Fordham Urban Law Journal

Volume 39 | Number 4 Article 2

February 2016

Free and Open Source Software in Municipal Procurement:The Challenges and Benefits of Cooperation

Justin C. Colannino

FREE AND OPEN SOURCE SOFTWARE IN MUNICIPAL PROCUREMENT: THE CHALLENGES AND BENEFITS OF COOPERATION

Justin C. Colannino[1]

ABSTRACT

The use of free and open source software by municipal governments is the exception rather than the rule. This is due to a variety of factors, including a failure of many municipal procurement policies to take into account the benefits of free software, free software vendors second-to-market status, and a lack of established free and open source software vendors in niche markets. With feasible policy shifts to improve city operations, including building upon open standards and engaging with free software communities, municipalities may be able to better leverage free and open source software to realize fully the advantages that stem from open software development.

TABLE OF CONTENTS

1. Justin C. Colannino graduated from Columbia Law School in 2010 and worked at the Software Freedom Law Center from 2010 to 2012. He holds a Master of Science in Computer Science from McGill University, and is now in private practice. The author would like to thank Aaron Williamson of the Software Freedom Law Center for his insightful comments and suggestions.

INTRODUCTION

Since its inception in the early 1980s, free and open source software[2] (free software), software released under a copyright license that permits the general public to study, use, copy, distribute, and prepare or distribute derivative works of the software, has demonstrated the promise and power of internet collaboration to produce no-cost digital tools for general public use and education. The coincidence of free software's growth and the development of other collaborative no-cost digital goods such as Wikipedia with the internet age is no accident. Whether it is a tweak that makes grandma's lasagna recipe even better, an improved work-out routine, or a new trick to getting the last bit of ketchup out of the bottle, it is human nature to share life improvements with others. Before the internet, these improvements either traveled by word of mouth or had to be published and distributed at significant cost. With the internet, each time someone accesses the information, she has a copy she can keep for herself and, once an improvement is written up, it can be shared and copied at no cost. Just put it online.

2. Over the years there has been significant debate in the community regarding how to refer to such software. *See, e.g.*, Joe Barr, *Live and Let License*, IT WORLD (May 22, 2001, 1:07 PM), http://www.itworld.com/LWD010523vcontrol4; Richard Stallman, *Why Open Source Misses the Point of Free Software*, GNU OPERATING SYS., http://www.gnu.org/philosophy/open-source-misses-the-point.html (last visited Apr. 2, 2012).

Take, for example, the now antiquated process of digitizing, or "ripping," a compact disc (CD). Typically, a computer digitizes an album on a CD by creating a different file for every track. At first, naming these tracks took time: for every track you had to type in the song title, the artist name, the album, and other information so that you or your digital music player could remember which file was which. This tedious task changed after the creation of the compact disk database (CDDB), an internet service that permitted users to query a database using a digital fingerprint of the compact disk that would return with song title and artist name for each track. If the fingerprint query turned up no results in the database, the user would need to enter in the information by hand, and CDDB would query users to upload the entered information to its servers. This query effectively enabled users to share the song title and artist name with every other person ripping the same disk.[3] Every person not only entered data to improve their own life, but, by sharing copies of the information through the internet, they were able to improve everyone else's life who was in the same situation.

Software, like Grandma's lasagna recipe or the CDDB database, can be represented digitally and so may be duplicated quickly, efficiently, and inexpensively over the internet. This makes collective development easy: whenever someone makes an improvement, he can share it online easily for all to use. These mechanisms in software development along with people and organizations making improvements in the programs they use and sharing those improvements with others have created a rich pool of stable software that can be used and improved upon by municipal governments to suit their needs.

Free software possesses three properties that enable this collective development that proprietary[4] solutions do not: its source code[5] is au-

3. Jason Fry, *Three Veterans Advise the Next Tech Wave: It's All About Business*, WALL ST. J. (Dec. 31, 2001), http://www.ibiblio.org/tkan/software/cddb_wsj_12.31.01.pdf; *see also iTunes 4: How to Send CD Information to the CDDB*, APPLE INC. (Sept. 18, 2003), http://docs.info.apple.com/article.html?artnum=93094 (explaining how to use iTunes 4 to submit CD information to CDDB).

4. This Article uses the term "proprietary" in the context of software and licensing to refer to the model of software production where the licensee receives money in return for a license for the single copy of the software, and the licensee reserves all or most of the rights granted by 17 U.S.C § 106 (2002).

5. A recurring concept in free software is the difference between "source code" and "object code." Software is different than many other copyrighted works in that the end-product run by the computer, the computer-executable "object code," is unintelligible to most human users without significant effort. Therefore, in order for a person to take advantage of the license terms permitting modification, he must be

ditable, its source code is modifiable, and it has a license cost of zero. By embracing this collective production method, municipalities are increasingly able to develop collective solutions to their common problems: when each municipality designs and builds the software it most needs, the collective wealth of all similarly situated municipalities is increased.

This Article examines this benefit and other related benefits derived from city governments' use of free software, and discusses current obstacles to free software adoption. It also provides practical advice to municipalities regarding how to leverage collaborative digital production in order to derive greater benefit for themselves and the public using limited resources.

Part I sets forth certain advantages of free software over its proprietary counterparts and discusses in detail the opportunity free software presents to municipalities, including operational autonomy, transparency and security, prevention of vendor lock-in, and secondary benefits to the public. Part II highlights some of the social, economic, and legal obstacles to municipal use of free software, and attempts to present practical advice to address each. Part III highlights steps that municipalities might take to leverage free software and collaborative development in order to improve their technological infrastructure and civic reach.

I. THE BENEFITS OF FREE SOFTWARE FOR MUNICIPAL GOVERNMENTS

A. Traditional Benefits of Free Software: Freedom, Autonomy, and Communal Pooling of Resources

Many benefits the general public obtains through the use of free software are also present in the municipal context. The first and most obvious benefit is the financial savings due to free software's zero cost license. Indeed, this is often the main reason that governments seek free software solutions to begin with.[6] For many municipalities that pay per installation of office suites such as Microsoft Office, the savings from a switch to an equivalent free software suite such as OpenOffice.org can be immense. For example, in a bundle deal for

able to access the source code and not merely the computer executable object code that actually runs on the machine.

6. Jay Lyman, *Economy up or down, Can Open Source Come out on Top?*, 451 GROUP (Aug. 11, 2011, 2:19 PM), http://blogs.the451group.com/opensource/2011/08/11/economy-up-or-down-can-open-source-come-out-on-top/.

computing services for which Microsoft Office software was a major part, New York City pays twenty million dollars a year.[7] Contrast this with Katowice, a city in southern Poland, which, while operating on a much smaller scale, saved one hundred thousand Euros per year by switching to OpenOffice.org using minimal effort.[8] These savings can be amplified across other systems with a switch to or a build out of a free software solution, even when accounting for added training costs.[9] And because additional licenses cost nothing, municipalities can increase operations with non-linear software costs, since existing personnel may be used to support additional workstations or servers.

While license costs themselves are zero, it is often argued that, considering all factors, the total cost of ownership[10] of free software solutions is higher than their proprietary counterparts. While this is a controversial issue, experts generally agree that analysis must be done on a case by case basis,[11] and much of the available evidence suggests that "total cost of ownership for [free software] is often far less than proprietary software."[12]

Another benefit of free software is that it provides municipalities with the capability to pick and choose the features they wish to include in their software solutions. This is quite unlike proprietary solutions, where vendors force both upgrades and obsolesce of software or hardware.[13] For example, in recently passed legislation requiring state agencies to consider free software in procurement, the New

7. Ashlee Vance, *Microsoft and New York in Software Deal*, N.Y. TIMES, Oct. 21, 2010, at B2, *available at* http://www.nytimes.com/2010/10/21/technology/21soft.html.

8. Konrad Dwojak, *Katowice Municipality: Saving Public Money with OpenOffice.org*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/katowice-municipality-saving-public-money-openofficeorg.

