

Contents

Articles

- Towards Free/Libre Open Source Software ("FLOSS") Governance in the Organisation 61
Richard Kemp

- Balancing free with IP: if open source solutions become de facto standards, could competition law start to bite? 73
Susannah Sheppard

- Packaging Open Source 83
Mark Webbink

- Passport Without A Visa: Open Source Software Licensing and Trademarks 99
Tiki Dare; Harvey Anderson

Case law reports

- BACK TO THE FUTURE: Hinton v Donaldson, Wood and Meurose (Court of Session, Scotland, 28th July, 1773) 111
Iain G. Mitchell QC

- A look at EDU 4 v. AFPA, also known as the "Paris GPL case" 123
Martin von Willebrand

Book reviews

- "Law and the Internet" - 3rd Edition, edited by Lilian Edwards and Charlotte Waelde 127
Andrew Katz

Tech Watch

- Patents and their effect on Standards: Open video codecs for HTML5 131
Sylvia Pfeiffer

- 5 Reasons To Care About Mobile Cloud Computing 139
Fabrizio Capobianco

Platform

- Open Source Policies and Processes For In-Bound Software 143
Karen F. Copenhaver



About IFOSS L. Rev.

Editorial Committee

This publication is managed by a rotating Editorial Committee. The membership of the Committee for this issue is as follows:

Amanda Brock

Amanda Brock is General Counsel of Canonical, the commercial sponsor of the Ubuntu project. Having graduated with Honours from Glasgow University, Amanda went on to obtain a Masters in Comparative Jurisprudence from New York University Law School and a LLM in IT and IP law from Queen Mary and Westfield, University of London. She is author of E:Business; The Practical Guide to the Laws and has written extensively on commercial and IT law.

Michel Jaccard

Michel is a partner in the technology and corporate practice groups of BCCC Avocats, an independent business law firm in Switzerland. He studied and worked in Lausanne, Geneva and New York, and taught and published extensively on legal issues relating to software licensing, information technology / business process outsourcing structures, data protection and online services. He regularly advises clients in Switzerland and abroad on intellectual property and technology as well as on corporate acquisitions, disposals and collaboration.

Andrew Katz

Andrew Katz studied Natural Sciences and Law at Cambridge University where he graduated with honours in 1989. In 1991 he was called to the Bar, and in 1993 requalified as a solicitor. He moonlighted as a programmer during his studies at Bar School, programming in Turbo Pascal. He has released software under the GPL. He is currently a partner at Moorcrofts LLP, a boutique law firm in England's Thames Valley and advises a wide range of businesses on free and open source related issues. He has lectured and published widely on the subject.

Iain G. Mitchell QC

Chairman, Scottish Society for Computers and Law; Chairman, Scottish Lawyers' European Group; Chairman, Faculty of Advocates IT Group; Lecturer, Honorary Board of Lecturers, Institut für Informations-, Telekommunikations- und Medienrecht, Westfälische Wilhelms-Universität, Münster; Freeman, Worshipful Company of Information Technologists.

Carlo Piana

Independent lawyer specialising in Information Technology and Telecommunication Law and Free Software Advocate. Started by using GNU/Linux and became intrigued by the legal and philosophical implications of it. Serves as Counsel to the Free Software Foundation Europe and advises projects and companies active in Free and Open Source Software.

Tomasz Rychlicki

Tomasz Rychlicki graduated from the University of Gdańsk, the Faculty of Law, Center of European Law. He also studied at Chicago-Kent College of Law in the LL.M. Program in International Intellectual Property Law. Tomasz currently works for PATPOL, Warsaw. He is a member of the Editorial Board (Copyright, Related Rights and Designs, including sui generis database right) at the Journal of Intellectual Property Law & Practice published by Oxford University Press, Oxford.

Brendan Scott

Brendan runs a legal practice based in Sydney, Australia. Brendan is a founding member and a director of Open Source Industry Australia Limited. He is a past president of the NSW Society for Computers and the Law and a past editor of its journal. He has over 15 years of experience in Technology and Telecommunications law and has a special interest in open source and the law related to it.

Ywein Van den Brande

Ywein Van den Brande (b. 1977) studied law, information science and economics at the Catholic University of Leuven and Groep T (Leuven, Belgium), the Université de Rouen (Rouen, France) and U.C.L.A. (Los Angeles, USA). Ywein is an active member of the Free Software Foundation Europe and is a member of workgroups on information architecture and usability. Ywein is the author of several legal contributions on law and technology, and regularly speaks at conferences. He is a guest lecturer at Groep T and blogs on the CIF blog of the journal www.tijd.be.

Martin von Haller Groenbaek

Martin von Haller Groenbaek is a founding partner at the leading Scandinavian IT law firm, Bender von Haller Dragsted. Among other thing he specialises in open source, open content and open data licences. He is a co-founder of the Danish Open Source Vendors Association and of the Danish chapters of the Internet Society and Creative Commons. His blog is at <http://www.openlife.dk>. His online CV is at <http://www.linkedin.com/in/vonhaller>.

Mark Webbink

Mark Webbink is a Visiting Professor of Law and Executive Director of the Center for Patent Innovations at New York Law School. Webbink is also a Senior Lecturing Fellow at Duke Law School and has served as an Adjunct Professor at NCCU Law School. From 2000 to 2007 Webbink served in various capacities with Red Hat, Inc., including General Counsel, Deputy General Counsel for Intellectual Property, Senior Vice President and Secretary. Webbink presently serves on the board of directors of the Software Freedom Law Center.

Editorial Coordinators

The editors wish to thank the Editorial Coordinators for their hard work and contribution to making the Review happen.

Graeme R. West (graeme@heliocentrik.net)

Graeme is a Masters student at the University of Edinburgh law school's SCRIPT IP and technology law centre, and is a former intern at the Free Software Foundation Europe. His academic background is in political science, in particular the global politics of IP. He has also worked in large-scale digital archival projects and web application development.

Shane Martin Coughlan (shane@opendawn.com)

Shane's academic background is in linguistics and politics, and his postgraduate research focused on cybernetic warfare. He founded FSFE's legal department in 2006, established the European Legal Network in 2007, and today focuses on improving efficiency in international technology markets.

Peer reviewers

The Editorial Committee wishes to thank the work of the many referees and peer reviewers whose professional expertise and dedication to high standards have made the publication of this issue possible.

Contact

All administrative, bibliographic and pre-publication enquiries should be directed to the Editorial Coordinators via email at:

admin@ifosslr.org

The Editorial Committee can be contacted via email at:

ed-com@ifosslr.org

Policies and bibliographic information

Copyright and licensing statement

IFOSS L. Rev. is committed to the improvement of understanding of legal issues in digital society. A licensing statement is therefore attached to each article, clearly outlining the particular terms which apply to the article. Most use Creative Commons licences with special exceptions for translations.

Graphic design

The Editorial Committee wishes to thank Tomasz Politański Design for its logo and associated graphic design work.

<http://tomaszpolitanski.com>

Publisher & sponsorship

IFOSS L. Rev. is published by its Editorial Committee, with financial and administrative assistance from NLnet Foundation and Mozilla Foundation. Please note that neither NLnet Foundation nor Mozilla Foundation accept correspondence on behalf of this publication. All correspondence should be directed to the Editorial Committee via email (see below). The Open Invention Network also sponsors this issue of IFOSS L. Rev.

Editorial policies

IFOSS L. Rev. accepts articles for publication from qualified personnel based on the criteria available to view on its web site (<http://www.ifossl.org>). Submissions are welcome from all, and

authors are strongly encouraged to read the style and content guidelines available on the web site. The review operates an anonymous peer review system for articles as appropriate, and expects all authors to meet the highest standards of scholarship and integrity.

Bibliographic information

The authors explicitly encourage libraries, archives and educational institutions to hold copies of IFOSS L. Rev. in their collections, in electronic and/or printed form. All users are advised that articles may occasionally be updated after publication. Linking back to original copies on the IFOSSL. Rev. web site, where authoritative versions are archived, is strongly recommended. Please contact the Editorial Coordinators for further information on best practices. It participates in the CrossRef system. ISSN: 1877-6922

Publication schedule

IFOSS L. Rev. is published biannually. Submissions for publication are welcome at any time, but publication deadlines exist for each issue. For the latest information on papers sought and deadlines for submission, please consult the IFOSS L. Rev. web-site or contact the Editorial Coordinators at (admin@ifossl.org)

openinventionnetworkSM

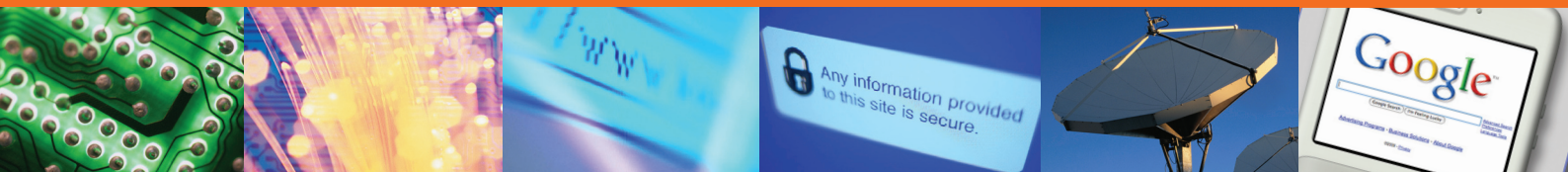
A Distinct Mission: Keeping Open Source Open

Open Invention NetworkSM is a collaborative enterprise that enables innovation in open source and an increasingly vibrant ecosystem around Linux by acquiring and licensing patents, influencing behaviors and policy and protecting the integrity of the ecosystem through strategic programs such as Linux Defenders. It enables the growth and continuation of open source software by fostering a healthy Linux ecosystem of investors, vendors, developers and users.

Open Invention Network (OIN) has considerable industry backing. It was launched in 2005, and has received investments from IBM, NEC, Novell, Philips, Red Hat and Sony.

One of the key methods in which OIN promotes Linux is by acquiring patents across a wide range of technologies.

Patents owned by OIN are available royalty-free to any company, institution or individual that agrees not to assert its patents against the Linux Community.



Linux Defenders (www.linuxdefenders.org)

Open Invention Network is seeking creative and energetic individuals from the Open Source Community to actively contribute to Linux Defenders.

Linux Defenders offers the Linux and broader open source community a unique opportunity to harness its collaborative passion, intelligence and ingenuity to ensure Linux's natural path of growth and innovation. The free program is designed to benefit open source innovation by significantly reducing the number of poor-quality patents that might otherwise be used by patent trolls, or other organizations, whose behaviors and business models are in opposition to those of the open source community.

The success of the program will be driven by contributions from the open source community. OIN encourages prior art and raw invention contributions that will be leveraged with the governmental patent authorities to invalidate poor-quality patents and patent applications.

openinventionnetworkSM

Research Triangle Park Center
4819 Emperor Boulevard
Suite 400
Durham, NC USA 27703
T: +1 919 313 4904
F: +1 919 313 4905
info@openinventionnetwork.com

www.openinventionnetwork.com



Towards Free/Libre Open Source Software (“FLOSS”) Governance in the Organisation

Richard Kemp^a

(a) Partner, Kemp Little LLP, London,
<http://www.kemplittle.com>

DOI: [10.5033/ifosslr.v1i2.19](https://doi.org/10.5033/ifosslr.v1i2.19)

Abstract

According to a Gartner survey, 85% of companies use FLOSS and the remaining 15% are expecting to use it in the next 12 months. However, there is still a disconnect between uptake and effective governance. This article articulates a practical approach towards implementing sensible, proportionate FLOSS governance focusing on the governance documentation concerned, and proposes a three level approach where the output of internal governance discussions are statements of Strategy, Policy and Process.

Keywords

Law; information technology; Free and Open Source Software; governance;

Info

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

TABLE OF CONTENTS

1. Introduction	62
(a) Recent surveys	62
(b) Purpose	63
(c) Scope	63
2. Fundamentals of FLOSS Governance	63
(a) Objectives	63
(b) Key principles	63
(c) FLOSS governance is particular to each organisation	64
(d) The range of organisations for which FLOSS governance is relevant	64
3. Contexts of FLOSS Governance – Building Blocks, Threads and Integration	65
(a) Thread 1: FLOSS achievements to date	65
(b) Thread 2: the people context	65
(c) Thread 3: the Strategy context	67
(d) Thread 4: the Policy context	67
(e) Thread 5: the Process context	70
4. Conclusion	71

1. Introduction

(a) Recent surveys

When IT consultancy Gartner released its survey in November 2008 of FLOSS use by 274 end user organisations around the world, it came up with two key findings. First, 85% of the companies surveyed then used FLOSS, with the remaining 15% then expecting to in the next 12 months – around the time of writing (November 2009). Secondly, 69% of the companies surveyed had no formal policy for evaluating or cataloguing FLOSS use in their organisation. As in the aftermath of the dotcom bust, continuing tougher economic times are hastening the uptake of FLOSS in the organisation, which on Gartner's figures, now approaches ubiquity. However, by all accounts there is still a disconnect between uptake and effective governance: on Gartner's figures, FLOSS governance remains more widely honoured in the breach than the observance. In the press release that accompanied its November 2008 survey, Laurie Worster, Gartner's research director said:

“Just because something is free doesn't mean it has no cost. Companies must have a policy for procuring FLOSS, deciding which applications will be supported by FLOSS and identifying the intellectual property risk or supportability risk associated with using FLOSS. Once a policy is in place, then there must be a governance process to enforce it”¹.

Gartner's findings are corroborated by a survey in March 2009 by Black Duck Software², which (although of a smaller survey sample) found that only 40% of larger companies (500 developers or more) had written governance policies and, of the sample as a whole, only one in five had written governance in place.

(b) Purpose

FLOSS governance is now, somewhat belatedly, rising up the corporate agenda³ and the purpose of this article is to articulate from a practical approach towards implementing sensible, proportionate FLOSS governance focusing on the governance documentation concerned⁴. This approach has as its start point that most organisations wish for reputational

¹ Gartner press release of 17 November 2008: “Gartner Says as Number of Business Processes Using Open-Source Software Increases, Companies Must Adopt and Enforce an OSS Policy”. Available at: <http://www.gartner.com/it/page.jsp?id=801412>

² Black Duck press release of 11 March 2009: “Black Duck Survey Reveals Open Source Development Trends”. Available at: <http://www.blackducksoftware.com/news/releases/2009-03-11>

³ See for example the abstract from IT research company Forrester's paper “Best Practices: Improve Development Effectiveness Through Strategic Adoption Of Open Source” of 2 February 2009: “[FLOSS] is getting renewed attention from application development professionals who are looking for cost-saving alternatives amid the economic recession. But many aren't asking the right question: Instead of “should we adopt [FLOSS]?” they should be asking, “how will we adopt [FLOSS]?” [FLOSS] is already seeping into development shops through a variety of channels, whether managers know it or not. Unchecked tactical adoption of [FLOSS] creates unmanaged risk and unrealized returns, and application development professionals should not tolerate it. Regardless of whether you view adoption of [FLOSS] as desirable or inevitable, the first step in moving from a tactical mess to a strategic plan is to specify the conditions under which [FLOSS] is permissible in your development shop. By creating a concise [FLOSS] policy, re-engineering the software acquisition process, and adding control points to [lifecycle management] processes and tools, application development professionals can shift from tactical responses to conscious integration based on realistic expectations and articulated economic benefits”, available at: <http://www.forrester.com/Research/Document/Excerpt/0,7211,46361,00.html>.

⁴ With the historically low take up of more formal FLOSS governance there has until recently been relatively little publicly available online material about FLOSS governance. FLOSS software/support developers Black Duck, Palamida and HP's FOSS Bazaar provide resources at:

- <http://www.blackducksoftware.com/resources/whitepapers#managingos> (Black Duck);

and competitive reasons to be seen in their use of FLOSS as in other matters to be good corporate citizens. It then proposes a three level approach where the output of internal governance discussions are statements of Strategy, Policy and Process that the relevant stakeholders buy into are then fully integrated around the organisation. There is no magic about such an approach, but it seeks to focus clearly on the high level issues, the policy that the organisation will define for its stakeholders and the day to day processes around implementation – the FLOSS governance toolkit.

