



2-21-2017

The New Foundations of Open Source

Heather Meeker

Stephanie Petit

Follow this and additional works at: <http://digitalcommons.law.scu.edu/chtlj>



Part of the [Intellectual Property Law Commons](#), and the [Science and Technology Law Commons](#)

Recommended Citation

Heather Meeker and Stephanie Petit, *The New Foundations of Open Source*, 33 SANTA CLARA HIGH TECH. L.J. 415 (2016).
Available at: <http://digitalcommons.law.scu.edu/chtlj/vol33/iss3/2>

This Article is brought to you for free and open access by the Journals at Santa Clara Law Digital Commons. It has been accepted for inclusion in Santa Clara High Technology Law Journal by an authorized editor of Santa Clara Law Digital Commons. For more information, please contact sculawlibrarian@gmail.com.

THE NEW FOUNDATIONS OF OPEN SOURCE

Heather Meeker[†] and *Stephanie Petit*[†]

In recent years, the news about open source software development has been full of new community players: OpenStack Foundation, CloudFoundry.org Foundation, the Document Foundation, and the Open Source Robotics Foundation. Each of these projects is run by its own corporate entity, created and funded by a handful of major technology company promoters. What are these new foundations of open source? How do they help potential competitors come together to collaborate on open source development? And why set one up, when plenty of open source projects are run informally?

I. THE OPEN SOURCE LIFECYCLE

Most companies go through several stages of open source involvement. They begin by using other people's open source code. At this stage, companies need to understand how to comply with the open source licenses that apply to the software they are using, and implement internal controls to be sure they keep track of components they are using and remain compliant with the licenses that cover them.¹ At the

[†] Heather Meeker is a lawyer in private practice who helps her clients understand and benefit from open source software licensing. She is a partner in the Mergers and Acquisitions group at O'Melveny & Myers in Silicon Valley and has over twenty years' experience advising on intellectual property transactions. Daily Journal named her among its Top 100 Women Lawyers in California in 2013, and Managing IP listed her as one of its IP Stars-Top 250 Women in IP in 2013. She was named San Francisco Information Technology Law Lawyer of the Year by Best Lawyers in 2012. She was given the prestigious Vanguard Award for private practice by the Intellectual Property Section of the California State Bar for 2016. Her book *Open Source for Business* is a handbook for businesspeople, engineers and lawyers who want to understand open source licensing in business. She is also the author of the ABA's SciTech Section publication *Technology Licensing: A Practitioner's Guide*, a widely-used textbook on technology licensing.

[†] Stephanie Petit is a tax-exempt lawyer who works exclusively with foundations, charities, their donors, and other nonprofits. She is a partner at Adler & Colvin, a San Francisco firm that focuses on the law of nonprofit and tax-exempt organizations. She is the exempt organizations editor of *Journal of Taxation*, an author of the chapter on obtaining and maintaining tax-exempt status in the CEB book *Advising California Nonprofit Corporations*, and the founding editor of Adler & Colvin's blog, nonprofitlawmatters.com.

1. See generally *Open Compliance Program*, LINUX FOUND. (2016), <http://bit.do/LinuxOpenSourceCompliance> (outlining the Linux Foundation's Open Source Compliance Program).

next stage, companies begin to contribute back to existing open source projects—a bug fix here and a new feature there. They do this in order to improve the code, and sometimes, to influence the direction of its development. At this stage, companies need to understand how to manage their intellectual property and set policies for employees to spend company or personal time on open source projects.²

In the last stage, a company takes a leadership role in developing open source projects. This may include releasing existing software under open source licenses, starting new open source projects, funding community development, or coordinating and managing development by others. Many companies never reach this stage, but if they do, they must make strategic decisions about the relationship between the open source project and the money-making activities of the company. This article discusses one of those decisions: whether to create a new entity to run an open source project.