9. *Open Source in Five Municipalities in Groningen*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/open-source-five-municipalities-groningen (finding a "[r]eduction of about 308 euros per workstation in comparison to Closed Source").

10. The total cost of ownership includes factors such as IT staff, hardware, software maintenance, and end of life-cycle considerations in addition to licensing costs.

11. Aaron Weiss, *Real World Open Source: The TCO Question*, SERVERWATCH (Aug. 24, 2005), http://www.serverwatch.com/tutorials/article.php/3529871/Real-World-Open-Source-The-TCO-Question.htm (concluding that there is "[n]o one size fits all answer to the [total cost of ownership] question").

12. David A. Wheeler, *Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*, DAVID A. WHEELER'S PERSONAL HOMEPAGE (Apr. 16, 2007), www.dwheeler.com/oss_fs_why.html (reviewing thirty studies or other data points concerning the total cost of ownership of free software systems as compared to their proprietary counterparts).

13. *See, e.g.*, *infra* notes 50–55.

Hampshire state legislature found that it was in the "public interest that the state be free, to the greatest extent possible, of conditions imposed by parties outside the state's control on how, and for how long, the state may use the software it has acquired."[14] Permitting upgrading on a schedule controlled by the municipality provides operational savings in two ways. First, it permits evaluation of each upgrade in a piecemeal fashion, allowing the city to apply only the upgrades it deems necessary, reducing the cost of training and support. Additionally, each incremental improvement can be implemented and rolled out as needed, instead of waiting for an entire new version of a system incorporating a laundry list of changes that necessitate a more comprehensive investment in training and support.[15]

The benefit derived from this flexibility is not limited to upgrades, but also applies to customization to fit local organizational needs.[16] For example, a reason the Evergreen ILS, a made-from-scratch solution for the Georgia Public Library system, registered initial success was that its development process allowed users to give feedback, and the information technology (IT) professionals on staff could quickly modify the software to suit these user requests.[17] This flexibility also enables extension of the Evergreen system to serve users with visual or other impairments.[18] The benefits of flexibility are also borne out empirically: for example, a survey of free software users and customers found that flexibility was the main benefit from free software

14. *See* H.B. 418-FN, 162nd Gen. Court, 2012 Sess. (N.H. 2012), *available at* http://www.gencourt.state.nh.us/legislation/2012/HB0418.html; *see also Dutch Municipality of Haren Migrating to Open Source Software*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/dutch-municipality-haren-migrating-open-source-software#introduction.

15. *Dutch Municipality of Haren Migrating to Open Source Software*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/dutch-municipality-haren-migrating-open-source-software#introduction.

16. Sandeep Krishnamurthy, *A Managerial Overview of Open Source Software,* BUS. HORIZONS, Sept.–Oct. 2003, at 47, 51–52.

17. Jonathan Weber, *Evergreen: Your Homegrown ILS*, LIBR. J. (Dec. 15, 2006), *available at* http://www.libraryjournal.com/article/CA6396354.html) (quoting a library director's comment on a free software system as "[t]he one thing that changed my mind [about the benefits] . . . was that I saw a demonstration of an early release, saw an immediate problem with one part of it, brought it up right then, and the next time I saw a demo, it was fixed.").

18. Josh McGee, *Georgia Public Library Service Receives National Leadership Grant For Software Development*, ROME NEWSWIRE (Oct. 4, 2011), http://romenewswire.com/2011/10/04/georgia-public-library-service-receives-national-leadership-grant-for-software-development/.

adoption.[19] This flexibility includes customization for municipal citizen's needs, such as language,[20] or fitting the solution to an altogether different purpose or environment.[21]

Finally, a municipality that uses free software works within a community of other adopters, allowing them to take and improve upon other users' improvements. This is a stated goal of at least a few adopters,[22] and it represents an important principle of free software: adoption drives innovation.[23] The nature of software, that once built it can be available to all at no further cost, means that each incremental improvement made by another user improves the common wealth; an improvement by one is an improvement to all.

When members can work together to implement new features, this community model also avoids duplication of effort and waste of resources. The Kuali Foundation, a nonprofit that designs software for institutions of higher education, illustrates both of these principals. Kuali members, which number over sixty and include many well-respected universities,[24] pool resources "to develop and sustain many of the software systems needed for higher education." In the Kuali Foundation's case, it appears that members are required to donate money, and are expected to devote manpower to improving the pro-

19. Jay Lyman, *Economy Up or Down, Can Open Source Come Out on Top?*, 451 GROUP (Aug. 11, 2011, 2:19 PM), http://blogs.the451group.com/opensource/2011/08/11/economy-up-or-down-can-open-source-come-out-on-top/.

20. *See, e.g.*, Dwojak, *supra* note 8 (noting the requirement that OpenOffice.org be installed in a stable Polish version).

21. Mary Brandel, *Adaptable Open Source*, NETWORKWORLD (May 10, 2010, 2:12 PM), http://www.networkworld.com/news/2010/051010-adaptable-open.html ("[flexibility means] the ability to . . . 'take a standard install and rip out the guts and do all kinds of weird stuff and make it fit our environment.'").

22. *Arles (France), in Progression Towards Open Source*, EUR. COMM'N JOINUP (Feb. 13, 2012), http://joinup.ec.europa.eu/software/studies/arles-france-progression-towards-open-source ("The major benefits of the IT modernisation of the city are the possibility to share and improve its Open Source experiences in the Adullact framework and to make investments more durable thanks to its participation in working-groups of other public sector organizations."); *Dutch Municipality of Haren Migrating to Open Source Software*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/dutch-municipality-haren-migrating-open-source-software#introduction; *Open Source Software Migration in the Belgian City of Schoten*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/open-source-software-migration-belgian-city-schoten.

23. *See* Robert A. Gehring, *The Institutionalization of Open Source*, 4 POIESIS AND PRAXIS 54, 70–71 (2006) (concluding that when there is a network of "stake holders" in a given program, the feedback loop of users making changes to a common resource provides a distinct market advantage).

24. *Members*, KUALI FOUND., http://kuali.org/Members (last visited Jan. 26, 2012).

jects for all other members,[25] creating a collaborative community that seeks to innovate in a common space for the collective good.

B. Secondary Benefits to the Public

When municipalities spend to procure or improve free software rather than proprietary alternatives, the public benefits both from the techniques taught and from the tools produced. This is because the software, if made available publicly, permits the general public to study and build upon the solutions produced by municipal spending.

Through their available source code, free software projects contribute to the public education by illustrating practical programming techniques and the details of their implementation. For example, computer science professors use free software operating systems based upon the Linux Kernel to teach students how to implement operating systems concepts by programming a component designed to work with the system.[26] Similarly, educators use the popular GCC compiler to teach students in compilation courses, and the free software X Windows system to teach graphics programming.[27]

In addition to these educational benefits, investing in free software spurs innovation, providing gains to all citizens that are paid for by the public sector at no additional cost. In the United States, this rationale has often been used to justify the high price of defense spending.[28] An example of this concept as applied to free software is a recent analysis of the European Space Agency's free software projects, which found that public and private organizations made use of the

25. *Kuali Foundation Membership*, KUALI FOUND., http://kuali.org/sites/default/files/old/Kuali_Membership_Agreement_Rev.9-22-2011.pdf (last visited Mar. 26, 2012).

26. *See, e.g.*, AMOS BROCCO & FULVIO FRAPOLLI, OPEN SOURCE IN HIGHER EDUCATION: CASE STUDY COMPUTER SCIENCE AT THE UNIVERSITY OF FRIBOURG (Feb. 14, 2011), *available at* http://www.unifr.ch/didactic/assets/files/travaux%20participants/BroccoFrapolli_diplome.pdf (providing a case study of the benefits of such a course).