(c) Scope

Organisations’ circumstances will differ widely so it is not practical to offer template ‘one size fits all’ documents. However this article offers in the tables below and the commentary pointers towards what stakeholders should consider in developing FLOSS governance for their organisation and the areas that strategy, policy and process statements should cover.

Although the purpose of effective FLOSS governance is to establish a practical, event-driven mechanism so as to enable an organisation to come to the right decisions on the range of particular questions that arise, this article does not itself address any of the granular technical FLOSS issues that continue to absorb significant amounts of management, technical and legal time. These issues include the multiplicity of FLOSS licenses; the ‘do’s and don’ts’ for licences and licence families themselves; and GPL-related issues as to what constitutes ‘distribution’ or the closeness of information communication triggering requirements on licensing of works of the organisation when combined with other works containing so called reciprocity or copyleft requirements.

2. Fundamentals of FLOSS Governance

(a) Objectives.

Embarking on the journey towards effective FLOSS governance can be a challenging process for any organisation. Starting out, it is critical to know the direction of travel: what are the organisation’s objectives for FLOSS and governance? As with other intellectual property based policies and governance, these can generally be succinctly stated around the high level aims of reducing/managing risk and maximising reward by:

- (i) avoiding disputes and managing regulatory risks;
- (ii) achieving good management/housekeeping for a financial event – for example, an investment round, IPO, trade sale, etc;
- (iii) customer satisfaction; and
- (iv) being a good FLOSS/corporate citizen.

(b) Key principles.

-
- http://www.palamida.com/themes/resources/Palamida_WhitePaper_PCComplianceAtRisk.pdf (Palamida);
 - <https://fossbazaar.org/openSourceGovernanceFundamentals> (White paper on FOSS Governance Fundamentals) and <https://fossbazaar.org/content/best-practices-open-source-governance> (Best Practices in Open Source Governance). See also the OLEX (OpenLogic Exchange) Wazi at <http://olex.openlogic.com/wazi/2009/create-open-source-policy/> (Best Practices for Creating an Open Source Policy) and <http://olex.openlogic.com/wazi/2009/create-an-open-source-governance-process/> (From Policy to Process: Best Practices for Creating an Open Source Governance Process); and <http://www.softwarefreedom.org/resources/2008/foss-primer.pdf> (a Legal Issues Primer for Open Source and Free Software Projects). In the published books, see in particular Meeker, *The Open Source Alternative*, Wiley, 2008, Chapters 10 (Developing a Corporate Open Source Policy) and 10A (Open Source Corporate Policy) and Woods/Guliani, *Open Source for the Enterprise*, O’Reilly, 2005, Chapter 7 (Designing an Open Source Strategy).

Supporting these objectives, the key principles of FLOSS governance may similarly be concisely articulated as:

- (i) **source reliability**: know where the FLOSS your organisation is using is coming from;
- (ii) **acquisition**: know what FLOSS your organisation is using;
- (iii) **tracking**: know what the FLOSS your organisation is using does and where it is being used and re-used;
- (iv) **roles and responsibilities**: know who is responsible for what; and
- (v) **licence compliance**: know that your organisation is complying with its FLOSS licence obligations.

(c) FLOSS governance is particular to each organisation.

Whilst the basic key FLOSS governance objectives and principles may easily be stated, applying them to any organisation moves quickly from the general to the particular. Effective FLOSS governance does not exist in a vacuum and needs to be anchored in the high level and the day to day – the strategic and the tactical – of the organisation and its operations.

(d) The range of organisations for which FLOSS governance is relevant.

On the one hand, if your organisation is engaged for example in internal use only of FLOSS – i.e. no re-distribution outside the organisation – the issues and so governance will differ from an organisation using FLOSS in the products or services that it markets. Equally, in the ‘internal use only’ case, the position of a public sector organisation – say a Government Department or Local Authority - will be different from the private sector as public sector organisations, in their drive to use public money wisely, may be encouraged or mandated to use FLOSS over proprietary solutions and may have more formal, even statutorily prescribed, procurement procedures which FLOSS governance will need to be consistent with.

On the other hand, if your organisation develops software using FLOSS and then distributes software with FLOSS components (whether as a service or as a licence), different and likely more complex issues will arise compared to the ‘internal use only’ case. In this ‘distribution’ case, emphasis is also likely to differ as a practical matter between a business to consumer (‘B2C’) organisation supplying FLOSS components within the software it commercialises for use by the consumer end user and a business to business (‘B2B’) organisation supplying FLOSS components within the software it commercialises for use by other businesses (and not the consumer end user).

Other factors relevant to the emphasis that FLOSS governance will take in any particular case include the geographical spread of the business(es) – a company with a number of development centres around the world will look at things differently from a company with all its developers under one roof; and product spread – to take an example from the communications industry, a manufacturer of devices with embedded software applications like mobile phones will be in a different position from a fixed or mobile operator principally supplying telecoms services rather than products (even if, as in the case of BT, the service may be delivered using a router containing embedded FLOSS applications as part of the service).

3. Contexts of FLOSS Governance – Building Blocks, Threads and Integration

It is helpful to think of the components of successful FLOSS governance as building blocks, linked or threaded together by context. These threads include ‘achievements to date’ and acquired FLOSS experience when embarking on FLOSS governance implementation; the people context; the strategic context; the policy context; and the process context. Each of these threads, and the individual building blocks within them, need then to be integrated across the organisation to take account of the context as a whole.

(a) Thread 1: FLOSS achievements to date

Each organisation at the stage where it is considering formalising FLOSS governance will almost certainly have arrived at a start point which likely has some notable FLOSS achievements to date – it might have shaped the core FLOSS issues it faces in its business and may already have done ad hoc work identifying the top FLOSS licences it uses in its business.

(b) Thread 2: the People context

FLOSS use in the organisation on anything other than a purely ad hoc basis will involve a number of stakeholder groups inside and outside organisation and effective FLOSS governance will depend on integration and cooperation between them in a way that is supportive and positive. There may well be many interested stakeholders whose interests will need intermediation in order to arrive at an agreed approach to governance. Table 1 below illustrates potential stakeholders in an organisation and summarises for each possible objectives in relation to FLOSS and how they may be achieved.

TABLE 1 – STAKEHOLDERS, THEIR FLOSS OBJECTIVES AND HOW THEY ARE ACHIEVED

STAKEHOLDER/GROUP	PRIME FLOSS OBJECTIVE	HOW PRIME FLOSS OBJECTIVE IS ACHIEVED
1 CEO/Leadership Team	Managing and ensuring effective use of FLOSS aligned with corporate strategy	Shaping and delivering best practice to achieve FLOSS governance
2 CFO/Finance Team	Organisation’s FLOSS benefits and risks identified, quantified and managed	FLOSS components and licences and other commitments (like other software assets) identified and recorded
3 CIO/Technical Team	Delivery of FLOSS components/developments on time and on budget; technical management of FLOSS governance programme	Implementing technical side of FLOSS governance (e.g. code indicator tool)
4 Contractors	See Customers, Developers Suppliers	See Customers, Developers Suppliers
5 Customers	Business advantage through use of Organisation’s technology/services with FLOSS risk managed	Performance of contractual commitments in Organisation/customer contracts
6 Developers	Knowledge that FLOSS use is encouraged & understand how he/she is able to use FLOSS in daily work	Follow FLOSS governance & feed back on possibilities for improvement
7 Directors/Supervisory Board	Organisation adopts appropriate FLOSS governance aligned to organisation’s strategy	Effective FLOSS governance properly implemented
8 FLOSS Compliance Officer (‘FLOSSCO’)	Developing, implementing and ensuring ongoing compliance with FLOSS governance	FLOSS strategy, policy and process statements articulated, agreed and implemented
9 FLOSS Working Party (‘FLOSSWP’)	Focal point for interests of organisation’s stakeholders; crucible for FLOSS governance	Manages FLOSSCO; communication back to other stakeholders
10 HR Team	To understand the HR and legal status to be given to FLOSS governance and Policy statements	FLOSS Policy statement to form part of the organisation’s employee/contractor handbook
11 Legal Team	Minimising legal risks maximising benefits to Organisation in its contractual commitments and FLOSS governance	Support other stakeholders in managing FLOSS governance, with particular emphasis on documents (statements, contracts, etc)
12 Sales & Marketing Team	Revenue generation/cost reduction, customer satisfaction	Risk of unauthorised FLOSS use managed
13 Shareholders	Shareholder value	Using FLOSS in an efficient, compliant way enables cost reduction, increase in profit, increased competitiveness, increased efficiencies and reduced IP leakage
14 Suppliers	Performance of contractual commitments in Organisation/supplier contracts	Compliance with Organisation’s inbound transactions/procurement policies for FLOSS

(c) Thread 3: the Strategy context

FLOSS governance does not live in a vacuum. At the highest level, it should align with other statements of organisational strategy – including corporate, risk management and IP strategy generally. The FLOSS Strategy statement is also the mechanism by which the internal consensus between the stakeholders is established and articulated. It is then a key point of reference for communication and education and for the development of the FLOSS Policy statement. The organisation’s leadership must be able to intermediate between the different groups and arrive at an agreed, short, clear, high level statement about where and why it will and will not use FLOSS. Table 2 illustrates pointers towards an organisational FLOSS Strategy.

TABLE 2 - Pointers towards a FLOSS Strategy statement for [Organisation]	
1.	[Organisation]’s FLOSS objectives. [Organisation] will continue to use FLOSS in order to increase [Organisation]’s: <ul style="list-style-type: none"> • ability to attract the best talent by building a development community at the forefront of FLOSS skills; • competitiveness by increasing development and operational efficiency and effectiveness, enabling faster time to market and reducing costs; and • value to stakeholders.
2.	FLOSS compliance. [Organisation] fully recognises and respects the rights of, and its agreements with, others just as it expects others to respect [Organisation]’s rights and perform their agreements with us. Accordingly, [Organisation] respects the need to ensure compliance with the terms of its legal obligations in licence agreements for FLOSS that it uses.
3.	FLOSS governance within [Organisation]: achieving the right balance. [Organisation] is committed to implementing best practice FLOSS governance. The purpose of [Organisation]’s best practice FLOSS governance is effectively, appropriately, proportionately and transparently to balance the objectives set out at paragraph 1 and the compliance expectation set out paragraph 2. This balance will be achieved: <p>within [Organisation]:</p> <ul style="list-style-type: none"> • by supporting [Organisation]’s development community in their work - as governance for developers by developers; • by effective communication, including educating, training and raising/maintaining awareness of FLOSS issues among all stakeholders; • by taking into account the interests of all stakeholders; and • through the active and timely support of all stakeholders; • with [Organisation]’s partners: • by ensuring that [Organisation]’s supplier and customer partners are aware of and comply with their FLOSS obligations, through [Organisation]’s contracts and appropriate relationship management.
4.	The mixed software environment. [Organisation]’s use of FLOSS will continue to be in a ‘mixed’ software environment: <ul style="list-style-type: none"> • using FLOSS and proprietary [Organisation]- and third party- owned software; • constantly evaluating where FLOSS is best used within [Organisation]; and • through re-use of FLOSS components where appropriate thereby leveraging [Organisation]’s knowledge and technical resources.
5.	Further details, etc. This Strategy statement forms part of [Organisation]’s FLOSS governance along with our Policy statement and Process statement. It is subject to review and change. For further details please contact [Organisation]’s FLOSS Compliance Officer at [email] and our FLOSS online resource kit at [intranet URL].

(d) Thread 4: the Policy context.

The heart of FLOSS governance is the FLOSS Policy statement. A well crafted Policy should:

- (i) be *clear and brief*, otherwise people will not read and understand it;
- (ii) be *event driven*, setting out roles and responsibilities as to who to go to and who does what in particular scenarios;
- (iii) set out *criteria and decision points* for FLOSS use: apply Occam’s razor – the simpler answer is usually right– and try to calibrate the Policy so it will settle 80% of decisions, while providing for effective exception management; and
- (iv) set out the *information* to be collected and tracked.

Table 3 below sets out pointers towards a FLOSS Policy around three main headings – scope and rationale; roles, responsibilities, training and awareness; and by transaction type – inbound, in house and outbound. Commentary on a number of the more difficult issues in practice is provided by way of footnote.

TABLE 3 - Pointers towards a FLOSS Policy statement for [Organisation]	
Scope and rationale	
1. Scope	<ul style="list-style-type: none"> • Purpose: This Policy statement is designed to supplement our existing policy and processes relating to [Organisation]’s products and services. It deals specifically with development and licensing considerations that must be fully understood and complied with when using and otherwise dealing with FLOSS within products and services that [Organisation] [markets][uses]; • Who does this Policy statement apply to?⁵ This Policy statement is mandatory and applies to everybody in [Organisation] who is responsible for [product design, launch and support across all [Organisation]’s solutions, whether as an employee or contractor. The intention is to ensure that [Organisation] understands, complies with, and is seen to understand and comply fully with the obligations and duties as contained in the relevant FLOSS licence terms; • What is the legal status of this Policy statement?⁶ This Policy statement [forms part of [Organisation]’s HR handbook (for employees) and part of the Contractor handbook (for corporate and individual contractors)] [has the same legal status as equivalent policy statements]; • Design process: All products and services that [Organisation] markets and that contain FLOSS must be [design approved by the [Organisation] review body] [(or other authorised body or process)], taking into account architectural, security, legal, commercial and all other relevant considerations. In particular, as part of that design approval FLOSS licence terms must be understood and processes put in place to ensure [Organisation] compliance once the product/service is launched; • Code indicator tool⁷: All source code in products and services that [Organisation] markets are to be scanned before launch using a FLOSS indicator tool. This will enable FLOSS code to be identified and all associated FLOSS licences to be checked for compliance with the relevant licence’s terms. Information from the scan must be acted upon so as to ensure [Organisation]’s compliance with the obligations in the relevant FLOSS licences; • Further details, etc: This Policy statement forms part of [Organisation]’s FLOSS governance along with our strategy statement and Process statement. It is subject to review and change. For further details please contact [Organisation]’s FLOSS Compliance Officer at [email] and our FLOSS online resource kit at [intranet URL].
2. Rationale	<ul style="list-style-type: none"> • The rationale behind this part of the Policy statement is to provide an introduction to FLOSS models and ensure FLOSS licences are given the attention and respect they require as a legal document; • The licensing of FLOSS code follows a different style of business model to the type [Organisation] has historically been used to. Most proprietary software is licensed under what can be called a Proprietary Model, where the copyright owner reserves all the rights the law grants, except for certain specific rights

⁵The HR aspects of the Policy are particularly important in considering how the organisation will ensure that FLOSS governance is effective. If it already has IT (email use for example) or intellectual property policies that are incorporated expressly or by reference into the HR handbook or even the contract of employment, it will be relatively straightforward to treat FLOSS governance similarly. If there is nothing comparable already in place, a number of questions need to be addressed, including particularly consequences of non-compliance where a developer uses FLOSS otherwise than in accordance with the FLOSS Policy or contributes to a FLOSS project otherwise than as permitted.

⁶HR difficulties can be compounded by the tension that generally arises between copyright law (where copyright in software developed by an employee in the course of his or her employment generally vests in the employer by operation of law) and code contributions to FLOSS projects (which generally provide that copyright in code contributed to the project is owned by the project). Again, corporate policy needs to be thought through and articulated in advance here. To complete the picture, it is worth remembering that under English law for example software developed by a contractor – whether an individual or a corporation – needs to be expressly assigned in order to belong to the organisation engaging the contractor. This requirement arises as a result of Section 11 of the UK Copyright Designs and Patents Act 1988, which provides that the individual who writes the software is the first copyright owner (S.11(1)) except where that individual is an employee writing software in the course of his or her employment, the employer is the first copyright owner in the absence of agreement to the contrary (S.11(2)).