Entities set up to run open source projects are often called “foundations”—such as the Free Software Foundation,³ the Linux Foundation,⁴ or the Mozilla Foundation.⁵ The popularity of creating open source foundations has waxed and waned over the last decades, more or less with economic cycles. Many were started in the early 2000s; the 2010s have also seen a crop of newly-created foundations.

II. INSIDE OR OUTSIDE

Many of the reasons for setting up a foundation are optical rather than actual. But optics are important, particularly in the open source world, which thrives on the delicate balance of many individual or corporate developers donating their time to a collaborative project. A company setting up—let’s call it promoting—a foundation wants to send the message that the project will serve the community instead of the promoter. Setting up a foundation also helps the promoter to separate its assets and resources—intellectual property and otherwise—from its profit-making business.

2. See generally HEATHER MEEKER, *OPEN (SOURCE) FOR BUSINESS: A PRACTICAL GUIDE TO OPEN SOURCE SOFTWARE LICENSING* (2015) (detailing open source code releases and contributions).

3. The Free Software Foundation, the original architect of the GNU project, has “a worldwide mission to promote computer user freedom and to defend the rights of all free software users.” See FREE SOFTWARE FOUND. (2017), <http://www.fsf.org>.

4. See *The Future is Open*, LINUX FOUND. (2013), <http://bit.do/FutureIsOpen>. The Linux Foundation is responsible for coordinating community development of the Linux kernel. *The Linux Kernel Organization*, LINUX KERNEL ARCHIVES (2016), <http://bit.do/LinuxKernelOrg>.

5. MOZILLA (2017), <http://bit.do/MozillaMission> (“Our mission is to ensure the Internet is a global public resource, open and accessible to all. An Internet that truly puts people first, where individuals can shape their own experience and are empowered, safe and independent.”).

Most foundations are set up as nonprofit, non-stock entities. They are run by a board of directors, which is either self-perpetuating or elected by members, who have many rights like those of stockholders, but not ownership. A nonprofit open source entity therefore is not a subsidiary of the promoter entity because the promoter entity neither owns the nonprofit (no one can) nor retains control over selection of its board (often necessary to demonstrate community involvement).

Founders frequently select Delaware to incorporate, perhaps for its familiarity and because of the assumption that Delaware favors nonprofits as it does for-profits. Much of this is true. Delaware is very flexible, allowing a nonprofit corporation wide latitude in the structure of its Board, committees, and membership.⁶ Delaware affords directors of its corporations more protection through its gross negligence (rather than simple negligence) standard for liability.⁷ Delaware does not require that a nonprofit apply for state tax-exempt status or file annual state information returns. If the foundation will be exempt under Section 501(c)(3) or hold assets for charitable purposes, Delaware's Attorney General does not require regular reports of charities or on the use of charitable assets, nor is it known for rigorous supervision of charitable trust assets.⁸ Where a new foundation is not initially certain of the tax-exempt status it seeks, Delaware is an excellent choice, because the initial certificate of incorporation does not require choices that preclude a particular tax-exempt status.

Others view Delaware with less enthusiasm, regarding its lack of reporting and enforcement as a lack of accountability, a core value of the nonprofit sector, and its more forgiving liability standard for directors and management as not-clearly-desirable in the nonprofit (and particularly the charitable) context, where the purpose is not profit but social good. Practically, Delaware's corporate law was fundamentally designed to address for-profits with stockholders. A Delaware nonprofit nonstock corporation must fit into a law originally written for corporations that have owners and very different goals from its own. The Delaware legislature attempted to fix this problem in 2010.⁹ The resulting amendments require anyone grappling with a Delaware nonprofit to analyze first whether any particular provision applies to a nonprofit at all, and if so, how. In short, Delaware has many benefits but can also be a clumsy fit for some nonprofits.

6. See Del. Code tit. 8, §§ 114, 141.

7. See Del. Code tit. 8, § 141.

8. PAMELA A. MANN, *THE ATTORNEY GENERAL'S REGULATORY AUTHORITY IN NONPROFIT GOVERNANCE AND MANAGEMENT* 396 (Victor Futter ed., 2002).