27. *See, e.g.*, *Computer Science Course Descriptions*, EMORY UNIV., http://college.emory.edu/home/academic/course/descriptions/computer_science.html (last visited Feb. 19, 2012) (offering a course in graphical user interfaces titled "X Window System Programming"); *Optimization Course*, GCC WIKI (Jan. 10, 2008, 7:38 PM), http://gcc.gnu.org/wiki/OptimizationCourse (describing the optimization course using the GCC as a computer).

28. *See, e.g.*, Binyamin Appelbaum, *A Shrinking Military Budget May Take Neighbors with It*, N.Y. TIMES, Jan. 7, 2012, at A1, *available at* http://www.nytimes.com/2012/01/07/us/a-hidden-cost-of-military-cuts-could-be-invention-and-its-industries.htm (discussing the impact that the proposed military budget cuts may have on innovation).

software for instruction and scientific research in a variety of areas.[29] That said, the public benefits from free software adoption are not only scientific or educational, but also practical. For example, WebKit, a layout engine developed by the KDE e.V. Foundation to help browsers render webpages [30] has been adapted and included in a variety of different highly used free software and industry programs such as the default Android browser, Apple, Inc.'s Safari browser, and Google's Chrome browser.[31] This example illustrates a common concept: the availability of tools developed to serve public needs can serve the public's interest by constructing platforms on which the public themselves can build other useful tools.

Finally, investing in free software increases the educational opportunities for IT professionals. Instead of being tasked with supporting proprietary systems only using the tools provided by the vendor, public and private sector employees may be tasked with understanding and developing their own solutions, providing additional opportunities for education and skill development. As at least one study has shown, this paradigm leads to a workforce with a higher skill level than those supporting proprietary solutions.[32]

C. Transparency and Safety

The ready availability of the source code of free software, which allows the public to audit essential systems, benefits municipalities in a few areas. First, permitting the public to audit the code contained in systems provides a layer of transparency, increasing the public trust in government. Second, especially when a project uses existing code in wide public use, it results in software with fewer bugs and fewer security risks. This section will explore these concepts further, focusing

29. *Reuse of ESA Software*, EUR. COMM'N JOINUP (Feb. 13, 2012), https://joinup.ec.europa.eu/software/studies/reuse-esa-software.

30. THE WEBKIT OPEN SOURCE PROJECT, http://www.webkit.org/ (last visited Jan. 19, 2012) ("WebKit's HTML and JavaScript code began as a branch of the KHTML and KJS libraries from KDE.").

31. *Companies and Organizations that Have Contributed to WebKit*, WEBKIT.ORG, http://trac.webkit.org/wiki/Companies%20and%20Organizations%20that%20have%20contributed%20to%20WebKit (last visited Dec. 30, 2011); Jason D. O'Grady, *Google's New Chrome Browser Based on WebKit (updated 3x)*, ZDNET (Sept. 2, 2008, 9:21 AM), http://www.zdnet.com/blog/apple/googles-new-chrome-browser-based-on-webkit-updated-3x/2208 ("[T]he new gBrowser is [sic] using components from WebKit . . . .").

32. *See, e.g.*, Stan Beer, *Open Source Professionals Higher Skills, Higher Paid: Survey*, ITWIRE (Mar. 10, 2008, 6:12 PM), http://www.itwire.com/business-it-news/open-source/17063-open-source-professionals-higher-skills-higher-paid-survey.

on electronic voting machine software, although the model is applicable to many areas of civic government, including traffic, fire, and other safety focused software.

Municipal governments engage in a number of areas requiring transparency. But perhaps there is no area more important to a citizen's belief in its government than elections. Municipalities seeking cost-savings, immediate results from election administration, and an immense amount of federal money[33] have turned to electronic means for counting votes. Adoption of electronic voting machines from companies that keep their source code as a trade secret, however, has raised a variety of questions surrounding the proper role of source code in politics. In a number of recent elections, voters believed that the electronic machines had tallied their vote for the wrong candidate.[34] Additionally, the most popular voting manufacturer in the United States admitted recently that there was a programming bug that caused votes to be lost, and thus uncounted.[35] These two problems have severely undermined public confidence in the electoral process, causing commentators to question the wisdom of the implementation.[36] This public fear led *The Simpsons* to parody the voters' belief that the machines are rigged.[37] While not all of the criticism or problems would likely be cured through public access to the machine's source code,[38] adding an additional layer of transparency

33. The Help America Vote Act authorized $3.9 billion of spending to help states adopt new voting technology, among other things. Clive Thompson, *Can You Count on Voting Machines?*, N.Y. TIMES MAG. (Jan. 6, 2008), *available at* http://www.nytimes.com/2008/01/06/magazine/06Vote-t.html. New York City spent $77 million of its own money and $88 million of federal money for the voting machines put into service in 2010. James Barron & David W. Chen, *Problems Reported with New Voting Machines*, N.Y. TIMES CITY ROOM (Sept. 14, 2010, 11:16 AM), http://cityroom.blogs.nytimes.com/2010/09/14/problems-reported-with-new-voting-machines/.

34. *See, e.g.*, Scott Flynn, *WV Voters Say Machines Are Switching Dem Votes to GOP*, W.V. PUB. BROAD. (Oct. 22, 2008), http://www.wvpubcast.org/newsarticle.aspx?id=5588; Kim Zetter, *ES&S Voting Machines in Tennessee Flip Votes*, WIRED THREAT LEVEL (Oct. 23, 2008, 11:10 AM), http://www.wired.com/threatlevel/2008/10/ess-voting-mach; Kim Zetter, *Votes Flipped in Ohio Race That Used E-voting Machines*, WIRED THREAT LEVEL (Nov. 8, 2007, 11:20 AM), http://www.wired.com/threatlevel/2007/11/votes-flipped-i/.

35. Mary Pat Flaherty, *Ohio Voting Machines Contained Programming Error that Dropped Votes*, WASH. POST: TRAIL (Aug. 21, 2008, 5:09 PM), http://voices.washingtonpost.com/44/2008/08/ohio-voting-machines-contained.html.

36. Thompson, *supra* note 33.

37. Deebold08, *Homer Simpson Tries to Vote for Obama*, YOUTUBE (Sept. 29, 2008), http://www.youtube.com/watch?v=1aBaX9GPSaQ.

38. *See, e.g.*, James Barron & David W. Chen, *Problems Reported with New Voting Machines*, N.Y. TIMES CITY ROOM (Sept. 14, 2010, 11:16 AM), http://cityroom.

would not only increase confidence in government, but, if released under a free software license, would also enable the public to reap benefits from the immense public expense for the machines.

The open development of free software source code also usually increases the security of the software. There is a common misconception that access to source code *increases* security risks. While it is true that publishing source code may, as one commentator notes, provide a "free education" to the would be attacker,[39] as the Department of Defense recently noted, the idea that hiding source code might itself increase security ("security by obscurity") "is widely denigrated."[40]

The reason for this is that by publishing source code there are both an increased number of eyes at work to find security flaws, and there are more interested parties able to fix security problems as they become publicized. In other words, the availability of the code permits those with the strongest incentive to secure the software, the users, to rapidly find and fix security issues,[41] increasing reliability and security.[42] The Department of Defense agrees, noting: "[c]ontinuous and broad peer-review, enabled by publicly available source code, improves software reliability and security through the identification and elimination of defects that might otherwise go unrecognized by the core development team."[43] By contrast, in closed source programs: "[v]ulnerabilities often go unnoticed, unannounced, and unfixed []because the vendor, rather than users who have a higher stake in maintaining the quality of software, is the only party allowed to evaluate the security of the code base."[44]

---

blogs.nytimes.com/2010/09/14/problems-reported-with-new-voting-machines/ (noting the administration issues with New York City's switch to voting machines, many of which have nothing to do with software).

39. KENNETH BROWN, ALEXIS DE TOCQUEVILLE INST., OPENING THE OPEN SOURCE DEBATE 8 (June 2002), *available at* http://nats-www.informatik.uni-hamburg.de/pub/OSS2004/PaperCollection/AdTIOpensourceWhitepaper.pdf.