⁷The products of specialist FLOSS service providers like Black Duck, Palamida and Fossology and the code indicator tools and other technology platforms they supply can automate and take significant cost out of manual processes. See also Table 4, Part B below (Processes).

	<p>which are granted for a licence fee (e.g. for £10 I license you (i.e. grant you permission) to use, but not to copy, modify or publish etc the software);</p> <ul style="list-style-type: none"> • FLOSS code on the other hand is in the main licensed either under: <ul style="list-style-type: none"> ◦ an Academic Model - such as the BSD, MIT, AFL, Apache licenses. Academic FLOSS Licences are typically light-touch agreements that basically seek “Freedom” for the software code. The main positive obligation on the Licensee is the duty to identify the origins of the FLOSS code – “attribution”; or ◦ a Reciprocal Model - such as the GPL, MPL, CPL and EPL. Reciprocal FLOSS Licences are generally more assertive in putting positive obligations on the Licensee with the objective of ensuring that all the copyright owner’s rights (use, copy, modify, publish etc) are passed down to other users; • [Organisation] will continue to operate in a ‘mixed’ software environment, using proprietary software under the Proprietary Model and (for FLOSS) the Academic Model and the Reciprocal Model; • Regardless of the underlying model, every software licence that attaches to software code (whether proprietary or FLOSS) constitutes a legal agreement between the licensor and the licensee. [Organisation] will comply fully with its legal obligations as set out in any licence agreement attaching to software code that is used within [Organisation], including used within [Organisation] products or services. <p>Roles, Responsibilities, Training and Awareness</p> <p>3. Roles and responsibilities</p> <ul style="list-style-type: none"> • FLOSS Compliance Officer⁸: In order to help [Organisation] achieve its FLOSS objectives, [Organisation] has created the position of FLOSS Compliance Officer (‘FLOSSCO’). FLOSSCO will be the first line of support for the development community within [Organisation] on questions you may have around FLOSS; • FLOSS Working Party:⁹ FLOSSCO will report to the FLOSS Working Party (‘FLOSSWP’). The FLOSSWP has members drawn from [Organisation]’s stakeholders. The role of the FLOSSWP is to give guidance to the FLOSSCO and, reporting to [—], to ensure that [Organisation]’s use of FLOSS is aligned to [Organisation]’s strategy and the FLOSS Strategy Statement. <p>4. Training and awareness¹⁰. FLOSSCO and the FLOSSWP will organise and carry out regular and frequent FLOSS training and awareness to ensure that the principles of [Organisation]’s FLOSS strategy and policy are understood and met throughout [Organisation].</p> <p>FLOSS Policy in inbound transactions, in house development and outbound transactions¹¹</p> <p>5. FLOSS Policy on inbound transactions</p> <ul style="list-style-type: none"> • FLOSS in [Organisation]’s procurement policies <ul style="list-style-type: none"> • Pre-contractual documents (RFIs, RFPs, etc) and contracts are to provide that software deliverables to [Organisation] are not to contain FLOSS unless FLOSS components individually identified before contract signature and expressly agreed by [Organisation]; • [Organisation]’s procurement contracts to reserve the right for [Organisation] to apply code indicator tool to carry out assessment in any case; • [Organisation]’s procurement contracts to include warranty/indemnity protection for non-identified/agreed FLOSS and (in addition to normal remedies) to provide for rewriting as remediation on case by case basis; • FLOSS in inbound development agreements: as per procurement policies outlined above; • FLOSS in M&A: <ul style="list-style-type: none"> • Technical and legal due diligence to be configured to enable all FLOSS in target company’s code base to be identified early on; • Consider using code indicator tool provider on escrow basis to carry out independent assessment; • Allow sufficient time for remediation by rewriting between signature of contracts and closing/completion; • FLOSSCO and [legal representative of [Organisation]] will be available to discuss particular issues arising on inbound transactions.
--	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

⁸ The FLOSSCO and the FLOSSWP are the lynchpins of the FLOSS governance process. The FLOSSCO is generally drawn from the development or technical rather than Legal team in practice, with Legal team representation on the working party.

⁹ See previous footnote.

¹⁰ An effective, continuing communication, training and awareness programme is of the essence of good FLOSS governance.

¹¹ The FLOSS policy should be event driven – i.e. it needs to think through and define in advance the sorts of issues that will arise. It should then aim to prescribe decision making which will deal with 80% of the issues that arise, with effective escalation to deal promptly with the other 20%. The events in this illustration are defined by reference to inbound, in-house and outbound transactions.

6.	FLOSS Policy on in house development
•	outline of authorisation mechanism:
•	FLOSS governance will operate across the organisation on the basis of pre-approved FLOSS components/software and the FLOSS licences that attach to them;
•	Assessments through indicator tool: [Organisation] will:
•	assess what FLOSS it [and its contractors] [is/are] using in [its/their] operations; and
•	associate the relevant FLOSS licences with the FLOSS so assessed to be used;
by:	
•	assessing ‘incoming’ code using the code indicator tool;
•	pre-launch/release code assessments; and
•	carrying out periodical assessments of internally developed code to verify that the FLOSS being used within [Organisation] is what is expected to be used;
•	Remediation where necessary: [Organisation] will develop a process to review, assess and remediate instances of non-compliance with [Organisation]’s Policy statement or otherwise in relation to a particular FLOSS licence;
•	FLOSS licence approval:
•	approval will be on the basis of the FLOSS licences determined to be most commonly used within [Organisation];
•	Approval ‘do’s and don’ts’: approval will be to enable use of the software concerned on the basis of clear, short, simple ‘do’s and don’ts’ addressing the needs of [Organisation] developers;
•	Scope of approval: Unapproved open source software, software licensed on an unapproved licence, or use outside the ‘do’s and don’ts’ will be prohibited;
•	Post-implementation approval: The post-implementation approval process will involve the FLOSSCO and will be designed to support the development community in giving timely positive assistance whilst respecting open source licence obligations.
7.	[Organisation]’s Policy on contributions to FLOSS projects. [set out here whether and if so to what FLOSS projects and on what terms [Organisation] developers may contribute code and other work] ¹² .
8.	FLOSS Policy on outbound transactions
o	[Organisation]’s template [licence/services agreements] set out [Organisation]’s approach to FLOSS in its customer contracts;
o	FLOSSCO and [legal representative] of [Organisation] will be available to discuss particular issues arising on outbound transactions.

(e) Thread 5: the Process context

The FLOSS processes should take the strain of FLOSS governance. The process context is where the interrelationships with and dependencies on policies outside the FLOSS area and other building blocks and threads within it need to integrate. These are illustrated at Part A of Table 4 below.

Pre-implementation (see Part B of Table 4), the project needs to be treated like any other development project in the organisation, with proper resource allocation, planning, mapping and timetabling. Consider using a pilot in one part of the business to gain experience that can than be rolled out across the organisation as a whole. Consider an amnesty to get the development community onside – this is the ‘hearts and minds’ time. As a practical matter, the importance of technology platforms to take out time and cost, add efficiency enhance collaboration, improve record keeping and ensure validation can scarcely be over emphasised.

Part C of Table 4 sets out consideration that the organisation will need to address on implementation. The FLOSS governance processes will need to be supple enough to cater for the range of activities post-implementation, and these are summarised at Part D of Table 4.

TABLE 4 - CHECKLIST FOR FLOSS PROCESS statement for [Organisation]	
A.	Dependencies
1.	Dependencies on/links with:

¹² See footnote 8.

	<ul style="list-style-type: none"> • FLOSS governance Strategy and Policy statements; • [Organisation] patents and other IPR policies; • Relevant stakeholder groups – e.g. architecture group, etc; • Source code management (CVS, subversion, etc); • HR policies; • Inbound/outbound contract groups; • Exit strategy (if applicable).
B.	Pre-Implementation
2.	Project planning, road mapping, timetabling. Treat implementation of FLOSS governance at the process level like any other development project – with sufficient/appropriate resources, and detailed project planning, road mapping, dependency management and timetabling.
3.	Indicator tool implementation. Consider procurement of and budget implications for indicator tool well in advance of FLOSS governance implementation.
4.	Initial assessment. Consider initial code assessment (NB: make sure you can continue to use the assessment results even after the contract with the indicator tool provider has terminated).
5.	Consider amnesty for developers pre-implementation to encourage/bolster need for compliance use post-implementation. For example, where a developer had perhaps mistakenly used FLOSS otherwise than in strict compliance with the relevant licence terms but, on realising the mistake, had not informed the his or her manager, the ‘amnesty’ would seek to encourage full reporting of the mistake within the Organisation before a certain date without fear of adverse consequences – on a sort of ‘confess and be forgiven’ basis. This would enable the Organisation to have a full picture and a firm foundation from which to assess and if necessary remediate.
6.	Consider pilot project implementation of the processes initially before roll out across [Organisation] in order to gain experience about what works best.
C.	Implementation
7.	Approval for FLOSS licences most commonly used.
o	Identify [Organisation]’s ‘top [X]’ FLOSS licences most commonly used within [Organisation], e.g.: [list];
o	Refer [intranet hyperlink] for methodology of how these FLOSS licences have been identified and analysis;
8.	Approval ‘do’s and don’ts’
o	Consider approval on the basis of short form, easily accessible/readable ‘Do’s and Don’ts’;
o	Consider maintaining intranet URL to show FLOSS [components] whose licences have been approved;
o	Consider maintaining separate intranet URL to show FLOSS licences that are approved for use;
o	Consider maintaining separate intranet URL of FLOSS components/licences (if any) whose use always requires prior specific approval from FOSSCO/legal;
9.	Pre-launch/release compliance check using code indicator tool or otherwise.
10.	Set out service levels for FOSSCO/FOSSWG responses to individual questions outside scope of policy/process guidance.
A.	Post Implementation
11.	Arrangements for code and other information repository.
12.	Periodical code assessment.
13.	Remediation where necessary.
14.	Training and awareness.

4. Conclusion

As FLOSS use in the organisation approaches ubiquity, FLOSS governance is rapidly becoming a ‘must have’ not just a ‘nice to have’ in order to manage risk and benefit effectively. Each organisation’s needs will be different, and senior management will need to consider all aspects of this complex question carefully before embarking on FLOSS governance implementation, as they would in any sophisticated software development project. At the end of the journey, management is looking to have in place integrated processes across all relevant business functions to manage effective use of FLOSS throughout the organisation. To get there, it should consider disassembling the various pieces into their building block components and threading them together by start point (achievements to date), people (stakeholders) and the strategic, policy and process aspects.

About the author

Richard Kemp is one of the UK's top technology and communications lawyers. He has a particular reputation for developing commercial and innovative legal solutions to the business challenges of technology development, application, implementation and regulation.

Richard is ranked in the global top 10 in both the IT and telecoms fields in the Expert Guide to the World's Leading Lawyers - Best of the Best 2008. He has been listed in the Best of the Best series since 2000 and in the top band of UK technology lawyers in the UK legal directories since setting up Kemp & Co in 1997. Richard has participated in teams that have won numerous legal industry awards and sits on the boards and editorial panels of a number of technology law specialist groups.

Richard qualified as a solicitor with Clifford-Turner in 1980 after graduating from Cambridge (1977) and from the ULB in Brussels (1979). He established Kemp & Co in 1997, which became Kemp Little LLP, the first UK law firm LLP, in 2001.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Kemp, Richard (2009) 'Towards Free/Libre Open Source Software Governance in the Organisation', *IFOSS L. Rev.*, 1(2), pp 61 – 72
DOI: [10.5033/ifosslr.v1i2.19](https://doi.org/10.5033/ifosslr.v1i2.19)

Copyright © 2009 Richard Kemp.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Balancing free with IP: if open source solutions become de facto standards, could competition law start to bite?

Susannah Sheppard^a

*(a) Partner and Head of Competition & Regulatory, Kemp
Little LLP*

DOI: [10.5033/ifoossr.v1i2.16](https://doi.org/10.5033/ifoossr.v1i2.16)

Abstract

Now mainstream within the software market, Open Source (“OS”), which began as an ideological pursuit to create greater freedom to innovate without concern for the constraints of proprietary licences, has today achieved such a key role in software development that it may begin to enter a level of maturity in which competition law risks could arise.

It is of interest to consider how, theoretically, competition law might apply to some of the OS licensing terms in situations where the commercial contributors to code have either individually, collectively or as a result of the network effects of the OS community attained a dominant market position in a relevant market, access to which the OS code is necessary. If were to this occur, some difficult competition law questions may emerge from the success of OS and as an unintended consequence of its principally pro-competitive and efficiency enhancing objective

Keywords

Competition law; Free and Open Source Software; Licensing; Article 101 TFEU; Article 102 TFEU

Info

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

Introduction

Now mainstream within the software market, Open Source (“OS”), which began as an ideological pursuit to create greater freedom to innovate without concern for the constraints of proprietary licences, has today achieved such a key role in software development that it may begin to enter a level of maturity in which competition law risks could arise. This is only likely to be the case in limited circumstances and probably only as a result of a potential arguments concerning anti-competitive ‘effects’ rather than any anti-competitive object. Nevertheless, it is of interest to consider how, theoretically, competition law might apply to some of the OS licensing terms in situations where the commercial contributors to code have either individually, collectively or as a

result of the network effects of the OS community attained a dominant market position in a relevant market, access to which the OS code is necessary. If this were to occur, some difficult competition law questions may emerge from the success of OS and as an unintended consequence of its principally pro-competitive and efficiency enhancing objective.

An area of particular interest when considering possible competition law claims arises in relation to two issues: firstly, from one of OS's basic principles, namely, that whenever included in software that is published or distributed, source code must be freely available to all, without payment, and secondly from the 'copyleft' principles of the GNU General Public License (GPL)¹.

Copyleft is the idea that the freedoms guaranteed by the GPL (including the use of the source code without payment) would also apply to new works 'derived from' or 'containing' (in the language of Article 2(b) of version 2 of the GPL) the original GPL-licensed software. In contrast to the traditional role of copyright, which grants exclusive rights to do certain 'restricted acts' (in the language of UK copyright law) to the author of original software copyleft gives the user certain rights of access to the source code without payment (other than to cover the cost of copying).

This aspect of the GPL has already been the subject of a US antitrust challenge. In the US Court of Appeal case *Wallace v International Business Machines Corporation and Others* (2006) ("*Wallace v IBMC and Others*")² the claim that the GPL was either a conspiracy or a form of price fixing by the open source community was firmly rejected. However, the case did not touch on issues of dominance or market foreclosure. It focused on conspiracy, predatory pricing and price fixing, and the judgment states that:

“Wallace does not contend that Linux has shown a large market share, or poses such a threat to consumers’ welfare in the long run that evaluation under the Rule of Reason could lead to condemnation”.

But what if it *could* be contended that an OS Solution had garnered a near monopoly market position and either collectively or individually the terms of participation posed a threat to consumer welfare?

This article discusses the possible claim that if an OS solution or platform were to become dominant in a relevant market, the terms of open source licenses (requiring redistribution free of charge) and the copyleft provisions of the GPL might be found to be exclusionary or unfair to a non-OS operator with a proprietary copyright or patent (hereafter referred to as an “IP right-holder”). This claim would be made on the basis that long term consumer harm would arise if an IP right-holder were excluded in practice from participating in the OS solution on commercially viable terms, which in turn would lead to the stifling or reduction of incentives to innovate and the removal of market choice.

There would be two separate legal bases of the competition claims: one involves the question of whether royalty-free licensing of IP rights included in an open source solution constitutes fair, reasonable and non-discriminatory ('(F)RAND') terms or whether, in some circumstances, royalty-free licensing is unfair to the IP right-holder; the second is whether, notwithstanding those arguments, the copyleft provisions of GPL also prejudice the IP right-holder who is not only faced with the loss of royalties but also the obligation to license forward its own rights for free.