9. See 2010 AMENDMENTS TO DELAWARE CORPORATE LAWS, DEL. DIV. OF CORP (2010), <http://bit.do/2010DelAmendments>.

By contrast, California has a clear, well-developed corporations code specific to nonprofits, with no for-profit aspects at all, making it much easier for nonprofits to use. In California, a new open source foundation must incorporate as either public benefit or mutual benefit corporation, each of which is exactly what it sounds like. Public benefit corporations must benefit the public. If they desire to be tax-exempt, they would apply under § 501(c)(3) or § 501(c)(4). California's Attorney General has robust authority (which it exercises) to ensure assets of a public benefit corporation are properly spent.¹⁰ (This is a benefit or drawback, depending on your point of view.) Mutual benefit corporations are formed to benefit their members rather than the public. For tax-exemption, a mutual benefit open source foundation could claim trade association status and apply under § 501(c)(6).¹¹ Whether a new open source foundation would choose public benefit or mutual benefit corporate status depends on the goals and operations of the foundation and the tax status it wishes to have. Key drawbacks to California—again, depending on your point of view—are its extremely technical rules regarding member rights, meetings and actions, robust Attorney General oversight, and, of course, the possible liability of its directors for simple, rather than gross, negligence.¹²

After selecting the state of incorporation, founders must prepare and file articles or a certificate of incorporation, design a governance structure to assure appropriate community inclusion and facilitate development of the software (which might include members who elect the board, or support the organization through dues, and special relationships allowing individuals or companies to fill seats on the board), prepare bylaws embodying the governance structure, and appoint an initial board. They often also create intellectual property policies, technical governance policies, antitrust policies, or other corporate governance rules. Often, they apply for tax-exempt status, discussed below. The new foundation can then proceed to conduct the activities for which it was formed.

III. THE PATENT TIGHTROPE

Companies new to open source development are particularly anxious about how the development and release of open source code will affect their patent portfolios. Some open source licenses require

10. See CALIFORNIA ATTORNEY GENERAL'S GUIDE FOR CHARITIES, STATE OF CAL. DEP'T OF JUSTICE OFFICE OF THE ATTORNEY GEN. 34-40 (2005).

11. See *Types of Organizations Exempt under Section 501(c)(6)*, INTERNAL REVENUE SERV. (Dec. 21, 2016), <http://bit.do/501c6ExemptOrgs>.

12. See, e.g., Cal. Corp. Code §§ 5510-17, 5231.

the licensor to grant express or implied patent licenses on behalf of the licensor and all its corporate affiliates. If the foundation is a non-stock entity, and the promoter does not effectively control it (such as by occupying all the board seats), then release of software developed and owned by the foundation will not encumber the patents of the promoter.

Some open source advocates call this tactic “patent laundering” and view it with disdain. They suspect that the promoter is resorting to trickery to introduce patent “landmines” into the code, and later sue users of the code for patent infringement. But that kind of threat is mostly theoretical; few patent holders who distribute open source code intend to enforce their patents against recipients of the code. Patent laundering—or to use a less pejorative word, *strategy*—is less about laying a trap for users than to reduce the amount of internal patent housekeeping of the company releasing code.

This housekeeping is necessary because of the tension between open source development and stewardship of a company’s patent portfolio. The job of the portfolio managers—usually the patent lawyers of the company—is to maximize the value of the portfolio. For some companies, the patent portfolio has a very high value, and the duty to conserve it is a key fiduciary responsibility to the company’s shareholders. But licensing a patent royalty-free under an open source license can significantly compromise a patent’s value. Keep in mind that a patent is only the ability to prevent others from practicing an invention. A company that makes a business decision to release open source code would not have much interest in stopping others from using the code it is releasing. But any grant of a royalty-free patent license can reduce the damages available for other infringements of the patent.¹³ Patent infringement damages are measured, in part, by what would be a market-driven license fee for a license of the patent. A shrewd defendant will simply use the royalty-free patent grant in the open source license as evidence that the economic value of practicing the invention is zero, and argue that even if the patent is infringed, there should be no damages.

