

About the authors

Tiki Dare is the Director of Trademarks & Marketing at Sun Microsystems, Inc. She is responsible for protection of Sun's trademark portfolio worldwide and she supports Sun's corporate and brand marketing. Sun's core brands include the Java technology platform, the Solaris Operating System, the MySQL database management system, Sun StorageTek storage solutions and the UltraSPARC processor. Her expertise includes advertising, intellectual property and competition law. Tiki has been with Sun since September 1997. She holds a B.A. from Dartmouth College and a J.D. and M.A. in the Humanities from Duke University. She is currently a member of the Board of Directors on the International Trademark Association.

Harvey Anderson has counseled internet, technology, and consumer goods clients in complex commercial transactions, as well as financial, corporate, litigation, and intellectual property matters for the past 16 years. He is currently Vice President Business Affairs and General Counsel of the Mozilla Corporation, distributor of the Firefox browser used by 200M people worldwide. He has led M&A activities for public and private companies, successfully directed significant patent litigation matters, set industry wide licensing standards, and created valuable IP portfolios. Prior to Mozilla, he served as SVP Corporate Affairs and General Counsel for Seven Networks, a

software company providing white label mobile email solutions to wireless carriers worldwide. Previously he was COO and General Counsel for Flywheel Communications, Inc. and Medscape (MSCP). At Medscape, a provider of electronic medical records, he led the company"s filing and registration activities resulting in a successful IPO and the launch of one of the first consumer personal health records. Prior to Medscape, he served as Assistant General Counsel for Netscape Communications Corp (NSCP). At Seven Networks, he managed the company"s legal and corporate affairs activities, including corporate and commercial transactions, licensing, finance, and patent litigation activities. As Assistant General Counsel of Netscape Communications, he directed technology licensing, intellectual property development and litigation, and other complex sales and corporate transactions. Prior to Netscape, his practice at McCutchen Doyle Brown & Enersen and Limbic & Limbach focused on intellectual property litigation. He has a J.D. from the University of San Francisco, a B.S. in engineering from Marquette University, and is a member of the United States patent bar.