¹ <http://www.gnu.org/copyleft/gpl.html>

² *Wallace v. IBM*, 467 F.3d 1104 (7th Cir. 2006). Available at: http://www.internetcases.com/library/cases/2006-11-09_wallace_v_ibm.pdf

It is, of course, recognised that there is no obligation for an economic operator to participate in open source licensing. However, if an IP right holder can only enter a market by engaging with the OS community and in so doing must waive all royalty claims to any IP rights it holds necessary to participate in the software, competition law arguments could arise if this were to result in either market foreclosure or unfair terms of participation.

In the same way that OS proponents have argued that proprietary, royalty bearing technology can exclude them from a technical market, so too could the reverse be true. If OS solutions became dominant in a given sector, the requirement to sign up to certain license terms, particularly where these involve royalty-free licensing could exclude, discriminate against or undermine the commercial viability of the proprietary right owner. There are two principles of EU competition law that should be assessed in this situation, Article 101³ (formerly Article 81 EC) and 102 (formerly Article 82 EC) of the EU Treaty. I set out below the basics of these Articles for those unfamiliar with EU competition law.

Article 101(1) prohibits agreements which prevent, restrict or distort competition within the Common Market and which affect trade between Member States, unless they are capable of exemption under Article 101(3). Exemption is granted where the agreement contributes to improving the production or distribution of goods or to promoting technical or economic progress, whilst allowing consumers a fair share of the resulting benefit. However, the restriction in question must not be indispensable to the attainment of the objectives in question and not substantially eliminate competition for the products in question.

Article 102 prohibits the abuse by one or more underleasings of a dominant position within the Common Market or in a substantial part of it which affects trade between Member States. Abuses can include imposing unfair or discriminatory terms, tying, bundling or exclusionary behaviour.

Article 101 Arguments

A network of OS or licensing agreements between existing OS participants might support an Article 101 claim, albeit probably based on a theory of anti-competitive effect rather than any object breach, as was contended in *Wallace v IBMC* and others. The Article 101 claim would be based on the premise that, as a result of license agreements between the OS participants, namely that involvement in distribution of the OS software could only be based on royalty-free licensing, the IP right-holder would have a choice either to not participate in the markets covered by the OS licensing structure at all or to do so on terms that were not commercially viable to the IP right-holder. This would be even more potent in the case of a GPL licence if, in addition, the IP rightholder had to comply with copyleft principles. If the network of license agreements gave rise to a solution with a high market share (say over 70%), by not participating the IP right-holder would be effectively unable to operate on the market in question. The practical effects of the OS license network would be akin to a group boycott by the OS community of a proprietary right holder from the market. The counter argument that the IP right-holder has a choice is meaningless if, in practice, there is no real choice but market exclusion. That said, OS license proponents might argue that the efficiencies provided by OS give rise to conditions which merit an Article 101(3) exemption. However, this would be a matter of fact to be determined in the circumstances of each case, and one cannot assume that OS solutions should automatically benefit from an

3 Dominance is not a requirements of the Article 101 prohibition, however, unless the OS solutions (rather than the individual participants) were dominant in a market, the exclusionary effect of the network of OS licensing agreements would be unlikely to be significant, because IP rightholders could decide not to participate without any major market impact.

exemption.

Article 102 Arguments

The Article 102 claim would be similar in terms of its economic effect, but be based on the exclusion arising either from individual or collective dominance of the existing OS participants. One can imagine a number of heads of ‘abuse’, unfair licensing terms (if royalty-free licensing was not commercially viable to the IP right-holder or in the case of the GPL where access was conditional upon licensing IP rights that it did not wish to give up royalty-free), discriminatory behaviour (on the basis that a proprietary owner was being treated in the same way as non-proprietary owner when they were in materially different positions), constructive refusal to supply (on the basis that the terms of participation would not be commercially viable) and even possibly predatory pricing (although given the low variable costs involved in software licensing this may be particularly difficult to substantiate).

This article focuses specifically on the arguments surrounding unfair and/or exclusionary terms of access to a market, by looking at the way in which this issue has previously been assessed in the context of standard setting. Again, efficiency arguments might arise on the basis that that royalty-free licensing or copyleft provisions were objectively justifiable to create the network in the first place. However, these would also be issues to look at on the basis of the facts of a particular case. There is no reason to conclude that, as a matter of principle, they always apply when OS licensing is involved.

The risk of dominance or network effects creating de facto standards that govern the market.

The risk that a commercial operator (or operators) who have been instrumental in developing an OS solution might become individually or collectively dominant in a market is not entirely theoretical because the network effects arising from new technologies can sometimes rapidly lead to high market shares. In such circumstances the need for interoperability with the technology can mean a software solution becoming so widely adopted within a given market that it becomes a form of *de facto* standard. Some of the competition law issues that may arise in these circumstances are analogous to those that have been considered in the context of standard setting involving high technology sectors.

In many cases, competition issues would not come into play as there are a number of ways in which IP right owners can seek to avoid the impact of the copyleft principles by means of technical mechanisms such as ‘shims’ and ‘APIs’⁴. This article does not consider those technical mechanisms and assumes that competition arguments would be called into question if the IP right-holder were unable to deploy such mechanisms successfully as a means of protecting its rights or could not find another commercial solution to monetise them.

Background to EU Standard setting in the technology sector

Competition law issues that could arise have similarities with those encountered at the beginning of the 1990s in the EU (or the EC as it then was) within the auspices of the European

4 The author recognises that the OS Community continues to debate whether these mechanisms are effective.

Telecommunications Standards Institute ('ETSI'). At the time, the economic fault lines that the European Commission sought to overcome were primarily based on national barriers to trade. However, with the burgeoning telecoms market and in particular the growth in mobile phones, technical trade barriers were superimposed upon national market fragmentation. ETSI was established to forge a solution to this problem. There were three ways in which the standards created by ETSI (European Telecommunications Standard: 'ETS') would have such an influence on the market. Firstly, in some cases ETSs would provide a presumption of conformity with the 'essential requirements' of the so-called 'New Approach' Directives, such as the Telecommunications Terminal Equipment Directive (Directive 1999/5/EC)⁵. Secondly, where a single technical solution was fundamental to intra-Community interconnectivity, the ETS formed part of a 'Common Technical Regulation' ('CTR') which was mandatory. Thirdly, under public procurement rules, technical specifications should, preferably, be defined by reference to national standards transposing European standards. Hence, if an ETS contained a specification which was covered by an IP right, the right-holder had a captive market: (1) if the ETS formed part of a CTR, every operator in the market was required to use it; or (2) if the ETS was the basis of a presumption of conformity with an 'essential requirement', although its use was voluntary, most market operators would adopt the ETS as the simplest means of demonstrating regulatory compliance. Furthermore, it would also be part of a preferred specification for the purpose of supplying public contracts.

There became a clear need to establish ground rules as to how IP could be both incorporated in a standard and then licensed. The ETSI IPR Policy was created to set a framework in which to require disclosure of IPR and agree, up front, that IPRs would be licensed to all on fair, reasonable and non-discriminatory terms. ETSI's IPR policy objectives state that:

"the ETSI IPR policy seeks a balance between the needs of standardisation for public use in the field of telecommunications and the rights of the owners of IPRs."⁶

Since an IP right-holder is required to disclose essential IP and is then given 3 months to provide an undertaking to license its IP on ETSI terms, it is implicit in the ETSI IPR policy that an IP owner has the right not to contribute its technology to the standard if it does not wish to do so.

Having established these basic principles, some of the most contentious issues to be assessed included determining whether IP is essential, the point at which IP disclosure should take place and the meaning of FRAND. Even within ETSI where there is a relatively comprehensive policy in place, such matters are far from certain. Moreover, ETSI is but one of a number of standards setting organisations ('SSOs') and much standards setting activity takes place outside formal bodies, in private standard setting groups or in trade associations. In these other types of private standards bodies there is often little experience of how to handle the problems of reconciling the conflicting pressure of IP, standards and competition requirements.

The recent investigations by the European Commission into Rambus⁷ and Qualcomm⁸ are

5 OJ L 91, 7.4.1999, pp. 10–28, available at: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31999L0005:EN:HTML>

6 <http://www.etsi.org/WebSite/AboutETSI/IPRsInETSI/IPRsInETSI.aspx>

7 Reference: MEMO/07/330 Date: 23/08/2007, available at: <http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/07/330&format=HTML&aged=0&language=EN&guiLanguage=en>
Reference: IP/09/1897 Date: 09/12/2009, available at: <http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/1897&format=HTML&aged=0&language=EN&guiLanguage=en>

8 Reference: MEMO/07/389 Date: 01/10/2007, available at: <http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/07/389&format=HTML&aged=1&language=EN&guiLanguage=en>
MEMO/09/516 Date: 24/11/2009, available at: <http://europa.eu/rapid/pressReleasesAction.do?reference=MEMO/09/516&format=HTML&aged=0&language=EN&guiLanguage=en>

evidence of how some of these issues can be problematic.

Standards and Open Source

As well as the questions currently being looked at by the Commission in the context of the Qualcomm and Rambus investigations, other potential Article 102 abuses and Article 101 breaches can arise in standards setting and in the use of standards. Just as a an IP right-holder can gain market power by having its essential IPR included within a standard, so SSOs (either formal or informal) could be used by the group to exert pressure on the IP right-holder in order to extort cheap licences. This could be done either by collective agreement or if a *de facto* standard was controlled by an individually dominant undertaking or collectively dominant group on a relevant market. It is, of course, recognised that, a right holder has a choice not to participate in standard setting, but if the standard will, in practice, control access to the market, this choice is of little practical value. As mentioned earlier in this article, such behaviour could give rise to two different competition arguments; a breach of Article 101 on the basis that the terms of access arising as a result of the network of licensing agreements are exclusionary and are akin to a group boycott; alternatively, if it could be demonstrated that there was dominance on a relevant market (which maybe the same as or adjacent to the new derivative software market) it might also be argued that such behaviour is an abuse of Article 102.

The ETSI IPR policy that was eventually adopted after a European Commission investigation into its proposed treatment of IPR following a complaint brought by CBEMA⁹ gives some support to the view that the Commission recognised that rightholders need to be protected as much as the open standards community. Originally, compulsory licensing of IP rights had been one of the options proposed. The eventual adoption of a policy in which IP rightholders could decide not to contribute IPR¹⁰ or if they did so would receive FRAND terms for its inclusion, demonstrates that the balance that was struck protected the interests of the right holders as well as those of the users. Furthermore, it was established in the US, in the Federal Trade Commission's Decision on Sanitary Engineering (1986)¹¹ that it was an abuse for The American Society of Sanitary Engineering ("ASSE"), to adopt a policy of refusing to develop a standard for a product which is patented or manufactured by only one manufacturer, regardless of the merits. The ASSE decision therefore implies that a right-holder cannot be arbitrarily excluded from participating in a standard technological specification, merely on the basis that others will have to pay for using the technology in question.

Accordingly, if the group were to recognise that a particular technology is a "must have" or even possibly just beneficial or advantageous and an IP right-holder of that technology would be excluded from operating in a relevant market unless it were to participate in the standard, the argument would be that the group position should not prevent the IP right-holder from setting a fair return for its investment. The requirement that IP rights be licensed on FRAND terms therefore, in some cases, may be necessary to protect the IP right-holder as much as the licensee: it ensures the owner of the right the ability to receive a reasonable return for the effort involved in

9 Computer and Business Equipment Manufacturers' Association.

10 Although early disclosure of IPR is required to prevent 'patent ambush': see the closure of the Commission's investigation into this issue in 2005 (Reference: IP/05/1565 Date: 12/12/2005, available at: <http://europa.eu/rapid/pressReleasesAction.do?reference=IP/05/1565&format=HTML&aged=0&language=EN&guiLanguage=en>)

11 American Society of Sanitary Engineering, Dkt. C-3169, 106 F.T.C. 324 (1985). The members of the ASSE include plumbing equipment manufacturers and designers, available at: [http://www.ftc.gov/os/decisions/docs/vol106/FTC_VOLUME_DECISION_106_\(JULY_-_DECEMBER_1985\)PAGES_291-360.pdf#page=34](http://www.ftc.gov/os/decisions/docs/vol106/FTC_VOLUME_DECISION_106_(JULY_-_DECEMBER_1985)PAGES_291-360.pdf#page=34) (at page 34).

creating the IP. This argument is supported by the general principle that underlies IP protection, namely, that innovation and investment of time and creative effort should be rewarded.

It is this type of behaviour which has the potential to impact the OS community and the IP rights that it may seek to encompass. This is because if an IP right-holder seeking access to a market which is dominated by an OS solution (either through the individual or collective dominance of the OS operators involved or as a result of a network of licence agreements) can only have access to that market by engaging with that open source community on their terms, then effectively it has no choice but to do so or be excluded. Of course, it can seek not to engage, but this is no answer if in fact, the technology covered by the OS licences represents the vast majority of the market in question. Therefore, an IP right-holder owner would either have to license its IP right royalty-free or not be able to enter the market at all.

The objectives behind the OS principles and the GPL are relevant to any potential consideration of competition law in this OS context. The principal objective of OS is undoubtedly efficiency improving and its agenda is far from anti-competitive. The insistence on royalty-free licensing is underpinned by the belief that licensing terms which require a royalty to be paid to the IP owner (beyond the mere recovery of costs of copying) are incompatible with the OS philosophy because they would limit the availability of the source code to all potential users. However, by adopting this approach there is a risk that, where OS developers look to other technologies to incorporate in the software to create new derivative products and the IP right-holder finds that market access is blocked other than by participating on the terms of the OS licence, the OS community may end up running into these types of competition argument. Those arguments are analogous with the ones used in the early debate surrounding the use of IP in ETSI standards.

Aside from the conspiracy and price fixing antitrust claims raised in *Wallace v IBMC and Others*, the question of whether it is discriminatory to impose free rather than (F)RAND licensing policies in technology standards has been publicly debated as a ‘principle’ rather than on a factual, market specific basis. Two opposing views have been aired- in the context of an Internet-based standards body in the United States, which incorporates OS principles as part of its licensing policies. The Overview on the definition of ‘Open Standards’ discussed on the Organization for the Advancement of Structured Information Standards (OASIS) website looks at this problem in the context of public procurement:

“CompTIA, BSA, ECITA, and congeners argue that government jurisdictions should not mandate the use of royalty-free IT standards in civil society projects because this discriminates against some business models which depend upon royalty revenues from RAND standards; open source advocates argue that government jurisdictions should not allow the adoption of royalty-bearing RAND-licensed IT standards in the public sector since the RAND licensing model (with a few minor exceptions) categorically discriminates against the [OSI] open source business model — as open source software development practices, open source licensing terms, and open source software distribution models are incompatible with RAND.”¹²

Translating this debate into EU public sector terms puts some context to a ‘market specific’ approach to the problem. Whilst an open competition for a public sector contract based on a particular specific technical specification does not, of itself, create a distinct market for the supply of that particular specification, the potential effect that public contracts can have on a market is sufficiently important that EU law has regulated to ensure a degree of open access. In the EU, under public procurement requirements, specifications must be based primarily on standards and,

¹² <http://xml.coverpages.org/openStandards.html>

wherever possible, the preference is for national standards that transpose European standards (such as ETSs). It follows from this that a proprietary standard should only ever be specified if there is no non-proprietary alternative. However, given the balance struck in the IP policies of the main European SSOs (such as ETSI), where standards are concerned, it is generally accepted that owners of IP rights contained within standards should be afforded FRAND terms for those rights.

In fact, OASIS handles these issues via its own IPR policy which enables participants in a technical committee to decide the policy applicable to IP rights involved in a standard, which can be RAND, royalty-free or involve non-assertion of rights. Participants are free to choose whether they want to participate on these terms¹³. This gives the proprietary right owner a chance to argue for RAND terms. However, in other settings, or where a de facto standard arises, this may not be the case.

Does IP involvement in Open Source “Standards” necessarily lead to a conclusion that OS ‘FRAND’ should always mean royalty-free?