Therefore, typically, if a company wants to contribute code to an open source project, or release code under an open source license,

13. One measure of damages for patent infringement is a reasonable royalty for a license to the patents. *Georgia-Pacific Corporation v. U.S. Plywood Corporation*, 318 F. Supp. 1116, 1121 (S.D.N.Y. 1970) describes the most common approach for determining reasonable royalty damages—a hypothetical license negotiation between a “willing licensor” and “willing licensee.” Those patent owners who have only licensed patents royalty-free—either via open source licenses or otherwise—have created a playbook for a defendant to argue that a reasonable royalty is zero. This argument may not win, but raises the cost of enforcement of the patent by complicating the damages discussion.

company IP policy will require those requesting the release to identify the company patents that would be licensed as a result. Approval of the contribution or release usually depends on the patent portfolio managers signing off on this decision, determining that the release will not devalue the company's patent assets. In a small startup, this process may be easy; in a large, or even mid-sized company, this process can be quite difficult. In companies with large portfolios, particularly those that have grown by acquisition, a single person may not be familiar with all the patents the company owns. Making this decision can therefore become a bottleneck that slows down the release process.

Engineers who become frustrated with barriers to making open source contributions may disregard the policy, or worse, simply leave the company. Today, the opportunity to contribute to open source projects is a key recruiting lure for the best engineers, so a bogged-down legal process for code releases can become a serious business problem. The portfolio managers also have their own travails. Their job is to create and maintain valuable patents. A patent that is encumbered by an open source license is not valueless; it can still be used to prevent uses other than the open source code, and the company may be able to argue successfully against defenses that a reasonable royalty is zero. But most companies making an open source code release do not have the luxury of prosecuting patents whose value is inherently limited. Prosecuting patents is expensive and consumes company resources. If a company is building a patent portfolio, its resources are better devoted to patents that have not been licensed to the world free of charge.

Setting up a foundation goes a long way toward breaking up this logjam. If the portfolio managers are comfortable that development and release of open source code via the foundation will not compromise the company's patents, they need not be gatekeepers of open source code releases.

IV. TRADEMARKS AND BRANDING

Another key strategic reason to create a foundation is trademark management. Trademark management in the open source landscape is a challenge. Trademark law and open source licenses are fundamentally at odds: the open source license grants broad freedoms to change software products, and traditional trademark management requires that they not be changed at all.¹⁴

14. See Pamela S. Chestek, *The Uneasy Role of Trademarks in Free and Open Source Software: You Can Share My Code, But You Can't Share My Brand*, 102 J. INT'L PROP. L. & PRAC. 126 (2012).

A trademark represents the quality control of a particular source. A robust open source project is developed by an entire community. The source of a promoter product is the promoter entity, and the source of the open source software product is the community. Some companies approach this concern by creating a combination mark, such as when a company selling FOOBAR software creates an alternative called Open FOOBAR or FOOBAR Community Edition. But such variations on a promoter's existing trademark will only weaken the brand and the trademark rights associated with it. For the best brand management, it is better to create a new name for the project, control the quality and official versions of the project via the foundation, and manage the use of the trademark via the foundation.

Most open source projects of any meaningful size are run by a small number of senior engineers, who have the last word on what contributions make their way into the official software release.¹⁵ In other words, theoretically, anyone can contribute to the Linux kernel, but only a small set of respected and experienced engineers, trusted by the community, have the authority to commit contributions to the official kernel. Practically speaking, this means the trademark represents the quality control exercised by the project committers. When a foundation is set up by a promoter, the initial project leads are likely to be promoter engineers, but over time, other community members will be likely to become committers as well. This means that in a practical sense, the promoter entity will not be the arbiter of quality control in a robust open source project. The promoter entity, therefore, should not be using its own trademarks for the project.