40. *DoD Open Source Software (OSS) FAQ*, DEP'T DEFENSE, http://dodcio.defense.gov/OpenSourceSoftwareFAQ.aspx (last visited Jan. 29, 2012).

41. Jaap-Henk Hoepman & Bart Jacobs, *Increased Security Through Open Source*, 50 COMM. ACM, Jan. 2007, at 79, 82; Peter P. Swire, *A Theory of Disclosure for Security and Competitive Reasons: Open Source, Proprietary Software, and Government Systems*, 42 HOUS. L. REV. 1333, 1337 (2006).

42. Sandeep Krishnamurthy, *A Managerial Overview of Open Source Software*, BUSINESS HORIZONS, Sept.–Oct. 2003, at 47, 51.

43. *DoD Open Source Software (OSS) FAQ*, *supra* note 40.

44. Karen M. Sandler et al., *Killed by Code: Software Transparency in Implantable Medical Devices*, SOFTWARE FREEDOM LAW CENTER (July 21, 2010), http://www.softwarefreedom.org/resources/2010/transparent-medical-devices.html.

Empirical research also supports the observation that publishing source code allows users to collaborate in an effort to combat attacks and improve defenses. In comparing free software programs to closed source programs, researchers have found that free software programs fixed security flaws faster and with a higher degree of efficacy than proprietary software.[45] The observation is also supported by a recent examination of voting machine software by security experts. After source code for a Diebold voting machine was leaked, security researchers performed an analysis of the code, concluding that there were "significant security flaws."[46] The researchers went on to note that "an open process would result in more careful development, as more scientists, software engineers, political activists, and others who value their democracy would be paying attention to the quality of the software that is used for their elections."[47]

D. Preventing Vendor Lock-in or Obsolescence

The last Section discussed the benefits derived mainly from the public availability of free software's source code. This Section will look at benefits that mainly derive from the rights granted by free software licenses, namely protection against obsolescence, abandonment, and vendor lock-in. When municipalities ensure that they obtain the rights to use and modify the software, their software procurement undergoes a paradigm shift. Instead of a good to be purchased, software becomes a commodity around which any vendor may design a competitive suite of services.[48] This is not to say that the vendor who initially wrote the software does not enjoy a competitive advantage; analogously, a dealer-licensed car repair shop has access to training materials and other knowledge not possessed by a generic

45. *Id.* (noting that an independent free software security analysis of 1591 commercial software applications concluded that free software applications took less time to fix software bugs, and that the quality of the repair was better in the free software context) (citing Veracode, 1 State of Software Security Report (2010), *available at* http://www.veracode.com/reports/index.html)).

46. Tadyoshi Kohno et al., *Analysis of an Electronic Voting System*, IEEE Symposium on Security and Privacy (May 2004).

47. *Id.* at 21.

48. Ian Murdock, *Open Sources 2.0/Open Source: Competition and Evolution/Open Source and the Commoditization of Software*, OPEN SOURCES 2.0, http://commons.oreilly.com/wiki/index.php/Open_Sources_2.0/Open_Source:_Competition_and_Evolution/Open_Source_and_the_Commoditization_of_Software (last visited June 12, 2012).

garage, increasing both service price and consumer trust.[49] Demanding licenses that permit copying, distribution and the preparation of derivative works, however, creates a secondary "after market" for the software. This after market can provide services related to the software, increasing competition.

Obsolescence of a product generally occurs in one of two situations: when a vendor discontinues the program or support, or when a vendor goes out of business. For example, according to current schedules, Windows XP will have no support, including patches of security holes, from Microsoft in 2014. Therefore, Microsoft customers must procure a new operating system from Microsoft or some other place if they wish to be able to patch security holes.[50] Also, the recent recession caused an increasing number of vendors to declare bankruptcy or otherwise fail to meet their support obligations to clients, with little or no notice.[51]

Faced with either of these situations, a municipality that seeks to continue support must make a choice: obtain the rights to modify the existing system so that new vendors may be found to support it, or incur the expense of building or purchasing an entirely new system. This situation most commonly occurs in niche markets.[52] Due to the wide variety of unique services that municipalities perform, these markets are often inhabited by municipal governments. Thus, obsolescence would leave a municipality without the ability to continue securely using the software they had already purchased, and unable to update this software system to meet future needs.

According to common commercial practice, software escrow, the act of storing source code with a third party to be released to the client under some restrictions in the event of discontinuation of support or bankruptcy, is the preferred solution to protect municipalities from

49. *See* Alina Tugend, *Who's Best for Your Car, Dealer or Independent?*, N.Y. TIMES, Feb. 25, 2011, at B6.

50. For example, many software vendors do not provide support for their system past a certain date, which ends fixes of security vulnerabilities. *See, e.g.*, David DeJean, *Windows XP: Going, Going . . . Gone?*, COMPUTERWORLD (Mar. 21, 2008, 12:00 PM), http://www.computerworld.com/s/article/9070119/Windows_XP_Going_going_..._gone_ (noting that Microsoft will not provide software upgrades for hardware).

51. *Recession Forces Software Escrow Releases to Jump by 150%*, OUT-LAW.COM (Jan. 5, 2010), http://www.out-law.com/page-10641.

52. Douglas Carnall, *Open Source Software in Healthcare*, LINUX USER MAG. (June 20, 2000), http://www.carnall.demon.co.uk/OpSrcHth.htm (in "[n]iche markets such as specialised healthcare applications software houses regularly fail and leave their customers in the lurch . . . .").

obsolescence.[53] The rights granted in the escrow agreement, however, may not protect the licensee completely. Often, the agreement may either limit the ability to modify the source code or does not permit redistribution of changes to contractors.[54] Additionally, even if these rights are fully granted, a common failing of the escrow solution is the time it takes for programmers to become familiar with the released code and be able to provide support. This failing, in turn, may drive up the cost to the point where it may be as expensive to attempt to rescue the project from escrow as it would to switch to a new system.[55] Although competent drafting of the agreement may resolve some of these issues,[56] problems commonly arise. For example, the New York State Elections Law was modified to require escrow for the machine's software in anticipation of the procurement of electronic voting machines.[57] The only use, however, of the escrowed code mandated by the law is for use in testing and not in granting of additional rights in case of a failure to support, or bankruptcy by the vendor.[58]

In contrast, free software, by definition, empowers municipalities to make changes. Because free software is also usually widely available, it often has an already established community support system or a secondary "after market" that may be hired or otherwise drawn upon for support in the event a partial vendor is unwilling or unable to continue supporting the software. For example, both the Evergreen

53. *See, e.g.*, Stephen M. McJohn, *The Paradoxes of Free Software*, 9 GEO. MASON L. REV. 25, 28 n.12 (2000) ("Where the licensee is dependent on the software, she may be concerned that the licensor will go out of business or, for some other reason, be unwilling or unable to modify the source code for future needs. In such settings, the parties often agree to put the source code in escrow, pending specified conditions. Such a transaction allows the licensor to maintain control over the source code while reassuring the licensee.").

54. *See* Dean Gloster, *Typical Source Code Escrow Agreements: What's Broken and What Works Instead*, FARELLABRAUN & MARTEL, LLP (May 25, 2005), http://www.fbm.com/media/uniEntity.aspx?xpST=PubDetail&pub=5253 ("And even if the source code escrow give[s] you the code, the license agreement may not give you the right to create derivative works, and may even actively prohibit you from showing that source code to any outside contractor brought in to assist with maintenance.").

55. Jonathan L. Mezrich, *Source Code Escrow: An Exercise in Futility?*, 5 MARQ. INTELL. PROP. L. REV. 117, 120–21 (2001) ("Even with access to the source code, the learning curve for a complicated software application is steep and may result in costs comparable to purchasing a whole new system or application.").