The OS community argues that the price the IP right-holder pays for access to the OS code necessarily involves foregoing any royalty rights in IP contributions. This is a condition of entry; as the benefit of the OS network only arises as a result of the IP contributions of those before them, all should contribute on the same basis in order to enjoy the OS advantage. On this basis, it could be argued that it is only fair that FRAND terms mean that the IP right-holder has to license its rights royalty-free. However, the balance of advantages between the OS founders and a subsequent contributor may not always be equal. Whether the OS community’s arguments lead to a conclusion that the FRAND terms should be royalty-free might, particularly in cases where a de facto OS standard has been created, be a matter of fact to determine on a case by case basis, depending on the circumstances of the players involved, the role that the new derivative technology will play, and the market effect. OS royalty-free principles may well to prevail in most cases. However, it is possible to imagine a situation where the IP ‘sacrifices’ of the founding members had little commercial or economic significance and the network may have grown as a result of the many business models used to commercially exploit the OS. If you add to this a dominant position held by a key player within the OS community and the potential for market foreclosure in the absence of participation, a number of competition law arguments may come into play. An example of this could be where hardware is sold as a result of a downstream or upstream software market and the hardware suppliers will benefit from the new IP, but this particular IP software holder does not have an equivalent hardware business to off-set its IP investment for which it would obtain no royalties.

Will IP rightholders always be able to avoid Open Source and have alternative markets open to them?

Another argument from the OS community that could be used to defend their principles would be that the IP right-holder has the choice not to participate. This point is made in the context of the arguments raised in *Wallace v IBM Corp and Others* and probably applies in most cases. However, as previously mentioned, there might become areas of software development where OS becomes such an important platform that a decision not to participate leads, in effect, to an

¹³ The author does not know whether there have been any circumstances in which OASIS participants have felt that the group decision on IP Mode adopted has, in fact, resulted in the exclusion or coercion of certain participants, and whether, in practice, royalty-free is generally the preferred solution.

inability to enter or compete in a market. This would be particularly relevant if switching costs were high. Moreover, if there were any evidence that the OS licensing participants had developed an OS solution with a view to excluding a particular proprietary technology, this would add grist to the mill.

In other circumstances, there may be commercial pressure for participants in the existing OS community to bring a particular technology on board as it may be better to include a particular technology rather than waiting until an IP unencumbered one is available. This principle has already been acknowledged by one Internet Standards Body, the Internet Engineering Task Force ('IETF') which uses an OS licence to make available source code included in its standards. The IETF 'Guidelines for Working Groups on Intellectual Property Issues' states:

"It will always be better for the Internet to develop standards based on technology which can be used without concern about selective or costly licensing. However, increasingly, choosing a technology which is not impacted by IPR over an alternative that is may produce a weaker Internet. Sometimes there simply isn't any technology in an area that is not IPR-impacted. It is not always the wrong decision to select IPR-impacted technology, if the choice is made knowingly, after considering the alternatives and taking the IPR issues into account"¹⁴.

Conclusion

OS principles have, in many cases, overcome the conflicts between the rights of the IP right-holder with competition law and the economic imperatives arising from collective agreements, dominance and standardisation by imposing royalty-free terms. However, competition arguments may be applicable if OS solutions begin to dominate certain markets and rightholders find themselves with no commercial options available to monetise their IP investments, or excluded from markets. If applied in such a context it is not clear that EU competition authorities will always reach the same conclusion as the US Court of Appeal *Wallace v IBMC and Others* that "The GPL and open source software have nothing to fear from antitrust laws".

About the author

Susannah Sheppard is a partner and Head of Competition and Regulatory at Kemp Little LLP, advising leading technology firms on competition issues, with particular expertise in standardisation and the inter-relationship between IP and competition.

¹⁴ <http://tools.ietf.org/html/rfc3669>

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Sheppard, Susannah (2009) 'Balancing free with IP: if open source solutions become de facto standards, could competition law start to bite?', *IFOSS L. Rev.*, 1(2), pp 73 – 82

DOI: [10.5033/ifosslr.v1i2.16](https://doi.org/10.5033/ifosslr.v1i2.16)

Copyright © 2009 Susannah Sheppard.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Packaging Open Source

Mark Webbink^a

*(a) Visiting Professor of Law and Executive Director of the
Center for Patent Innovations at New York Law School.
With thanks to Daniel Hopkin for his invaluable assistance in
the preparation of this article*

DOI: <http://dx.doi.org/10.5033/ifoosslr.v1i2.26>

Abstract

Now with the increasing popularity of virtual computing environments we are observing the packaging of operating systems, including Linux, with applications to create software appliances, essentially applications that carry their operating system with them. With this rapidly expanding market opportunity, traditional proprietary software vendors are increasingly interested in the “rules of the road” for open source licensing and, in particular, for packaging Linux with an application into a software appliance. This paper is intended to provide background information for such application developers interested in creating software appliances utilizing open source components while ensuring both their open source license obligations as well as protection of their own copyrights and patents.

Keywords

Law; information technology; Free and Open Source Software

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

This article has been independently peer-reviewed.

Running proprietary (non-open source) software on an open source operating system, such as Linux, has become commonplace. Major proprietary software vendors such as IBM, Oracle, and Adobe have adapted their proprietary applications to run on top of a Linux operating system without concern for open source licensing issues. In addition, Linux vendors frequently ship proprietary applications with their Linux distributions, although the packaging practices and open source license compliance may vary from one to the next. Major device manufacturers, including Sony, Philips, Cisco, and Nokia, have utilized a Linux operating system in both their open (modifiable) and closed (non-modifiable) devices for some time, although their open source license compliance has also frequently required greater effort. Even with open source versions of JAVA we are seeing both open source and proprietary files combined in the same class libraries.

Now with the increasing popularity of virtual computing environments we are observing the packaging of operating systems, including Linux, with applications to create software appliances, essentially applications that carry their operating system with them. With this rapidly expanding market opportunity, traditional proprietary software vendors are increasingly interested in the “rules of the road” for open source licensing and, in particular, for packaging Linux with an application into a software appliance. This paper is intended to provide background information

for such application developers interested in creating software appliances utilizing open source components while ensuring both their open source license obligations as well as protection of their own copyrights and patents.

The paper will first review the copyright, patent, and key license provisions that arise in an open source context. The paper will address the more commonplace packaging of open source applications within mainstream Linux distributions and the license compliance issues that arise. Next, the paper will examine alternative packaging models, including embedded devices, JAVA class libraries, and software appliances. Finally, the paper will suggest some best practices for software application developers to maximize their value and to minimize their risks.

Copyrights and Patents in the Software Context

Under U.S. law any software will be automatically covered by copyright at the time it is developed unless the author expressly disclaims copyright protection or the code, for a variety of exceptional reasons, does not rise to the level of copyright protection. For our purposes, we will assume that most code is subject to copyright protection. Such copyright protection extends to both the source code and binary versions of the software, the latter being viewed much in the same light as a translation of a literary work.

There are three principal rights under copyright law that are held by the copyright holder in software: the right to copy the software, the right to modify the software, and the right to distribute the software. Under traditional proprietary models, the copyright holder elects not to share these rights with the party receiving a license in the software. Thus, under most proprietary software licensing models the end user has no right to copy the software (other than the statutory right to make a back-up copy of the software under 17 U.S. Code § 117). In addition, proprietary vendors commonly include restrictions on the decompiling or reverse engineering of the software (such restrictions being of limited scope with respect to such actions taken purely for the purpose of interoperability), thus barring the modification of the software. Finally, proprietary licenses will generally limit the redistribution of the software to the absolute transfer of the single copy covered by the license. As a consequence, packaging issues in the purely proprietary software context tend to be matters of negotiated contracts, not merely end user licenses.

By contrast, under open source licenses¹ copyright holders share these rights to copy, modify, and redistribute, sometimes without restriction and sometimes with limited restrictions as to obligations arising upon redistribution. As a consequence, the licensee under an open source license will enjoy rights to modify, combine, copy, and redistribute software that they would not typically enjoy under a proprietary license. In this context the open source licensee must then be aware of treatment under copyright law of modified or combined software code.

U.S. copyright law addresses such modifications and combinations as, respectively, derivative works and compilations, which includes collective works. The formal definitions for these terms may be found in 17 U.S. Code § 101:

A “derivative work” is a work based upon one or more preexisting works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other

¹ When referring to open source licenses in this paper, the author limits the reference to those licenses that have been certified as complying with the Open Source Definition as promulgated by the Open Source Initiative. That list of licenses may be found at <http://www.opensource.org>. The one exception to this convention is the inclusion of the GNU General Public License, version 3, as an open source license.

form in which a work may be recast, transformed, or adapted. A work consisting of editorial revisions, annotations, elaborations, or other modifications which, as a whole, represent an original work of authorship, is a “derivative work”.

A “collective work” is a work, such as a periodical issue, anthology, or encyclopedia, in which a number of contributions, constituting separate and independent works in themselves, are assembled into a collective whole.

A “compilation” is a work formed by the collection and assembling of preexisting materials or of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship. The term “compilation” includes collective works.

It is important to understand the distinctions between these three forms of works. The first distinction is between compilations and collective works. Note that all collective works are considered a form of compilation, but not all compilations are collective works. That said, the primary distinction between the two is that compilations may consist entirely of non-copyrightable materials, whereas a collective work will contain only copyrightable material. This can be better visualized in non-software terms. If I have factual information, say the names of all of the people who live in a city and their addresses, that information is not inherently protectible by copyright, i.e., there is nothing unique in the expression of that information – it is what it is. However, if I organize that same information in a unique way, say by first names, that particular organization may be protectible as a compilation even though the data within the compilation is not inherently protectible by copyright. Compilations may also exist in a combination of copyrightable material and non-copyrightable material, e.g., a book containing court opinions (which are public information and in the public domain) which are organized and annotated to provide more useful information to persons wanting to understand the decision.

By contrast, collective works consist of separate and independent copyrightable materials that have been organized into a single work. The typical non-software examples are periodicals, anthologies, or encyclopedias. Although collective works are considered a subset of compilations, they actually have as much, if not more, in common with derivative works than they do with non-collective work compilations in that both derivative and collective works are based upon pre-existing copyrightable works.

The key question is then where to draw the line between a work which is merely a collective work and a work that constitutes a derivative work. The treatise Nimmer on Copyright² identifies the following distinction:

A derivative work consists of a contribution of original material to a pre-existing work so as to recast, transform or adapt the pre-existing work. . . . A collective work will qualify for copyright by reason of the original effort expended in the process of compilation, even if no new matter is added.

In either case, the contributions required to produce a new work, whether a derivative or collective, must be more than minimal and meeting the standards for copyright protection.³ So what is the distinction between derivative works and collective works when it comes to software. The distinction largely arises in the number of underlying works involved. By definition a collective work consists of more than one underlying work while a derivative work consists of a

² Nimmer on Copyright, Lexis-Nexis 2010, release no. 72, §3.03[A]

³ For a more thorough discussion of what constitutes sufficient originality to qualify for copyright protection, see Originality Requirements Under U.S. and E.U. Copyright Law, © 2007 Software Freedom Law Center found here: <http://www.softwarefreedom.org/resources/>

single underlying work. But as with all such matters, there is a point at which this simple distinction no longer holds. When one underlying work is materially modified and so combined with a second work such that the combination of the two effectively takes on a singular identity in terms of use and perception, the combination has moved more closely to the definition of a derivative work. At the same time, a collective work may include one or more underlying works which are, themselves, derivative works of pre-existing works.

This distinction between derivative and collective works may more clearly be drawn in software when examining how two independent works relate to each other. Where two independent works are capable of sharing information, passing such information back and forth through published interfaces or by a temporary connection, such as a pipe, and not by a modification of one of the works by the other work, those works when combined in a single package would constitute a collective work but neither would likely be considered a derivative of the other. On the other hand, where one such application, when compiled at the same time as a second application, modifies the second application in such a way as to cause that second application to act in a unique manner with the first application, the combination would likely constitute a derivative work of the first program even though the two works, in their source code form, are separate and independent.

Finally, the copyright holder in the compilation holds a copyright only in the compiled work, not necessarily in any of its component parts (although it would be somewhat atypical for the copyright holder in a collective work to not have produced some new, separately copyrightable work, in producing the collective work). Thus, the copyright holder in a compilation or collective work must have permission from the copyright holder of each component part to include it in the compilation or collective work.

As we will see, some open source licenses address only derivative works and others address both derivative works and compilations/collective works. And, of course, whether a new work, which utilizes only some but not all of a pre-existing work, is a derivative work is a matter of statutory interpretation and case law. For our purposes in this paper, we will consider all works that incorporate some or all of the code of a pre-existing work to be derivative works.

At the same time, software code developed in or imported into the U.S. may be subject to patent protection under U.S. law. Such patents coverage is not defined by specific source or object code but, rather, by the claims set forth in the patent. Whether specific source or object code infringes a given patent depends on whether the structure and operation of that source code (i.e., the methods it employs) reads on the claims of the patent.

Such so-called “software patents” do not distinguish between code licensing models. As a consequence, both open source and proprietary software licensing models need to be concerned with software patents. The manner in which such licenses address the subject of software patents varies considerably. In this paper we will only concern ourselves with the manner in which various open source licenses address software patents.

Open Source Licenses – Derivative Works, Collective Works and Patents

The Open Source Initiative lists 60 licenses that meet the OSI definition of open source. Although large in number, the differences among many of these licenses are relatively small. In fact, some are virtually identical but for their names, e.g., the Common Public License and the IBM Public License. All of these licenses cover the source code, and all permit the licensee to copy, modify,

and redistribute the code.

The simplest group of licenses are those that permit the exercise of the rights to copy, modify, and redistribute without limitation. These include the widely-used Berkeley Software Distribution (BSD) license, MIT license, and Apache Software License. Source code, and corresponding binaries, of software licensed under these licenses may be readily incorporated with other open source code or even into proprietary code without concern over license compatibility. That is the upside. The downside is that, in their brevity, these licenses do not address patents in any manner. One solution to this problem, as advanced by Intel, has been to combine the BSD with an express patent license, found here: <http://infiniband.sourceforge.net/duallicense.htm>.

Of the open source licenses that incorporate a restriction on redistribution, the most widely used are the GNU General Public License (both version 2 and version 3), the GNU Lesser General Public License (version 2), the Apache License version 2, Artistic License version 2, Common Public License, and Mozilla Public License version 1.1. Let us consider how each of these licenses addresses issues of derivative works, collective works, and patents. For our purposes we will consider any mere compilation that does not rise to the level of a collective work to be a non-issue.

GNU General Public License, version 2

("GPLv2" - <http://www.opensource.org/licenses/gpl-license.php>)

As with other open source licenses, GPLv2 permits licensees to make modifications of the work and to redistribute the work, either in its original form or as modified (GPLv2 refers to such derivative works as "works based on the Program") so long as the distributed work continues to be covered by this license (section 2 of the license). That aspect of GPLv2 is fairly well understood. But GPLv2 does not only address derivative works; it also addresses compilations in the form of collective works (which it refers to as "mere aggregations"). For compilations, or mere aggregations, GPLv2 does not apply or place limitations on the licenses pertinent to non-GPLv2 works included in the compilation. The same is not true for collective works. With respect to collective works, GPLv2 states in Section 2:

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

This language has been commonly misunderstood to mean that all works included in such a collective work are re-licensed under GPLv2, thus overriding their original license. That is not the case for two reasons: (1) it is not what the language says; and (2) the copyright holder of GPLv2 code has no legal right to impose a license change on the holders of the copyrights in the other code included in the collective work. What the GPLv2 licensor can, and does, say is, if you are going to include my work in a collective work, then the collective work must also be licensed under GPLv2. What that means is all works included in the collective work must either be under

GPLv2 or under a license that is compatible with GPLv2.⁴ This concept of compatible licensing is manifested in the Section 2 phrase “whose permissions for other licensees extend to the entire whole.” It doesn't say that GPLv2 is applied to each component, only that the permissions granted under GPLv2 (those being the permissions to copy, modify and redistribute in source code form) apply to each component.