The open source world has historically done a rather poor job of trademark management.¹⁶ That is a shame, because trademark law protects the community as much as it does the foundation. While open source projects don't like to be heavy-handed about trademark enforcement, and want to allow and promote the creation of user groups, developer groups, and viral promotions, some minimal level of trademark control is a part of good project stewardship. Otherwise, bad actors can hijack the project name. For instance, Mozilla Foundation, the developer of Firefox web browser, has leveraged trademark infringement claims to stop scammers from distributing copies of

15. There are many ways to run open source projects, but the gold standard is probably the Eclipse Foundation's "Eclipse Development Process." See *Eclipse Development Process*, ECLIPSE FOUND. (2015), <http://bit.do/EclipseDevProcess>.

16. For a discussion of these issues, see HEATHER MEEKER, *THE OPEN SOURCE ALTERNATIVE: UNDERSTANDING RISKS AND LEVERAGING OPPORTUNITIES* 111 (2008).

Firefox that contain spyware, phishing schemes, viruses, or other harmful technology.¹⁷

One of the key challenges of setting up an open source foundation can be the transfer of trademark rights from the promoter to the foundation. The promoter has often spent significant resources developing the software that will form the basis for the initial project of the foundation. Transferring control of the commercial rights to an associated product name is a big decision. Once the trademark has been transferred to the foundation, it is not feasible to give it back if the foundation fails.

V. OTHER BUSINESS BENEFITS

Intellectual property is not the only reason to create a separate entity for an open source project. Many projects are funded by community members—though these days, they are usually large corporate funders rather than smaller grass-roots contributors. Accepting funding from other companies to develop software in a private company can be awkward and bureaucratic. Today, particularly after the “Heartbleed” crisis for OpenSSL,¹⁸ the technology industry is increasingly skeptical of projects that run on a shoestring budget and/or without transparent community management. A separate entity can help organize and segregate the business resources for the project. Moreover, the management of a private company is anything but transparent. Private companies need to maintain aspects of their business as confidential, for the benefit of their shareholders. A separate entity helps avoid conflicts between promoter secrecy and community transparency.

Of course, there are drawbacks to creating foundations. It is always expensive (though not unduly so) to create entities. Filings must be made and bylaws must be written. Employees may be reluctant to work for foundations without benefits and stock options plans. Independent directors and officers may be hard to find. But overall, many companies today are finding that the benefits outweigh the costs.

17. Mozilla polices its trademark as described at *Protect the Fox (and More!)*, MOZILLA (2017), <http://bit.do/ProtectTheFox>. For an example of a violation, see *Is Your Firefox Genuine? Phishing at its Pinest!* 404 TECH SUPPORT (Jan. 30, 2010), <http://bit.do/IsYourFireFoxGenuine>.

18. Jeremy Kirk, *Critical OpenSSL 'Heartbleed' bug puts encrypted communications at risk*, PCWORLD (Apr. 8, 2014), <http://bit.do/CriticalOpenSSL>.

VI. TAX CONSIDERATIONS

The tax status of open source foundations can be a wild ride. Early open source foundations such as Free Software Foundation, Eclipse Foundation, and Mozilla Foundation were set up as charities under § 501(c)(3) of the Internal Revenue Code.¹⁹ The mission of a charity is to assist a charitable class (the sick, the poor, the elderly, and so on) or to benefit the community in specific ways, such as by advancing education or scientific research in the public interest. Among exempt organizations, 501(c)(3)s are special because contributions to them may be deductible by the donor as charitable. Until approximately the mid-2000s, the Internal Revenue Service approved some open source foundations as exempt under § 501(c)(3) reasonably quickly on the theory that they benefitted the community in fundamental ways. But over time, open source development has changed. These original foundations were mostly staffed by volunteers; over the years, corporate participation in open source development has burgeoned. Perhaps in conjunction with burgeoning corporate involvement, the IRS grew more skeptical of the charitability of open source foundations and more concerned about the levels of improper private benefit to founders and others that could be generated in the process of achieving charitable goals.