56. A well-drafted agreement, however, will not resolve every issue. If the code is kept as a trade secret, very few programmers will have intimate knowledge of its workings, and it will still take time to get new developers up to speed.

57. N.Y. ELEC. LAW § 7-208 (McKinney 2009).

58. *Id.*

ILS and the higher education software produced by the Kuali Foundation have many vendors offering support for their products.[59]

A similar problem to software obsolescence, vendor lock-in, occurs when the vendor makes it difficult to use another vendor without incurring substantial switching costs. Vendor lock-in can be accomplished through data dependence[60] or through application of patent rights, but is also a result of the fact that the copyrighted human readable code is kept as a trade secret or only permitted to be modified by the right-holding vendor.

The free software paradigm shift from products to services prevents this sort of lock-in. When software is a commodity good, it enables competition in support services through the removal of copyright and trade secret restrictions. Additionally, because the software source code is available, it is possible to determine how the software reads the customer data, and to design a competing solution using the same data. If we take as granted the notion that competition reduces cost, the savings to municipalities from reducing vendor lock-in are real and immediate. For example, out of ten new software or software related contracts apparently not procured through existing New York State contracts in 2011,[61] New York City has awarded six based on single source negotiations.[62] In one example, a five million dollar contract was awarded to a firm in a single source negotiation after the original software provider informed the city that they would not be supporting the integration with any other vendor's work.[63] By con-

59. *See infra* notes 93–94.

60. In other words, making the data created by the software in a format that is not publicly documented, and thus only readable by the proprietary solution.

61. Section 3-09 of the New York City procurement policy permits agencies to take a shortcut to procurement of goods if they determine that procuring through an existing state or federal contract results in a price "[l]ower than the prevailing market price" or, for services, a price that is "[f]air and reasonable." N.Y. CITY, N.Y., PROCUREMENT POL'Y BD. RULES, tit. 9, § 3-09 (2009).

62. *Search Archived Bid and Award Notices*, N.Y.C. CITYWIDE ADMIN. SERVS., http://a856-internet.nyc.gov/nycvendoronline/vendorsearch/asp/startSearchArchive.asp (select "Award" under the "1. Type of Notice" hyperlink; then select "By publish date" under the "2. Sort" hyperlink; then type "1/1/2011 to 1/1/2012" in the box under "3. Notices published After"; then type "software" in the box under "6. Keyword(s)"; then click "Submit.") (the six software contracts awarded through a single source negotiation process are: (1) PIN# 836081211612, Published 12/19/2011; (2) PIN# 83611S0007, Published 10/21/2011; (3) PIN# 81611S0009, Published 8/11/2011; (4) PIN# 52886846, Published 8/2/2011; (5) PIN# 2-1505-1040/11, Published 5/18/2011; (6) PIN# 02510XMIS041, Published 4/22/2011).

63. *Procurement Search Results For Archived Notices*, N.Y.C. CITYWIDE ADMIN. SERVS. (Apr. 22, 2011), http://a856-internet.nyc.gov/nycvendoronline/vendorsearch/asp/startSearchArchive.asp (type "02510XMIS041" in the box under "7. PIN Num-

trast, had the program been free software, the state would have been able to procure services from any competent vendor to integrate the new module, likely decreasing costs.

The inclusion of patent licenses within free software licenses[64] provides additional protection against vendor lock-in through the use of a patent's exclusive rights.[65] This is because both of the patent licenses included in Apache version 2 and GPL v3 inhibit the use of patents to restrict competition on the commodity software by providing a loss of patent rights to any licensee that instigates a patent suit alleging a patent claim is infringed by the program to which the license is applied. In this way, these licenses also enforce a "patent commons" surrounding each program to which they are applied. This potential loss of rights from all other contributors makes an attempt at patent lock-in over a particular commodity program to be socially, legally, and economically difficult, if not impossible.

Though both copyleft free software licenses and permissive[66] free software licenses provide the already-discussed protections against

ber"; the click "Submit"; and view the entry labeled "Medical Bill Review Software License Integration of Stratware Software With Gensource Software").

64. A recent development in free software licensing is the inclusion of a limited patent license in addition to the copyright license. These patent licenses often differ in a few ways. *Compare* APACHE SOFTWARE FOUND., APACHE LICENSE VERSION 2.0 § 3 (Jan. 2004), *available at* http://www.apache.org/licenses/LICENSE-2.0.html (granting a license only for claims necessarily infringed by the contributor's patch or the patch in combination with the work to which the patch was submitted), *with* FREE SOFTWARE FOUND., GNU GENERAL PUBLIC LICENSE § 11, GNU OPERATING SYS. (June 29, 2007), http://www.gnu.org/copyleft/gpl.html (granting a patent license for the claims that would be infringed by the work to which the contribution was made as a whole, but do not extend to claims that would only be infringed because of a further modification of the program).

65. Both the GPLv3 and the Apache License version 2.0 terminate all patent licenses that were granted to an entity if that entity instigates a patent suit over the program, thus attempting to build a safe harbor among contributors to the program. *See* APACHE SOFTWARE FOUND., APACHE LICENSE VERSION 2.0 § 3 (Jan. 2004), *available at* http://www.apache.org/licenses/LICENSE-2.0.html ("If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed."); FREE SOFTWARE FOUND., GNU GENERAL PUBLIC LICENSE §§ 8, 10 (June 29, 2007), http://www.gnu.org/copyleft/gpl.html (Section 10 provides that it is a license violation to "[i]nitiate litigation . . . alleging that any patent claim is infringed by . . . the Program or any portion of it." Section 8 provides that all rights granted under the license, including patent rights, terminate upon a violation.).

66. "Permissive" licenses permit derivative works created and distributed to be released under any terms of the author's choosing, including all rights reserved. Notably, these licenses impose no requirements on a licensor to make the source code

vendor lock-in, copyleft licenses go one step further and provide additional legal protection. Copyleft licenses require that modified versions of the code must also be licensed under the same copyleft license, and so improvements to the software continue to posses the legal and social properties preventing lock-in.[67] In the permissively licensed context, however, a vendor can take a permissively licensed commodity application, create its own application, and withhold all of the copyrights and source code to the improvements from the public. This limitation effectively creates a new product that the vendor may use to lock-in clients the same way as a vendor does who develops the software from scratch. Such an act may also fracture the community support for the project, with some following the now-proprietary vendor, and others sticking with the freely licensed version. This outcome erodes the commoditization of the project, which in turn degrades its resistance to vendor lock in.[68]

These challenges should not be taken to mean that vendor lock-in cannot be defeated using permissively licensed free software projects. To the contrary, some of the most well respected communities, such as the community supporting the Apache web server or the community fostered by the Kuali Foundation,[69] surround such projects, and combat lock-in by encouraging sharing and development of communi-

available, permitting an entity that modifies the code to distribute the program under a proprietary business model. Prominent examples of permissive licenses are the Apache License, Version 2.0, the MIT License (originating from the Massachusetts Institute of Technology), and the various forms of the BSD License (originating from the University of California, Berkeley).

67. "Copyleft" licenses are licenses that grant the public the rights necessary to use, modify, and distribute modifications of the software provided that any derivative works are released using the same license. These licenses thus use the statutory monopoly grant of copyright to ensure that a downstream recipient of a licensee, someone who receives a copy of the code from that licensee, has their freedoms protected. "Copyleft" licenses generally mandate that the source code for a derivative of the licensed program be distributed with the computer executable object code or otherwise be made available. *See* FREE SOFTWARE FOUND., GNU GENERAL PUBLIC LICENSE, VERSION 2 § 2 (June 1991), *available at* http://www.gnu.org/licenses/gpl-2.0.html; FREE SOFTWARE FOUND., GNU GENERAL PUBLIC LICENSE § 5 (June 29, 2007), *available at* http://www.gnu.org/copyleft/gpl.html.

68. *See* Murdock, *supra* note 48 (arguing that the fracturing of Unix utilities led to the acceptance and dominance of the Microsoft platform).