Finally, GPLv2 does not include an express patent license grant. Rather, in Section 6 the GPLv2 makes clear that no other restrictions can be imposed on recipients, which would include any restriction arising from a patent held by the distributing party. In section 7 the GPLv2 makes clear that, if conditions are imposed on the distributing party that would interfere with the rights granted under the license, the distributing party is not to redistribute the software. These provisions have been construed as granting an implied license from a GPLv2 distributing party under any patent claims of that distributing party that read on GPLv2 code they distribute and preventing such a distributing party from entering into any form of license agreement with respect to patent rights that would not extend to all downstream recipients. GPLv2 does not prevent distributing parties from entering into other forms of agreements related to patents as evidenced by the Microsoft-Novell arrangement announced in the fall of 2006.

GNU General Public License, version 3

(“GPLv3” - <http://www.fsf.org/licensing/licenses/gpl.html>)

While in most instances GPLv3 operates in the same manner as GPLv2, there are some important distinctions. As with GPLv2, GPLv3 addresses both derivative works and collective works with the following language from Section 0:

To “modify” a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a “modified version” of the earlier work or a work “based on” the earlier work.

A “covered work” means either the unmodified Program or a work based on the Program.

The language from Section 5(c) makes clear that all such derivative and collective works must be licensed under GPLv3:

You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.

A key difference regarding collective works between this language and that of GPLv2 is that GPLv3 now states that it will apply to all parts of the collective work. Gone is the reference in GPLv2 to “permissions” which allowed GPLv2-compatible licenses to govern other parts. This is an important distinction that should not be lost.

GPLv3 deals with compilations in much the same manner as GPLv2 with the following provision in Section 5:

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined

⁴ See <http://www.fsf.org/licensing/licenses/> for a list of GPL-compatible licenses

with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

This language now provides a clear statement that distinguishes a compilation from a collective work and makes clear that works licensed under GPLv3 can be included in such compilations so long as nothing in the license for the compilation interferes with the operation of the GPLv3 license with respect to the code licensed under GPLv3.

Unlike GPLv2, GPLv3 contains an express patent license in Section 11. That license only applies if the licensor makes a modification to GPLv3 licensed code and then redistributes the code, at which point the license applies not only to the licensor's modification but to the entire modified work, whether merely a derivative work or a collective work. There continue to be restrictions in GPLv3 on patents licensed from third-parties that apply to the GPLv3-licensed work, and any party looking to redistribute a modified GPLv3-licensed work would be well advised to be familiar with these provisions.

GNU Lesser General Public License, version 2.1

(“LGPLv2” - <http://www.opensource.org/licenses/lgpl-license.php>)

The LGPLv2 mirrors the GPLv2 in its application to derivative works, collective works, and compilations. The difference arises in Sections 5 and 6 of LGPLv2 where it permits the use of LGPLv2-licensed code (typically, libraries) with code licensed under a different license, including a proprietary license. The one limitation imposed on such combinations is that the licensor of the entire work (including both the non-LGPLv2 code and LGPLv2 code) must provide the licensee with the source code for the LGPLv2-licensed code and permit the licensee to make modifications in that code AND not prohibit the licensee from reverse engineering the non-LGPLv2 code included in the entire work solely for purpose of debugging the modifications to the LGPLv2 licensed code. This provision does not grant the licensee the right to copy, otherwise modify, or redistribute the non-LGPLv2 code included in the entire work.

LGPLv2 contains provisions in Sections 10 and 11 pertinent to patents that correspond with Sections 6 and 7 of GPLv2.

Apache License, version 2

(<http://www.opensource.org/licenses/artistic-license-2.0.php>)

The Apache License does not follow the same formula for derivative and collective works, but it still operates in much the same manner as the GPL. The Apache License uses the following definition:

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

Note two things about this definition. A modification that does not rise to the level of an original work of authorship, i.e., it is not protectible by copyright, does not create a derivative work. Thus

the author of such a de minimis change would have no independent copyright in the work. Second, no collective work is created if a separate work merely links or binds by name to the interfaces of the Apache-licensed work or derivative thereof. Using the terms linking and binding in this context causes some confusion because in GPL semantics these terms are used to imply two pieces of code that have been combined in such a manner as to create a single work unless the licensor provides an exception.⁵ Hence, the standard approach for the Apache License in this context is somewhat equivalent to the GPL+exception. By contrast with the GPL, the Apache License makes no specific reference to compilations likely on the premise that no such reference is necessary.

The Apache License contains an express patent license grant in Section 3. Unlike GPLv3, the patent license only extends to those patents held by a contributor of code and that apply to that contributor's contribution, not to the entire work.

Artistic License, version 2

(<http://www.opensource.org/licenses/artistic-license-2.0.php>)

The Artistic License deals separately with derivative works, collective works, and compilations. In addition, the Artistic License differentiates derivative works into (a) those works for which the modifications were explicitly requested by the copyright holder for the licensed work, referred to as the Standard Version, and (b) those works containing modifications not explicitly requested by the copyright holder for the licensed work, referred to as Modified Version. This approach parses copyright law in a rather unusual way in that it permits modifications, but the manner in which you document and redistribute the derivative work varies depending on whether it is a Standard Version or a Modified Version, with the redistribution obligations being far more burdensome for Modified Versions.

The Artistic License addresses compilations in Section 7:

You may aggregate the Package (either the Standard Version or Modified Version) with other packages and Distribute the resulting aggregation provided that you do not charge a licensing fee for the Package. Distributor Fees are permitted, and licensing fees for other components in the aggregation are permitted. The terms of this license apply to the use and Distribution of the Standard or Modified Versions as included in the aggregation.

This is pretty straightforward and should not be a cause for confusion or concern.

The Artistic License addresses collective works in Section 8:

You are permitted to link Modified and Standard Versions with other works, to embed the Package in a larger work of your own, or to build stand-alone binary or bytecode versions of applications that include the Package, and Distribute the result without restriction, provided the result does not expose a direct interface to the Package.

This is one of the more challenging turns of a phrase you will run into in an open source license. For example, in saying you can redistribute the result “without restriction,” does it mean you can't place any limitations on what the licensee does with the collective work or does it mean that none of the restrictions in this license apply. The former would be quite onerous, and the latter could be stated more clearly. The same for the last phrase of this paragraph when it says “provided the

⁵ See <http://www.fsf.org/licensing/licenses/gpl-faq.html#LinkingWithGPL>

result does not expose a direct interface to the Package.” What does that phrase mean? Does it mean that you can embed the Package and use the Package within the context of your larger work so long as some using your larger work is unaware that the Package is included and can't use it independently. That would appear to be the most logical interpretation, but it could certainly have been stated more clearly.

Finally, on the subject of patents, the Artistic License included a patent grant, but only from the Copyright Holder. Specifically, Section 13 states:

This license includes the non-exclusive, worldwide, free-of-charge patent license to make, have made, use, offer to sell, sell, import and otherwise transfer the Package with respect to any patent claims licensable by the Copyright Holder that are necessarily infringed by the Package.

To fully understand this section, you need to refer back to the license's definition of Copyright Holder:

"Copyright Holder" means the individual(s) or organization(s) named in the copyright notice for the entire Package.

Because the patent grant is not limited to the contributions of a particular copyright holder, a subsequent modified version could, in fact, implicate the patents of another, earlier copyright holder in a manner that copyright holder did not intend. Let's take a hypothetical. “A” is the copyright holder of a Standard Version, and that Standard Version reads on claims 1 and 2 of a patent also held by A. However, the Standard Version does not read on claim 3 of that same patent. “B” comes along and modifies the Standard Version, creating a Modified Version, in a manner that now reads on claim 3 of A. Under this language, A's Claim 3 would be licensed with B's Modified Version because A would necessarily be named in the copyright notice (remember, the party creating a derivative work only holds the copyright in their modifications, not in the original work that was modified). A further complicating factor in this language is the statement that the licensed patent claims would include “any claims licensable by the Copyright Holder,” not solely the claims owned by that Copyright Holder. What if those “licensable” patents are royalty-bearing or have use restrictions? This requirement would have made more sense had it been limited to claims “licensable by the Copyright Holder without restrictions or royalties.”

It is worth noting that the Artistic License, version 2 does not appear to be widely used at this time, and perhaps for good reason. I have included it here to indicate another variation in approach and to demonstrate the need for careful drafting.

Common Public License, version 1

(“CPL” - <http://www.opensource.org/licenses/cpl1.0.php>)

Eclipse Public License, version 1

(“EPL” - <http://www.opensource.org/licenses/eclipse-1.0.php>)

The CPL and EPL are treated as the same license since the text of the licenses are identical in all but name. The CPL takes yet another approach to derivative works, somewhat defining them in circular fashion. The CPL defines a Contribution to include a change or addition to an existing program licensed under the CPL, but excludes from this definition contributions “which: (i) are separate modules of software distributed in conjunction with the Program under their own license agreement, and (ii) are not derivative works of the Program.” So, in permitting the creation of derivative works, the CPL makes clear that compilations are permitted, but it does not really

address the concept of collective works at all. In theory, then, a person could modify a CPL-licensed work specifically for the purpose of including it in a larger work, and so long as they made the source code for their modified, CPL-licensed work available, they could license the larger work under any terms they choose, including proprietary license terms. Of course, a key question in this context is what constitutes a “separate module.” This approach is further supported by the language of Section 3 which states:

A Contributor may choose to distribute the Program in object code form under its own license agreement, provided that:

- a) it complies with the terms and conditions of this Agreement; and
- b) its license agreement:
 - i) effectively disclaims on behalf of all Contributors all warranties and conditions, express and implied, including warranties or conditions of title and non-infringement, and implied warranties or conditions of merchantability and fitness for a particular purpose;
 - ii) effectively excludes on behalf of all Contributors all liability for damages, including direct, indirect, special, incidental and consequential damages, such as lost profits;
 - iii) states that any provisions which differ from this Agreement are offered by that Contributor alone and not by any other party; and
 - iv) states that source code for the Program is available from such Contributor, and informs licensees how to obtain it in a reasonable manner on or through a medium customarily used for software exchange.

This approach gives a licensee wishing to redistribute the ability to incorporate the CPL-licensed code into larger works, including binary-only works, and to license those larger works under different license terms so long as the license terms of the CPL continue apply to the incorporated CPL-licensed work.

The CPL contains an express patent license grant in Section 2.b that is purely contribution based. This provides certainty to a contributor as to the patent license obligations the contributor is assuming. In this regard the provision is quite similar to that of the Apache License, version 2.

Mozilla Public License, version 1.1

(“MPL” - <http://www.opensource.org/licenses/mozilla1.1.php>)

The final license considered is the MPL. It is important because it is the license used for the popular FireFox browser and Thunderbird e-mail client. One thing unique about the MPL is the breadth of its scope when defining modifications. Like the CPL and EPL, the MPL defines modifications as being any change to the code whatsoever. However, both the CPL and EPL then narrow their scope by stating that the definition of “Contributions” does not include that which would not constitute a derivative work. The MPL contains no such limitation and, thus, claims to apply to changes that, in and of themselves, may not constitute copyrightable material. This would appear to introduce a rather unique, and perhaps undesirable, aspect to the MPL in that any attempt to assert the license with respect to a non-copyrightable change could only be enforced under contract law, not copyright law.

That anomaly aside, the MPL is also unique in its approach to both compilations and collective works, treating them as one and the same. The MPL does so through the grant language contained in sections 2.1(a) (covering initial contributions):

under intellectual property rights (other than patent or trademark) Licensable by Initial Developer to use, reproduce, modify, display, perform, sublicense and distribute the Original Code (or portions thereof) with or without Modifications, and/or as part of a Larger Work

and 2.2(a) (covering subsequent contributions):

under intellectual property rights (other than patent or trademark) Licensable by Contributor, to use, reproduce, modify, display, perform, sublicense and distribute the Modifications created by such Contributor (or portions thereof) either on an unmodified basis, with other Modifications, as Covered Code and/or as part of a Larger Work

and reaffirmed with the language in section 3.7:

You may create a Larger Work by combining Covered Code with other code not governed by the terms of this License and distribute the Larger Work as a single product. In such a case, You must make sure the requirements of this License are fulfilled for the Covered Code.

As a consequence, the MPL provides broad rights to combine MPL covered code with any other code, including a larger work (either a compilation or collective work) so long as you make the source code of the MPL licensed code available.

The MPL follows the contribution approach to its express patent license grant language, i.e., a contributor is only providing a patent license to that contributor's contribution, including any necessary patent license pertaining to the combination of that contribution with the existing work. There is one last catch to the MPL, and it arises in the context of the express patent license. According to the language of sections 2.1(b) and 2.2(b), the patent license only pertains to the code in its source code form. The pertinent language appears in the following:

1.10. "Original Code" means Source Code of computer software code which is described in the Source Code notice required by Exhibit A as Original Code, and which, at the time of its release under this License is not already Covered Code governed by this License.

1.9. "Modifications" means any addition to or deletion from the substance or structure of either the Original Code or any previous Modifications. When Covered Code is released as a series of files, a Modification is:

A. Any addition to or deletion from the contents of a file containing Original Code or previous Modifications.

B. Any new file that contains any part of the Original Code or previous Modifications.

2.1 (b) under Patents Claims infringed by the making, using or selling of Original Code, to make, have made, use, practice, sell, and offer for sale, and/or otherwise dispose of the Original Code (or portions thereof).

2.2 (b) under Patent Claims infringed by the making, using, or selling of Modifications made by that Contributor either alone and/or in combination with its Contributor Version (or portions of such combination), to make, use, sell, offer for sale, have made, and/or otherwise dispose of: 1) Modifications made by that Contributor (or portions thereof); and 2) the combination of Modifications made by that Contributor with its Contributor Version (or portions of such combination).

Section 3.8 of the MPL permits executable versions of the work to be distributed under other license terms, so long as the source code associated with the MPL-licensed portion of those executables is made available under the MPL. This is a potential trap for the unwary in that an executable version of the MPL-licensed source does not necessarily carry with it the patent claims licensed with the source code, and the licensor of the executable is not obligated to provide a patent license covering the executable.

Applying License Terms to Software Distributions

Having examined the way a variety of open source licenses treat the issues of derivative works, collective works, compilations and patents, we now turn to the mechanisms that may be utilized to effect these distributions. In this context we will focus on two principal means of packaging software: media packaging and electronic packaging. Given that both forms of packaging technically exist only in electronic form, a couple of definitions are helpful. Media packaging is defined as providing content on the same physical media or by means of the same media channel. Electronic packaging is defined as the organization of content into distinct electronic packages.

As a first premise, I assert that the proper measure of the relationship of one or more software applications is not the packaging but the actual interrelationship of the packages. In other words, regardless of mode of delivery, the issue is whether the works are derivative, collective, or merely aggregated (compilations) when they are resident on the computer that will run them.

Consistent with that first premise, let's examine various forms of physical packaging. At least three forms of such media packaging come to mind: fixed media, movable media, and electronic transmission, such as FTP. No one would logically argue that the mere fact two software applications reside on the same fixed media, such as a computer hard drive, creates any form of special relationship between those applications under copyright law. If that premise is true, which I believe it to be, then the fact the media is movable, such as a flash drive, or the content is delivered via electronic transmission, such as via FTP, would fall into that same category. So the mere presence of two software applications in the same media packaging should, in and of itself, never be a concern. This second premise is true under all of the licenses discussed above.

The second form of packaging, electronic packaging, may, at first blush appear to differ. Electronic packaging of software can occur in any of the following examples: an electronic package management system, such as RPM; a JAVA class library, or a software appliance. Because of the distinct differences of each of these electronic packages, it is worth examining them individually.

A package management system does not itself define the interrelationship of the software applications that may be included in any one package, other than to contain information about the included software. An example of a popular open source package manager is RPM. RPM consists of a software package file format and a free software tool which installs, updates, uninstalls, verifies and queries software packaged in this format. Each RPM package contains a package label holding the following information: the name of the software; the version of the software; the

package release; and the architecture on which the package is intended to run, e.g., X86. As a consequence, an RPM package does not, by itself, indicate a relationship among the code contained therein and any other code. In this manner, an RPM carries many of the same characteristics as a media package and not unique characteristics that define the interrelationship of the packaged software.