Today, open source foundations often struggle with obtaining 501(c)(3) status.²⁰ It was revealed that in 2010, the IRS had placed the term “open source” on its “Be on the lookout” or “BOLO” lists along with “Tea Party,” “Patriot,” and other terms the IRS viewed as red flags, in a scandal that ultimately led to Congressional investigations and the resignation of top IRS officials.²¹ Some open source organizations applying for exempt status tolerated waits of five, six, and even seven years along with numerous rounds of questions from IRS reviewers attempting to understand how open source worked and whether the proposed uses of it by a particular organization were

19. For a current list of open source foundations, including many 501(c)(3) organizations, see <http://flossfoundations.org/foundation-directory>.

20. See, e.g., Private Letter Rulings 201533014, 201505040

21. “These organizations are requesting either 501(c)(3) or 501(c)(6) exemption in order to collaboratively develop new software. The members of these organizations are usually the for-profit business or for-profit support technicians of the software.” Simon St. Laurent, *IRS wasn't fond of Open Source, either*, O'REILLY RADAR (June 24, 2013), <http://bit.do/IRSWasntFond>; see also Kelly Phillips Erb, *Not Just The Tea Party: IRS Targeted & Turned Down Tax Exempt Status Tied To Open Source Software*, FORBES (Jul. 17, 2014), <http://bit.do/NotJustTheTeaParty>.

charitable.²² Practically speaking, exempt status under § 501(c)(3) in recent years has required a clear charitable class application for the open source code, as well as measures to limit or preclude private benefit through commercial exploitation.

Other open source foundations seeking exempt status have turned to § 501(c)(6). § 501(c)(6) exempts trade associations directed to improving the business conditions of one or more lines of business. For open source foundations that fit this exempt status, § 501(c)(6) has proven successful.²³ Others have encountered long waits, many rounds of questions, and denials.²⁴ The IRS has found (among other adverse conclusions) that some would-be 501(c)(6) open source foundations are instead engaged in a for-profit business, that they offer too many services to members, and that they do not actually further a line of business.²⁵

Finally, some open source foundations have taken yet another route to exemption: § 501(c)(4), for social welfare organizations.²⁶ Roughly speaking, 501(c)(4)s must benefit the community, but the benefit need not be charitable in the classic feed-the-poor eleemosynary sense. (As one example, a 1962 ruling recognized the 501(c)(4) status of an organization that helped people in the community receive television reception.²⁷ And of course, more recently, 501(c)(4)s have become infamous for providing a vehicle for donors to anonymously channel “dark money” to influence elections.²⁸) While arguably more private benefit is permissible with 501(c)(4)

22. See, e.g., Robert MacMillan, Open Source Voting Machine Reborn After 6-Year War With IRS, *WIRED* (Aug. 6, 2013), <http://bit.do/OpenSourceVotingMachine> (discussing the long and difficult travails of the Open Source Digital Voting Foundation, which were ultimately resolved in favor of the Foundation).

23. The IRS recognized OpenStack Foundation, for example, as exempt under § 501(c)(6) (per [guidestar.org](http://www.guidestar.org)).

24. See, e.g., Private Letter Rulings 201420021, 201024066; also, the IRS initially determined that OpenStack Foundation was not exempt under § 501(c)(6). See Ernie Smith, *Open-Source Projects Failing to Pass IRS Nonprofit Muster*, ASSOCIATIONS NOW (July 22, 2014), <http://bit.do/OpenSourceProjectsFailing>.

25. See Smith, *supra* note 24.

26. 26 U.S.C. § 501(c)(4)(A) (“Civic leagues or organizations not organized for profit but operated exclusively for the promotion of social welfare, or local associations of employees, the membership of which is limited to the employees of a designated person or persons in a particular municipality, and the net earnings of which are devoted exclusively to charitable, educational, or recreational purposes.”); The Open Source Geospatial Foundation is one example, see OSGEO (2017), <http://bit.do/OSGeo>.