69. *See supra* notes 24–25. Notably, the license used by the Kuali Foundation has a weaker copyright license than both the GPLv3 and the Apache License Version 2.0. It only requires the grant of a patent right in a contribution when "[t]he individual that is the author of the Work is also the inventor of the patent claims licensed, and where the organization or institution has the right to grant such license under applicable grant and research funding agreements." *See* OPEN SOURCE INITIATIVE, EDUCATIONAL COMMUNITY LICENSE, VERSION 2.0 (ECL-2.0) § 3 (Apr. 2007).

ty solutions. One reason why these communities continue to provide free solutions without legal protection is the social enforcement of the sharing ethic that copyleft enforces with legal right.[70]

II. CHALLENGES TO FREE SOFTWARE ADOPTION

Despite the social, economic, and technological benefits to free software in the municipal context, its adoption faces many challenges. These include economic challenges stemming from software production, the costs associated from training staff to work with new systems, finding vendors to build or support the product, and the legal task of designing city procurement policies to account for the long term benefits of free software. This Part discusses each of these challenges in turn, and proposes means for municipalities to overcome these obstacles.

A. Entrenched Proprietary Vendors

Perhaps the greatest challenge for free software's adoption is proprietary solutions entrenchment in many markets. There is an inherent cost in switching software solutions in training staff on the new software and porting data.[71] Cost is one of the largest incentives attracting enterprise users to free software[72] and so the cost savings, *including the cost of switching systems*, must be attractive in comparison to the licensing fees charged by the proprietary vendors.[73]

70. *See, e.g.,* Rob Weir, *An Invitation to Apache Open Office*, ROBWEIR.COM (June 1, 2011), http://www.robweir.com/blog/2011/06/apache-openoffice.html (touting the Apache Software Foundation culture, which produces only permissively licensed software).

71. Many businesses and governments, including municipal governments, now insist on open data standards to ensure portability of data between software solutions. *See, e.g.*, Martin LaMonica, *Massachusetts to Adopt 'Open' Desktop*, ZDNET (Sept. 1, 2005), http://www.zdnet.com/news/massachusetts-to-adopt-open-desktop/144466 (quoting Massachusetts' Chief Information Officer as saying "[t]hese discussions have centered on open formats, particularly as they relate to office documents, their importance for the current and future accessibility of government records, and the relative 'openness' of the format options available to us"); Press Release, Office of Mayor Sam Adams, City of Portland, Oregon, Mayor Adams Introduces Open Source Resolution (Sept. 30, 2009), *available at* http://www.portlandonline.com/shared/cfm/image.cfm?id=265067) ("Moving to open data will have the added benefit of allowing inter-governmental and non-governmental agencies to readily access and leverage each others' information, lessening past practices of institutional siloing away of data.").

72. *See* Lyman, *supra* note 6.

73. *See* David A. Wheeler, *Open Source Software (OSS) in U.S. Government Acquisitions*, DAVID A. WHEELER'S PERSONAL HOMEPAGE (Dec. 17, 2010), http://

Compounding this issue is that proprietary software vendors for "off-the-shelf"[74] solutions are able to offer licenses to municipalities at nearly zero cost and still turn a profit. This is because the marginal cost of an additional license to the vendor of an off-the-shelf program is very near-zero,[75] enabling steep discounts in the event of a competitive bid from another vendor.[76] For example, a recent licensing deal between New York City and Microsoft touted discounts of fifty million dollars in software licenses and services that commentators attribute to New York City's threat to move to Google apps or open source software.[77] In addition, a renegotiation of Microsoft's license with MIT resulted in no-cost licenses for all MIT students.[78]

This dynamic puts municipalities into a "payday loan" type of situation with software vendors of off-the-shelf software. While the total cost of ownership for the software may be less for the free software solution, administrators cannot justify the up-front cost to move away from the proprietary vendor, especially when the vendor can sweeten the pot just enough for the administrator to view the transaction as a net savings.

Entrenchment also protects proprietary vendors where changes to the law or internal procedure require customization of an already licensed program. In the event that the customization needs to be performed, the software vendor retaining copyright is the only one who

www.dwheeler.com/essays/oss-government-acquisitions.html ("All too often an alternative system (OSS or not) will have a radically smaller [total cost of ownership], yet will not be used because of significant transition costs.").

74. "Off the shelf" software refers to software that does not require customization between customers.

75. *See, e.g.*, Johan Soderberg, *Copyleft vs. Copyright: A Marxist Critique*, FIRST MONDAY 7(3) (Mar. 4, 2002), http://firstmonday.org/htbin/cgiwrap/bin/ojs/index.php/fm/article/view/938/860 ("Digital information can be duplicated infinitely in perfect copies at a marginal cost approaching zero.").

76. Microsoft Volume Licensing for Local, City, and Regional Governments, MICROSOFT, http://www.microsoft.com/industry/government/howtobuy/state/local_regional.aspx#OpenLicense (last visited Apr. 2, 2012) (noting discounts on license procurement options for state and municipal governments with the purchase of at least five licenses).

77. Simon Phipps, *New York City Got a Better Deal from Microsoft — You Can Too*, INFOWORLD (Oct. 22, 2010, 3:00 AM), http://www.infoworld.com/print/141488.

78. Joseph De Avila, *New York City Sets Deal with Microsoft*, WALL ST. J. METROPOLIS (Oct. 20, 2010, 3:00 PM), http://blogs.wsj.com/metropolis/2010/10/20/new-york-city-sets-deal-with-microsoft/ ("The agreement is expected to save the city $50 million over five years."); Deborah Bowser, *Microsoft Office Now Available to MIT Students at No Cost*, MITNEWS (Aug. 5, 2011), http://web.mit.edu/newsoffice/2011/microsoft-office-students.html (noting the aggressive negotiation by MIT to secure the additional licenses).

can authorize modification, requiring a single-source negotiation resulting in monopolistic prices, such as the one commented on earlier.[79]

B. Services vs. Goods

Another challenge is the shift from the categorization of the major expense in procurement from goods to services. As one government project manager noted, development and labor costs raised red flags among the staff, while high licensing fees were the norm.[80] This notion is codified in the New York City procurement guidelines, where procurement for services (but not goods) over $100,000 requires additional justification by the procurement officer.[81]

This scrutiny is not limited to procurement officers. In one city, ostensibly to protect jobs for government workers, any proposal for services needed to clear a labor review, introducing an added layer of review for the procurement officer.[82]

Thus, although there are benefits when free software licensing and source code availability shifts the paradigm from the procurement of goods to the procurement of services, this shift also introduces barriers for free software adoption that may only be overcome with systemic changes to procurement methods and procedures.[83]

C. Finding Solvent Vendors

A larger issue for some free software solutions, is that vendors are either unavailable or do not place bids responding to government requests.[84] This puts municipalities desiring to use free software that fits their needs, but that lacks commercial support, in a difficult position.

79. *See supra* notes 61–63 and accompanying text.

80. *Procurement Interview Subject 1*, CIVIC COMMONS, http://wiki.civiccommons.org/Procurement_Interview_Subject_1 (last visited Jan. 16, 2012).

81. N.Y. CITY, N.Y., PROCUREMENT POL'Y BD. RULES, tit. 9, § 2-01 (2009).

82. *Procurement Interview Subject 1*, *supra* note 80 ("One scope of services with a vendor that the city was drafting for customization to a [product] had to go before a 'Civic Servants Group' which would question the purchase for its need to use outside labor vs. labor from the IT department.").

83. *See infra* notes 95–109 and accompanying text.

84. *Procurement Interview Subject 1*, *supra* note 80 (noting some free software solutions do not have vendors providing commercial support); *Procurement Interview Subject 12*, CIVIC COMMONS, http://wiki.civiccommons.org/Procurement_Interview_Subject_12 (last visited Jan. 16, 2012) ("Most [free software vendors] lack the marketing resources required to perform elaborate, speculative 'demos' or to respond to multi-hundred-page RFPs.").