This same approach is true for a JAVA class library as packaged in a JAR file. Such files are mere aggregation tools; they again do not define the interrelationship of the individual programs or functions that may be included therein. In fact, there is no particular reason that everything in a single JAR file be licensed under the same license. It is entirely possible for a JAR to contain code licensed under the GPL, the LGPL, or any of a number of different open source licenses. The mere fact that these program or functions have been aggregated into a single JAR to make them readily available in a JAVA application context does not impute a relationship among the varying contents of a JAR. Such a relationship may or may not exist, and it is only when such an interrelationship exists that one need to go to the question of the nature of that relationship, i.e., derivative work or collective work. From this viewpoint, JAR files are mere compilations.

A software appliance is a software application packaged with just enough operating system components (abbreviated as JeOS) to allow it to run on hardware or a virtual machine. A JeOS is a customized operating system designed to fit the needs of a particular software application. However, the mere fact that the operating system has been customized to minimize its size and to work with a specific software application does not change the characteristic of the operating system as an independent work. This is especially true with respect to Linux, and it is worthwhile to digress for a second in looking at the standard Linux construct.

Linux, or perhaps more appropriately, GNU/Linux, consists of a kernel and various utilities and applications built to run on that kernel. The Linux kernel itself is of a modular construct, with groups of files organized into these modules. The internal Linux kernel modules pass information back and forth among themselves using internal symbols. In GNU/Linux the kernel is licensed under GPLv2. The utilities and applications, which exist in what is commonly referred to as “user space,” are licensed under a variety of open source licenses, including most of the licenses mentioned above.

The Linux kernel provides a number of defined application interfaces that the user applications are permitted to use, and so long as a user space application does not seek to export an internal symbol, i.e., one of the symbols passed among the internal Linux modules, it is construed to be an independent work and neither a derivative work of the Linux kernel or a collective work as combined with the Linux kernel. Hence, a user space application may be under almost any license, including a proprietary license, as long as it behaves in the specified manner. Both non-commercial and commercial Linux distributions combine a Linux kernel with a wide variety of user space applications to provide a robust, general purpose operating system that is reasonably easy to install, manage and update. However, the mere fact that the kernel and user space applications are packaged in the same media does not change their characteristics as independent works.

Applying this construct of a typical Linux operating system to a software appliance that utilizes a Linux kernel-based JeOS, it is consistent to view the software application married with the JeOS and the JeOS as independent works in the same manner as the Linux kernel and user space applications. The mere fact that they have been placed on the same media or in the same electronic package does not change that relationship. Of course, this assumes that the application has been developed independently from the JeOS, and the more development distance that can be

placed between the two, the stronger the argument for independent works.

If a software application developer decides to build its own JeOS utilizing Linux and elects to export internal kernel symbols for use by the application, the software application and the JeOS may be construed as either a derivative work or a collective work, and the GPLv2, the license applicable to the Linux kernel, would be the governing license.

Lessons Learned and Lessons Applied

From the foregoing analysis we see that mere packaging of code rarely impacts the licenses applicable to the various components of the code. Rather, one needs to look at the interrelationship between software applications, or software applications and the operating system, to determine whether the two are independent works merely compiled into the same package or whether they could be construed as either derivative works or a collective work. Consequently, software application developers looking to utilize one of these packaging techniques can adopt practices that decrease the likelihood of a finding of a derivative or collective work and increase the likelihood of the combination being construed as a mere compilation or aggregation, regardless of packaging. Specifically, software application developers should consider the following practices:

- Maintain distinct development trees for the software application and any operating system on which it is to run, including a JeOS.
- If possible, utilize an operating system, including JeOS, that has been developed by an independent third party. This increases the likelihood that the two works will be properly construed as independent works.
- Do not export or seek to export Linux kernel internal symbols for use by your application. If such an export is necessary, make sure the driver or interface effecting that export is available under the GPLv2.
- Ensure that your software application in the form distributed is equally capable of running on other forms of operating systems. For example, if the application is being distributed as a part of a software appliance utilizing a JeOS, it is helpful if that same application code is capable of running on any standard version of the Linux operating system. This strengthens the argument of independence.

Conclusion

Open source licenses differ in their treatment of derivative and collective works, although almost all take the same benign approach to compilations or mere aggregations. It is not the manner in which an application is packaged with other applications or an operating system kernel that determines whether one of the two works is a derivative of the other or whether the combination is a collective work, it is the actual manner in which the two interoperate and are dependent on each other. Following a few simple practices will increase the likelihood that your software application will always be considered an independent work regardless of the packaging mechanism that places it in proximity to an open source operating system like Linux.

About the author

Mark Webbink is a Visiting Professor of Law and Executive Director of the Center for Patent Innovations at New York Law School. Webbink is also a Senior Lecturing Fellow at Duke Law School and has served as an Adjunct Professor at NCCU Law School. From 2000 to 2007 Webbink served in various capacities with Red Hat, Inc., including General Counsel, Deputy General Counsel for Intellectual Property, Senior Vice President and Secretary. Webbink presently serves on the board of directors of the Software Freedom Law Center. Webbink has written and spoken extensively on the subjects of open source software, software patents, and patent reform. Webbink received his B.A. Degree from Purdue University in 1972, his Masters in Public Administration from the University of North Carolina – Chapel Hill in 1974, and his J.D., magna cum laude, from North Carolina Central University School of Law in 1994. Webbink maintains a website on open source and intellectual property law at www.walkingwiththeelephants.com.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Webbink, Mark (2010) 'Packaging Open Source', *IFOSS L. Rev.*, 1(2), pp 83 – 98
DOI: <http://dx.doi.org/10.5033/ifosslr.v1i2.26>

Copyright © 2010 Mark Webbink.

This article is licensed under a Creative Commons unported 3.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Passport Without A Visa: Open Source Software Licensing and Trademarks

Tiki Dare^a, Harvey Anderson^b

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

With an introduction by Amanda Brock^c

(c) General Counsel, Canonical Group.

DOI: [10.5033/iffosslr.v1i2.11](https://doi.org/10.5033/iffosslr.v1i2.11)

Abstract

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

Keywords

Law; Free and Open Source Software; Trademarks

Info

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

Introduction by Amanda Brock

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

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

a country's domestic trade mark registry. This means that during the process of registration, the mark will be registered even though there are conflicting national marks, unless the owner of the conflicting mark notices that an application has been made¹. There is also the potential for owners of conflicting marks to challenge the mark in a territory for a period after registration, which is why applicants are generally advised to undertake a search themselves prior to applying for the mark.

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

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

Usage of trade marks must, as the article explains, be consistent - and this requirement exists in the UK and Europe as well as in the US. The rather amusing Penguin biscuit cases² were a great example of this. At the time, the Penguin chocolate biscuit was well-known in the UK. Each biscuit was packaged in a wrapper decorated with a lighthearted image of a penguin. When a supermarket brought out a copycat "Puffin" biscuit, the Penguin manufacturers felt that they had a case for trade mark infringement and passing off. However, Penguin's claim suffered when it was discovered that the marks which Penguin had registered, which included a number of illustrations for use on the biscuits, had not been used for some time and that the images which had been used were not in line with the registrations. The Penguin ultimately won its claim for unregistered trade mark infringement against the Puffin (but lost in its registered trade mark claim).

Anderson and Dare's article suggests a split in branding between enterprise and community versions of open source projects. However, marks in brands with a strong community contribution may not necessarily split in this way. Ubuntu, the operating system distributed by Canonical, does not have differentiated enterprise/community versions for important philosophical reasons. This stands in contrast to the dual-branded distributions, like Red Hat³ and its community version Fedora⁴. This is easily resolved, in Canonical's case, by the brand allowing a non-commercial use of the protected mark by the community, and by making a clear distinction in the brand's trade mark policy between the freely licensed not-for-profit or non-commercial usage which is granted for free in a general licence to the community (subject to compliance of the non-commercial user with the trade mark policy) and commercial usage. Commercial usage may not only be subject to the rules of a trade mark policy, but also subject to the terms of a commercial licence available at the brand owner's discretion. In other words, there is no guarantee that the trade mark owner will grant a commercial licence to the trade mark.

For useful information on trade marks registrations in Europe or to check if a trade mark registration exists in Europe, see the OHIM web site⁵.

1 Though commercial 'watch' services which will monitor potentially-conflicting marks do exist

2 United Biscuits (UK) Limited v Asda Stores Limited (Chancery Division, 18th March, 1997) Robert Walker J

3 See Red Hat, <http://www.redhat.com/>

4 See Fedora Project, <http://fedoraproject.org/>

5 <http://oami.europa.eu/>



Introduction

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

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

Licensing plays a critical role in the open source community as it is the operative tool to convey rights and redistribution conditions. The F/OSS licenses have focused primarily on copyright and patent rights, which directly protect the underlying "bits" of code - the software itself. Addressing the third pillar of intellectual property rights – trademarks⁶ – at all in F/OSS licenses is a relatively recent trend, and none of the open source licenses grant trademark rights. It was logical to start with copyright and patents, because trademark law protects the name and logos or other branding elements that are applied to the underlying code, but not the code itself. As open source software has become widely adopted among consumers and now generates significant revenues for some companies, the need to understand trademark law and to develop licensing and other conventions for managing trademarks is increasingly evident in the community.

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

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

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

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

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

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

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

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

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

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

Trademark 101

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

⁷ The GNU Project, “The GNU General Public License” version 3, at s7 (e). Available at: <http://www.gnu.org/licenses/gpl.html>

⁸ See http://en.wikipedia.org/wiki/Lanham_Act#History

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

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

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

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

Trademark Licensing and Quality Control

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

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

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

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

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

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

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

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

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

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

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

Implied Licenses

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

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

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

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

¹¹ The courts have also had few opportunities to review open source software licenses. Recently in the U.S., the August 2008 *Jacobsen v. Katzer* decision by the Court of Appeals for the Federal Circuit upheld enforcement of the copyright restrictions of an open source license.

¹² Trademark Licensing, Neil J. Wilkof & Daniel Burkitt, Section 11-23 p. 252.

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

Passport without a Visa - Challenges in the Community Context

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

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

First is a lack of clarity about when a trademark license is needed, and what use of the trademark is permissible without one. A license is needed for use of a trademark in a company name, product name or service name, or whenever a name or logo¹⁴ is “affixed” to a product or service. In practice, this means that in the absence of a trademark license, a source code licensee should not use the name of the source code base as the name of her or his own software distributions, or use any logos associated with the source code F/OSS project teams may inadvertently create some confusion in their community by including the name and especially splash-screen or logo files in the code base they make available under a source license.

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

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

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

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

branded it “Iceweasel”.¹⁵ This is often called forking¹⁶ which results in two similar but distinct code bases. The lack of compatibility between the two code bases can create real problems.

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

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

Newcomer status for trademarks is a third major hurdle in protecting trademarks in the F/OSS arena. The Software Freedom Law Center¹⁷ (SFLC) and other non-profit groups counsel in this area, but independent resources like these are few and not widely known.

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

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

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

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

Best Practices

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

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

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

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

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

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

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

Conclusion

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

About the authors

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

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

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

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Dare, Tiki & Anderson, Harvey (2009) 'Passport Without A Visa: Open Source Software Licensing and Trademarks', *IFOSS L. Rev.*, 1(2), pp 99 – 110
DOI: [10.5033/ifosslr.v1i2.11](https://doi.org/10.5033/ifosslr.v1i2.11)

Copyright © 2009 Tiki Dare & Harvey Anderson.

This article was first published in the International In-house Counsel Journal.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Case law report - BACK TO THE FUTURE: Hinton v Donaldson, Wood and Meurose (Court of Session, Scotland, 28th July, 1773)

Iain G. Mitchell QC^a

*(a) Chairman, Scottish Society for Computers and Law;
Chairman, Scottish Lawyers' European Group; Chairman,
Faculty of Advocates IT Group; Lecturer, Honorary Board of
Lecturers, Institut für Informations-, Telekommunikations- und
Medienrecht, Westfälische Wilhelms- Universität, Münster;
Freeman, Worshipful Company of Information Technologists.*

DOI: [10.5033/ifosslr.v1i2.23](https://doi.org/10.5033/ifosslr.v1i2.23)

Abstract

History moves in cycles, and recent debate around the extent and term of copyright is no exception to this rule. There are two forces at play, one being the view that knowledge is of value and indeed belongs to all mankind, and the other being that knowledge, as a creation of individuals, is property to be regulated and shared according to the wishes of its owner. This case law report examines the debate through a different lens, moving back from software and the Internet in favour of Stackhouse's History of the Holy Bible, Scottish printers and the year 1773.

Keywords

Law; information technology; Free and Open Source Software; Scotland; copyright

Info

This item is part of the [Case law reports](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

1. Introduction

Human nature does not change, but what new technology does achieve is to create a new milieu in which the battles of the past can come back to haunt us in new guise. Because of the new technology, the issues can seem very new, but often what lies behind them is as old as the hills.

Take, for example, the battle which has raged for years over copyright term extensions, DRM as a means to try to control access to content which may, itself, have passed into the public domain, and the turf wars between copyright and copyleft. What lies at the back of all of this is, arguably, a conflict of values.

On the one hand, there is the value that all knowledge, all ideas are the patrimony of mankind, and they can and should be free.

On the other hand there is the equally deeply held value of property: "what is mine is mine".

What sets up that conflict is the very idea of Intellectual Property itself: the trick of copyright was, in essence, to commoditise, if not ideas, then, at least, the expression of ideas. Section 1 of the Copyright, Designs and Patents Act 1988 provides:

"Copyright is a property right which subsists in accordance with this Part in the following descriptions of work—

- (a) original literary, dramatic, musical or artistic works,
- (b) sound recordings, films, broadcasts or cable programmes, and
- (c) the typographical arrangement of published editions."¹

That is the key to the whole concept of intellectual property: it is property². Property is valuable: it can be rented out, it can be bought and sold, (or, to use the familiar term in the law of incorporeal movables, it can be assigned). In short, it becomes a commodity.

And so, there is set up what can be a fundamental conflict of values: on the one hand, a common patrimony which should be free, and, on the other hand, private property which can be immensely valuable to its owner.

This battle is played out every day in a society where the IP owners try to persuade an increasingly deaf public that the downloading of file-shared music and other copying of content is theft and, though, as the warnings on every DVD used to tell us:

"You wouldn't steal a car. You wouldn't steal a handbag. You wouldn't steal a television. You wouldn't steal a movie. Downloading pirated films is stealing. Stealing is against the law. Piracy: It's a crime",³

the *zeitgeist* would seem to see nothing wrong in unrestricted copying and file sharing.

It was the genius of the invention of copyright and other forms of intellectual property that there is set up a Faustian bargain between the State and the author or creator (or, in any event, the first owner of the copyright, which, thanks to the essentially commercial nature of the concept, may well not actually be the author himself) that there is granted, in effect, a monopoly for a fixed term, during which the author can make his money, but at the end of that term, his work passes into the public domain and becomes part of the common patrimony. But, no sooner is that bargain set up than people want to renegotiate it: on the one hand those who wish to see the enlargement of the public domain, and, on the other hand, those who want to go on exploiting their property. In the middle, there may be a dispassionate debate about how best to balance these competing interests,

1 Copyright, Designs and Patents Act 1988, S1. Available at: http://www.opsi.gov.uk/acts/acts1988/UKpga_19880048_en_1.htm . Unofficial consolidated version available at: <http://www.ipo.gov.uk/cdpact1988.pdf>

2 The characterization of intellectual property as property, and not (as in the Napoleonic system) an emanation of the person is mandated by the Act. The philosophical basis of intellectual property in the Common Law jurisdictions is clearly as a property right. The Copyright, Designs and Patents Act is a direct linear successor of the Statute of Anne (as to which, see below) and it was that statute which, in effect, invented Intellectual Property. Thus, it is understandable why, in the Common Law jurisdictions, the nature of Intellectual Property as property seldom comes into question, and may, in part, explain why, notwithstanding the different philosophical approach of the Napoleonic systems, those systems are increasingly coming to reflect an economic model where Intellectual Property is sought to be exploited. It may also explain why International Agreements, such as the WIPO treaties, reflect a strong bias in favour of property rights. It is little wonder that, whatever continental legal systems may say, entrepreneurs the world over are attracted to the proprietary model.