27. Rev. Rul. 62-167, 1962-2 C.B. 142 (1962).

28. See *Questions and Answers on 501(c) Organizations*, INTERNAL REVENUE SERV. (May 15, 2013), <http://bit.do/QA503c>; John Avlon & Michael Keller, *The Dark Money Shuffle*, DAILY BEAST (Sept. 19, 2012), <http://bit.do/DarkMoneyShuffle>.

organizations than with 501(c)(3)s, an open source foundation hoping for exemption as a social welfare organization will ultimately still have to show that it is organized and operated to benefit the community without excessive private benefit, the same roadblock which some would-be § 501(c)(3) open source foundations encounter.²⁹

Of course, some open source foundations may not fit well in any of these three tax-exempt frameworks. In the absence of reforming the tax code to specifically include open source development as an exempt activity, one possible answer to this conundrum is to incorporate as a nonprofit under state law, and dispense with the idea of tax-exempt status altogether. Even if member or supporter contributions constitute taxable income to the foundation, after expenditures, they rarely make any profit, which should result in little or no income tax most years. (Franchise taxes, typically much lower, may still apply.³⁰) Organizations who contribute are often indifferent about a foundation's tax status, as their dues are usually deductible as business expenses. Individual contributions to the foundation would probably not be deductible, but over time, such contributions have played a shrinking role in the open source world.

These foundations may now take the view that, whether they get tax-exempt status or not, they will nevertheless operate with the transparency and mission of a not-for-profit entity. In any case, the corporate governance of an open source foundation is a different animal from that of a business corporation. Boards of directors of open source foundations are typically selected based on community participation rather than ownership; it is often not possible to “buy” one’s way into control of a foundation, in sharp contrast to the ability

29. See, e.g., Private Letter Ruling 201507025. Unlike § 501(c)(3) organizations, neither § 501(c)(6) nor § 501(c)(4) organizations are *required* to apply for exempt status and may “self-declare” as exempt, provided they meet the requirements for exemption. IRS Service Center Advice 200046038 (“Because section 501(c) organizations other than those described in sections 501(c)(3), 501(c)(9), and 501(c)(17) are not required to obtain recognition from the Service of their tax-exempt status, these organizations qualify for exemption, if they meet the requirements of the Internal Revenue Code.”) In the past, such “self-declarations” were supported by filing Form 990, required of both § 501(c)(4) and § 501(c)(6) exempt organizations. Now, newly-enacted § 506 of the Internal Revenue Code requires new § 501(c)(4) organizations to make a very brief filing within 60 days of formation. See IRC § 506 and Notice 2016-9. In practice, however, organizations that have not applied for and received IRS recognition of exempt status cannot easily demonstrate their status. This is frequently an undesirable situation, but particularly so where the organization’s main activity does not inherently fit any particular exemption category and where the IRS’s skepticism of exemption is well-documented.

30. For example, in California, the minimum annual tax for a corporation is \$800. *What is the Minimum Franchise Tax?*, STATE OF CAL. FRANCHISE TAX BD. (2017), <http://bit.do/MinFranchiseTax>.

to purchase voting stock or interests in a business. In some open source foundations, board seats go to those who have contributed time, development resources, and even thought leadership to the community. So the teachings of the nonprofit world may still apply.

VII. WALKING THE WALK

The good news on all these topics is that the best way to manage an open source foundation—whether to manage a promoter’s intellectual property, engage in proper corporate governance, or seek special tax status—is to walk the walk. Most foundations start out being funded by one or a few players; anything else is a collective action problem that makes foundation set-up impossible. Even some of the most respected open source foundations today, like Eclipse and Mozilla, started with a single promoter willing to do a code release, and grew from there. The key to a successful foundation is true community involvement and transparency. With these, the foundation can achieve a true life of its own, and benefit the community or the industry, and perhaps incidentally, its promoter, from the outside.