A related obstacle for the municipality in either self-support or the choice of a small vendor is a lack of indemnity for copyright or patent infringement. Because free software licenses usually explicitly disclaim warranties,[85] if the vendor wishes to disclaim warranties, it does so without community support.[86] This is particularly troubling when procuring services from smaller vendors, who may not be large enough to cover potential damages even if they attempted to. While patent or copyright damages are generally barred by the Eleventh Amendment for suits against states,[87] they are available against municipalities,[88] making this an important consideration for municipalities that states do not face.

While there is no magic bullet to overcoming these two related issues, the example of the Evergreen Project provides support for an 'if you build it they will come' model for finding solvent vendors. The Evergreen Integrated Library System (ILS) was born after the Georgia Public Library Service (GPLS) found that its needs were "frustrated by the commercial [integrated library system] market."[89] So GPLS turned to an in-house solution by hiring three software developers full time to work on creating the Evergreen ILS.[90] The fact that Evergreen was free software permitted it to deliver both flexibility and immense cost savings, and it was released under the GNU General Public License, version 2 or later.[91]

85. *See, e.g.*, APACHE SOFTWARE FOUND., APACHE LICENSE VERSION 2.0 § 7 (Jan. 2004), *available at* http://www.apache.org/licenses/LICENSE-2.0.html.; FREE SOFTWARE FOUND., GNU GENERAL PUBLIC LICENSE § 15, GNU OPERATING SYS. (June 29, 2007), http://www.gnu.org/copyleft/gpl.html.

86. Linda M. Hamel, *Nine Ways to Protect your State from the Legal Risks Posed by the Use of Open Source Software*, MASS.GOV (Sept. 17, 2004), http://www.mass.gov/anf/research-and-tech/it-serv-and-support/application-serv/open-initiatives/open-source-legal-toolkit/nine-ways-to-protect-your.html.

87. *See* Fla. Prepaid Postsecondary Educ. Expense Bd. v. College Sav. Bank, 527 U.S. 627, 627 (1999) (holding that a provision of the Patent Remedy Act abrogating the States' sovereign immunity under the Eleventh Amendment from patent infringement suits was invalid).

88. Jinks v. Richland County, 538 U.S. 456, 465–66 (2003) ("Although we have held that Congress lacks authority under Article I to override a *State's* immunity from suit in its own courts . . . it may subject a *municipality* to suit in state court if that is done pursuant to a valid exercise of its enumerated powers . . . .") (citing Alden v. Maine, 527 U. S. 706 (1999)).

89. Jonathan Weber, *Evergreen: Your Homegrown ILS*, LIBR. J. (Dec. 15, 2006), *available at* http://www.libraryjournal.com/article/CA6396354.html).

90. *Id.*

91. *Id.*

Although originally an in-house project, there are now over fifty libraries using the system[92] and at least twelve corporations advertising Evergreen services.[93] As more libraries turn to Evergreen for their systems, these vendors will most likely continue to mature, extend their services, and develop deeper pockets to provide assurance and indemnity to the cities they serve.[94]

D. Procurement Policies

Another obstacle is that current procurement practices do not know how to account for the free software benefits, which, as community benefits, are hard to quantify. In the last decade, however, many municipalities and other governments have recognized the benefits of free software and adopted preferential or equal based treatment for free software in their procurement policies. For example, in 2010, San Francisco put a policy in place that requires procurement officials to seek out and consider free software alternatives for purchases of new software over one-hundred-thousand dollars. If the chief information officer determines that a "department has not made a good faith effort to consider open source alternatives," she is authorized to nullify the purchase.[95] Similarly, a new Australian policy requires (1) the Australian government to consider free software on an equal footing with proprietary software; (2) suppliers "to provide justification outlining their consideration and/or exclusion of open source software in their response to the tender"; and (3) the Australian Government to participate actively in free software projects.[96] In

92. *Evergreen Libraries*, EVERGREEN (May 8, 2011), http://evergreenils.org/dokuwiki/doku.php?id=evergreen_libraries.

93. *Commercial Companies that Advertise Evergreen Services*, EVERGREEN (Jan. 9, 2012, 1:05 PM), http://evergreen-ils.org/dokuwiki/doku.php?id=faqs:evergreen_companies.

94. Another successful example of this phenomenon is the Kuali Foundation, which, after building a successful community of educational institutions, has attracted many software commercial affiliates, including well known entities with deep pockets. *See Current Affiliates*, KUALI FOUND., http://kuali.org/current-affiliates (last visited Jan. 26, 2012).

95. *COIT Software Evaluation Policy*, CITY & COUNTY S.F. COMM. INFO. TECH. (Feb. 1, 2010), http://www.sfcoit.org/index.aspx?page=616.

96. AUSTL. GOV'T DEP'T OF FIN. AND DEREGULATION, OPEN SOURCE SOFTWARE POLICY, CIRCULAR NO. 2010/004 (Jan. 13, 2011), *available at* http://www.finance.gov.au/e-government/strategy-and-governance/docs/2010-004_AGIMO_Circular_Open_Source_Software_Policy.pdf.

contrast, the United States has declared a "technology neutral" policy.[97]

Preferential policies, which are designed to encourage free software procurement and development, also spark controversy. Critics of preferential policies argue that preferential policies would deprive governments of closed-source efficient solutions and discourage research and development.[98] Regardless of this contention, there is another question as to whether such policies actually result in increased free software procured.

A 2010 study set out to find just that. The study examined requests for proposals subject to a three-year-old Dutch procurement policy that gives preference to free software in cases where it is equally suitable to a task and gives providers of free software "[t]he same opportunities in practice . . ." in software procurement.[99] The study found that in over forty-five percent of tenders, free software was not given an equal chance to win the bid.[100] Whether this inefficacy is due to the newness of the policy, or for some other reason, remains to be seen as more policies come into effect, additional data is created, and potentially more free software vendors are able to service more of the public sector.

A recent development may increase the efficacy of procurement policies that favor free software. Instead of the 'equal footing' considerations in the Dutch policy,[101] or Australian requirement that *suppliers* search for free software solutions,[102] the recently passed New Hampshire policy[103] and the San Francisco[104] policy mandate

97. VIVEK KUNDRA ET AL., U.S. OFFICE OF MGMT. AND BUDGET, MEMORANDUM FOR CHIEF INFORMATION OFFICERS AND SENIOR PROCUREMENT EXECUTIVES (Jan. 7, 2011), *available at* http://www.cio.gov/documents/Technology-Neutrality.pdf.

98. Francis M. Buono & McLean B. Sieverding, *Government Procurement of Software: Provident Policies for Ensuring the Greatest Possible Return on Investment in Troubled Economic Times*, 3 BLOOMBERG LAW REPORTS—INTELLECTUAL PROPERTY 23 (2009), *available at* http://www.willkie.com/files/tbl_s29Publications%5CFileUpload5686%5C3029%5CGovernment%20Procurement%20of%20Software%20Provident%20Policies%20for%20Ensuring.pdf.

99. NETHERLANDS MINISTRY OF ECON. AFFAIRS, THE NETHERLANDS IN OPEN CONNECTION 17 (2007), *available at* http://www.whitehouse.gov/files/documents/ostp/opengov_inbox/nl-in-open-connection.pdf.

100. Mathieu Paapst, *Affirmative Action in Procurement for Open Standards and FLOSS*, 2 INT'L FREE & OPEN SOURCE SOFTWARE L. REV. 181 (2010).

101. *See supra* note 97.

102. *See supra* note 94.

103. *See* H.B. 418-FN, 162nd Gen. Court, 2012 Sess. (N.H. 2012), *available at* http://www.gencourt.state.nh.us/legislation/2012/HB0418.html.

oversight by the relevant technology agency to help identify and determine the cost efficacy of a free software solution. This approach may change the empirical results by bringing more specialization into the procurement process, and help bring additional free software solutions to the attention of procurement officials.

Additionally, recent thinking on the topic has begun to explore a new procurement vector that can be nurturing to free software. Recently, the New York City Metropolitan Transit Authority (MTA) procured a new bus tracking system. The MTA separated its procurement into two stages, a hardware stage, and a software stage.[105] This practice requires detailed open specifications to be developed for communication between the hardware and software layers of the same project. This documentation is a requirement often overlooked or neglected when a vendor wins a bid and keeps its architecture, both hardware and software, in a secret monolithic "black box." These open specifications, called application programming interfaces (API) increase the modularity of the end product, enabling different software to communicate with the hardware using the API. Additionally, the MTA required that the software that collects the information from the buses be open source and have a well defined, publicly facing API so that its data can be obtained by any application using the proper calls.[106] This practice results in procurement that builds "systems that become platforms upon which *anyone* can build new services."[107] There are a few benefits to this trend. First, it provides additional insulation against vendor lock-in since the hardware and software each has a well defined point of interface, and the policy forbids that interface from being kept as a trade secret. This protection ensures that "[t]he MTA can use different software or hardware vendors for future phases, or even use multiple different vendors simultaneously" to design hardware or software that works with the

104. *See supra* note 93; *see also City and County of San Francisco Software Evaluation Method*, CITY & COUNTY S.F. COMM. INFO. TECH. http://sfcoit.org/Modules/ShowDocument.aspx?documentid=385 (last visited Feb. 19, 2012).

105. METRO. TRANSIT AUTH., FINANCE COMM. MEETING SEPTEMBER 2011 V-57 (Sept. 26, 2011) [hereinafter MTA SEPT. MEETING], *available at* http://www.mta.info/mta/news/books/pdf/110926_1230_Finance.pdf; METRO. TRANSIT AUTH., FINANCE COMM. MEETING JULY 2011 V-8 (July 25, 2011) [hereinafter MTA JULY MEETING], *available at* http://www.mta.info/mta/news/books/pdf/110725_1230_Finance.pdf.

106. MTA SEPT. MEETING, *supra* note 105; MTA JULY MEETING, *supra* note 105.

107. Karl Fogel, *New York City Bus Tracking: Procuring for an Open Architecture*, CIVIC COMMONS (Dec. 7, 2011), http://civiccommons.org/2011/12/nyc-bus-tracking-as-platform/.

API.[108] This advantage will likely drive out future single source bids and increase competition on renewal bids, thus driving down costs. Second, the software interface's publicly facing API permits any application to interact with the server data, permitting the public to craft their own software using the platform, expanding the reach of the MTA's initiative beyond what the MTA could have afforded on its own.[109]

This analysis reveals a new dimension around how procurement may be improved to prevent lock-in and encourage open development in response to procurement requests. By designing requests for proposals in a way that ensures open documentation, municipalities can prevent lock-in. Further, they are able to realize the power of an interested community to expand the reach of their initiatives. In this way, the public expense, paired with private initiative, may result in a better community for all.

III. STEPS MUNICIPALITIES CAN TAKE TO LEVERAGE FREE SOFTWARE

After this broad overview of the benefits and obstacles to municipalities' use of free software, it may be useful to focus on five obtainable policy choices that can be implemented by city governments in order to best leverage the benefits offered by free software in their operations.

A. Create and Foster Collaboration Among Peer Cities and Vendors

First, municipalities should design their policies to capitalize on the observation that widespread adoption of a program drives innovation as each adopter adds its own improvements to the community. Municipalities desiring to use free software should attempt to create and foster collaboration among the city's employees, paid or volunteered developers, and other municipal users of the software. This approach is especially crucial for software that performs or supports services

108. *Id.*

109. *Id.* This concept is implemented by Civic Common's Open311 project, which provides open API's for 311 city data, permitting the public to design applications to interact with the data in various ways, and expanding the reach of the city's data collection and management. Philip Ashlock, *311 Pioneering Baltimore Continues to Lead with Open311*, OPEN311 (Sept. 10, 2011, 1:41 PM), http://open311.org/2011/09/baltimore/.

that are unique to municipalities.[110] In one example of what ideal collaboration between municipalities could look like, Canadian policy analyst David Eaves proposes a government-built free software repository and website specifically designed for municipal governments.[111] Indeed, Civic Commons was recently founded with an aim to bring governments together around free software, build open platforms, and develop community.[112] Such organizations serve as a locus for development, a communication point for municipalities, developers, and vendors, and an educational resource for municipalities looking to learn more. If the trend continues, these types of resources will continue to make free software solutions with robust easy-to-find community support, which will increase potential success in adoption.

B. Begin with Solutions That Already Have a Robust Community

The Evergreen ILS is a bright example of a successful municipal project started from scratch. As the previous Section suggests, however, designing a process around an existing project probably increases the chances of success because there is more likely to be an established community to provide advice, as well as solvent vendors to provide support and indemnity. Beginning with an established project will help the city become familiar with the process of procuring free software, and can help the city set a foundation for future projects.

C. Licensing Matters

Although this Article highlights only a few of the most used licenses and their legal features, it is important to keep in mind that there are literally dozens of free software licenses,[113] and that some of these licenses are incompatible, meaning that, when code released under both licenses is combined into the same program, there are situations

110. For example, the Sahana project is a free software project designed to assist in disaster response. *About Us*, SAHANA SOFTWARE FOUND., http://sahanafoundation.org/about-us/ (last visited Jan. 29, 2012).

111. David Eaves, *MuniForge: Creating municipalities that work like the web*, EAVES.CA (Dec. 8, 2009), http://eaves.ca/2009/12/08/muniforge-creating-municipalities-that-work-like-the-web/.

112. *See, e.g.*, *About*, CIVIC COMMONS, http://civiccommons.org/about (last visited Jan. 26, 2012).

113. For one list, see *Licenses by Name*, OPEN SOURCE INITIATIVE, http://www.opensource.org/licenses/alphabetical (last visited Dec. 28, 2011).

where there is no way to satisfy both licenses' requirements at the same time.[114]

There are other important characteristics of a chosen license to consider as well, for example, whether or not a patent commons surrounding the project is desirable,[115] or whether it is a priority to preserve all derivative works of the project as free software using the legal force of copyleft licensing.[116] Various resources for assisting with these decisions exist; one example is the list maintained by the Free Software Foundation[117] or the Open Source Initiative.[118]

D. Design Procurement Policies to Include a Step Where the Procurement Officer Consults with City IT Staff About Available Free Software Solutions

Municipalities should plan a centralized process including a step with oversight or consulting with city IT staff regarding available free software solutions. Building such a step into the procurement process promotes community participation and encourages corporations to pitch free software solutions. This approach broadens the breadth of available solutions brought to the attention of the procurement officer, and does not rely upon interested parties to bring free software solutions to the attention of the procurement officer.

E. Design Requests for Proposal that Require Well Defined Open Standards

As the MTA example demonstrates, city procurement that contemplates well defined interaction between software systems can reduce vendor lock-in and encourage community participation. Building software platforms that provide public access to services allows citizens to donate time and energy improving their interactions with city government; and permits citizens to lend a hand in constructing the digital city infrastructure they would like to have.

114. For example, the Free Software Foundation maintains a list of licenses with comments including information about whether the license is compatible with different licenses released by the organization. *See Various Licenses and Comments About Them*, GNU OPERATING SYS., http://www.gnu.org/licenses/license-list.html (last visited Dec. 28, 2011).

115. *See supra* notes 64–65 and accompanying text.

116. *See supra* note 67 and accompanying text.

117. *Various Licenses and Comments About Them*, GNU OPERATING SYS. (Feb. 22, 2012, 10:27 PM), http://www.gnu.org/licenses/license-list.html.

118. *Open Source Licenses*, OPEN SOURCE INITIATIVE, http://www.opensource.org/licenses (last visited Feb. 19, 2012).