3 See "Piracy: It's A Crime" advertisement, video available online at: <http://www.youtube.com/watch?v=iPcHhOBd-hI>

but that should not blind us to the fundamental tension which lies beneath, which is there and which will never go away.

In that tension, the rightsholders can often come off best. It is relatively easy to demonstrate the value of the property on the basis of the income it generates – more difficult by far to ascribe a value to the public domain⁴. So, in a debate conducted on purely economic terms, the rightsholders have a clear advantage. Thus, a proper debate requires also broader social and philosophical issues to be addressed. Yet, even in such a broader debate, the tension is still there: especially when the rightsholders see their property rights in absolute terms.

This has been played out before the United States Congress more than once. Jack Valenti, President of the Motion Picture Association of America, giving testimony before a congressional committee in 1982, argued against limitations on intellectual property:

"Creative property owners must be accorded the same rights and protection resident in all other property owners in the nation."⁵

There is, in the United States a problem with that: Article 1 §8 of the United States Constitution provides:

"he Congress shall have Power...To promote the Progress of Science and useful Arts, by securing *for limited Times*⁶ to Authors and Inventors the exclusive Right to their respective Writings and Discoveries."

Nothing daunted, in the hearings on the Sonny Bono Copyright Term Extension Act, Congresswoman Mary Bono stated:

"Actually, Sonny wanted the term of copyright protection to last forever. I am informed by staff that such a change would violate the Constitution. I invite all of you to work with me to strengthen our copyright laws in all of the ways available to us. As you know, there is also Jack Valenti's proposal for term to last forever less one day. Perhaps the Committee may look at that next Congress."⁷

And so the battle rages on, through WIPO Treaties, European Directives, the DMCA, all played out in the fora of politics, the courts, by campaigners, by industry lobbyists and by lawyers: the money battles the ideals.

We've never known anything like it before, or have we?

2. Back to the Future

Once upon a time, in England, the members of the Stationers' Company enjoyed a monopoly on the printing of books, granted to them by Queen Mary in 1557. They would buy a manuscript from an author, and once printed, would enjoy a perpetual monopoly in the printing of the work. The

4 Although some have tried – Rufus Pollock is probably the best known:
http://www.rufuspollock.org/economics/papers/value_of_public_domain.ippr.pdf

5 Home Recording of Copyright Works: Hearings on H. R. 4783, H. R. 4794, H. R. 4808, H. R. 5250, H. R. 5488 and H. R. 5705, before the Subcommittee on Courts, Civil Liberties, and the Administration of Justice of the Committee on the Judiciary of the House of Representatives, 97th Congress, 2nd Session (1982): 65 (testimony of Jack Valenti).

6 Emphasis supplied

7 Hearings on Sonny Bono Copyright Term Extension Act, House of Representatives, 7th October, 1998. Congressional Record, Vol. 144 page H9951.

authors received no further reward; nor could authors become members of the Stationers' Company, neatly excluding them from any chance of participation in the income stream.

This monopoly finally expired in England in 1695; there was never any similar monopoly in Scotland.

After the Union of Scotland and England, the Parliament of Great Britain passed the Statute of Anne 1709⁸ the Long Title of which is *An Act for the Encouragement of Learning, by vesting the Copies of Printed Books in the Authors or Purchasers of such Copies, during the Times therein mentioned*. The preamble narrates the evil which the Act sets out to address:

“Whereas Printers, Booksellers, and other Persons, have of late frequently taken the Liberty of Printing, Reprinting, and Publishing, or causing to be Printed, Reprinted, and Published Books, and other Writings, without the Consent of the Authors or Proprietors of such Books and Writings, to their very great Detriment, and too often to the Ruin of them and their Families: For Preventing therefore such Practices for the future, and for the Encouragement of Learned Men to Compose and Write useful Books; May it please Your Majesty, that it may be Enacted, and be it Enacted....”

The Act provided that, in respect of books existing at the date of commencement of the Act, there was granted to the authors of those books (and booksellers who had bought existing books from their authors): “the sole Right and Liberty of Printing such Book and Books for the Term of One and Twenty Years”, and, to authors of books yet to be written, a similar right and liberty for an initial period of fourteen years, with (if still alive at the end of that term) an additional fourteen year term.

The Act provided for penalties against those who infringed the right (including a fine, payment to the owner of the right, and forfeiture and destruction of offending copies).

There are some noteworthy features of the regime, not least the shift of power from booksellers to authors and the provision of a mechanism for the setting of a reasonable sale price in circumstances where it is recognised that a bookseller was asking an excessive and unreasonable price for a book. In this provision, we see an acknowledgement by the legislature of the potential evils of a monopoly and an attempt to ameliorate them.

Although the right was known from the beginning as “copyright” it is not copyright as we know it, being, rather, a monopoly on the printing of certain books, which, of course, has inherent therein the right to stop others printing those books.

Pause a moment, and consider the parallels between the eighteenth century experience and the present day: first, there was a concentration of economic power in the entertainment industry (granted, London publishers and not Hollywood moguls, but that is just a function of changing technology) rather than the creators; second, that power was both expressed through and reinforced by control of distribution and was driven by a focus on technology - it is easier to print a book than it is to copy it by hand, and it is impossible to control what the buyer does with a printed book once it is in his possession - the parallels with digital rights management are obvious; and, third, there is legislative intervention, but here the parallels with today break down, for the legislation is seen as being directed against the economically powerful media industry and in

8 *An Act for the Encouragement of Learning, by vesting the Copies of Printed Books in the Authors or purchasers of such Copies, during the Times therein mentioned* (8 Anne c 19) - a facsimile and transcription can be found at <http://www.copyrighthistory.com/anne.html>

favour of the perceived temporary benefit of creators and the long term benefit of society as a whole.

3. A Preliminary Skirmish

The booksellers were not happy at this threat to their comfortable monopoly. They had a number of very profitable books, which had, however, under the Statute of Anne, inconveniently passed out copyright. Competitors were taking it upon themselves to print competing editions. Perhaps because the parliamentary lobbying industry was not what it is today, or perhaps because none of these publications featured an anthropomorphic mouse, the booksellers' target was not to try to get the legislature to extend the term of copyright; rather, as the entertainment industry might wish it could do today, they went running off to Court with their sights set on something altogether more lucrative: getting the courts to declare that there was a common law right of copyright, which (as in Sonny Bono's fantasy) is perpetual.

The test case concerned James Thomson's poem "The Seasons", the perpetual common law copyright in which, it was claimed, belonged to one Andrew Millar, who, in the Court of King's Bench in England, sued Robert Taylor for breach of copyright⁹. This notion of perpetual copyright was accepted by the Court on a majority decision of three to one, the leading judgement being delivered by Lord Mansfield¹⁰ who, when he had been a barrister, had acted for the booksellers.

The dead hand of the bookseller's monopoly descended upon England, and, indeed, shortly thereafter, the Booksellers obtained an injunction in *Donaldson v Beckett* to prevent the reprinting of the same poem; but, of course, what an English Court says to be the law of England, is not binding upon a Scottish Court in determining the law of Scotland. The reprint industry continued to flourish in Scotland, about which jurisdiction the London Booksellers felt much as Hollywood does today about China, or (amusingly) Canada, even using the term "pirates" to describe those who printed books in breach of the claimed copyright.

So, enter one James Hinton, the Jack Valenti of his day, who joined battle in Scotland over one of his particularly profitable, and statutory copyright expired publications: *Stackhouse's History of the Holy Bible*¹¹, which was being reprinted by Alexander Donaldson and John Wood, Booksellers in Edinburgh and James Meurose, bookseller in Kilmarnock. Armed with the sword of the decision in *Millar v Taylor*, Hinton sued Donaldson, Wood and Meurose in the Court of Session for declarator, interdict and damages in respect of breach of copyright.

However, Donaldson had a formidable friend and ally in the Scottish Advocate, James Boswell, whose reputation today rests almost entirely upon his biography of that great man of letters and lexicographer, Dr Samuel Johnson. But in his day, Boswell was associated with a number of high-profile cases in the cause of liberty¹². Boswell was amongst those who appeared for the defence.

9 Millar v. Taylor, 4 Burr. 2303, 98 Eng. Rep. 201 (K.B. 1769)

10 In view of the main subject of the present article, there is a delicious irony in the fact that Lord Mansfield, though a distinguished English lawyer, was Scottish and shares with the present author the distinction of having attended school in Perth.

11 Or, to give it its snappy full title: *A New History of the Holy Bible from the Beginning of the World to the Establishment of Christianity, with Answers to most of the controverted Questions, Dissertations on the most remarkable Passages, and a Connection of Profane History all along; to which are added, Notes explaining the difficult Texts, rectifying Mistranslations, and reconciling seeming Contradictions: The whole illustrated with proper Maps and Sculptures.*

12 For example, in 1777 he assisted the runaway slave, Joseph Knight, in his successful legal case in the Court of Session against his master, John Wedderburn of Ballendean, in which Lord Kames, delivering the leading judgement ringingly declared that 'we sit here to enforce right not to enforce wrong'. (Knight v Wedderburn (1778) Mor 14545)

The decision in *Hinton v Donaldson* is only telegraphically reported in the law reports¹³, but Boswell himself published a full report, containing the complete written judgements¹⁴, the full text of which is available online at the IFOSS L. Rev. web site¹⁵.

4. Hinton v Donaldson

By a majority of eleven to one, the judges decided that there was no common law right of copyright in Scotland. What is especially illuminating for a modern reader is the reasoning in their respective judgements.

Certain common threads run through the judgements: the acknowledgement that, although the judgement of the English Court deserves respect, it cannot tie the hands of the Court of Session; an examination of just what the Statute of Anne achieved; and a consideration of whether a perpetual proprietorship in literary works was conform to the law of nature, or, as a modern lawyer might put it, consistent with principle. In considering that last question, although the judges did look at economic issues, this was in the context of the sort of wider debate looking at underlying social and philosophical issues which is so needed today but which so often does not happen¹⁶. A lot of what they said remains both relevant and pertinent today.

Lord Kennet's view was that there was no basis for copyright in the law of nature: rather, it was "only such a kind of right as particular states have, in some instances, conferred by a patent or *privilegium* for a limited time".

Lord Auchinleck also saw the limited monopoly as an exercise of the prerogative power of the crown and saw no basis in common law for a perpetual right of copyright. The sense of the primacy of the public domain comes over strongly in his Opinion:

"If a man throws out a thing in company, whether instructive or entertaining, can he maintain that he has a right of property in this bon mot to him and his heirs for ever?

"And here I beg leave to say, unless it can be shewn there is a right of property in what a person utters verbally, there can be none in what he publishes to all mankind by printing it. Indeed, when a man publishes his thoughts, he gives them away still more than the man who utters them in conversation. The latter gives them only to his hearers; but the former to the whole habitable earth."

Lord Hailes found no trace in the common law of the claimed right. He was particularly scathing of the London Booksellers, *the Sages of St. Paul's Church-yard*:

"The doctrine of these *sages* is commodious: they *limit* or *enlarge* this common-law right as best suits their own conveniency."

And he points up their hypocrisy, noting that what they were in effect claiming was:

13 *Hinton v Donaldson* (1773) Mor 8307

14 *The Decision of the Court of Session upon the Question of Literary Property; in the cause John Hinton of London, Bookseller, Pursuer against Alexander Donaldson and John Wood, Booksellers in Edinburgh, and James Meurose, Bookseller in Kilmarnock, Defenders* (Pub: Boswell, Edinburgh, 1774)

15 <http://www.ifoSSLr.org/public/hinton-donaldson-resources.html>

16 As discussed above, the arguments today tend to be almost entirely economic, giving an inbuilt advantage in the debate to the rightsholders, especially where the income stream is seen to be attractive to government as source of taxation revenue.

"...it would be hard that any one should steal from me, what I have stolen from others."

The sole dissenting judge, Lord Monboddo, though dismissive of the notion of property in ideas, was, however, persuaded that:

"The common law of Scotland and of England, must, I think, be the same in this case, as the common law of both is founded upon common sense and the principles of natural justice, which require that a man should enjoy the fruits of his labours".

He sees this as the rationale of the Statute of Anne, but seems to have made the intuitive leap that it should also be the common law, and should apply in perpetuity. In effect, he sees property rights as having primacy, though, since the history of copyright is about the balancing and reconciling of the competing principles of public domain and private property, just where to strike that balance is what we should, today, describe as a "judgement call" and that was a call which he was entitled to make; but it is significant that, of these judges, speaking from the heart of the Scottish Enlightenment, the only one who was prepared to vote for primacy of private property was Lord Monboddo. It may be, however, that in reaching his view, he was unduly influenced by the romantic image of the poor starving author, rather than the economic reality of the wealthy publisher:

"I think it would be very hard and much to the discouragement of literature if an author, after spending a laborious life in composing a book, did not provide by it, not only for himself, but also for his family: nor is the remedy in the statute against this evil sufficient; for, the best books may be twenty years published, without having their merit known, and afterwards have a great and universal sale."

The reasoning of the Lord Justice Clerk was substantially similar to the reasoning of his brethren: the statute of Anne grants a privilege, but there is no basis in the common law for the claimed perpetual right:

"For it is certain, that neither Homer, nor Virgil, nor Chaucer, nor Spencer, had any idea, that, after they had published their works to the world, they, and their heirs and assigns, retained this property, this exclusive right of transcribing, or re-printing their works for ever.

"In short, upon examination it will be found, that there is no foundation for this *copyright* in authors, in the common principles of law, and that the only ground for it is this, that, from the love of knowledge, and the admiration of the works of learning and genius, mankind are prone to give authors, not only the merit, but the reward that is due to them for their works; and upon this principle every civilized state in modern times has introduced exclusive privileges to authors, in the publication of their own works, some for a longer, some for a shorter time. But this suggests no idea of an original property in the author; on the contrary, it is inconsistent with it..."

At the heart of this lies the moral argument that copyright is a limited privilege given by a civilised state to authors; it is not a right to be taken. At the jurisprudential level, that analysis does not altogether hold up today, at any rate in the Common Law systems where the law has moved towards seeing intellectual property as, indeed, property (albeit existing for a limited period only); but at the moral level, the analysis is as true as it ever was: this right of property does not exist inherently, but only because society has chosen to give it, to carve it out of the public domain.

Perhaps if the legislators to whom the lobbying efforts of the entertainment industry were directed were to bear that thought in mind, they would see more clearly the true balancing exercise which they are called upon to perform.

Lord Gardenston was likewise of the view that:

"...authors have in reason and equity a right to be protected in the sole and exclusive publication of their own works for a limited time. But the nature of the thing, and the practice of nations admits not of a real and perpetual property."

A substantial part of his lengthy Opinion is devoted to pouring scorn on the very idea of property in literary works. For example, he advances the picturesque argument that theft of all other property is gainful, but that theft of literary property, if it exists, is a "perilous theft by the nature of it", as "many a publication is attended with loss":

"...it would be but petty larceny; at worst, in a very few, the most aggravated and capital crime. - Who steals from common authors, steals trash; but he who steals from a *Spencer*, a *Shakespeare*, or a *Milton*, steals the fire of heaven, and the most precious gifts of nature. - So we must have new statutes to regulate those literary felonies."

The irony is ratcheted up further when, having stated that, on that argument, quotation is literary theft, he refers to "the author of a book called *The Elements of Criticism*" (who was, in fact his fellow judge, Lord Kames), whom he had always believed to be:

"an ingenious man, and a very honest gentleman; but in this view of the matter, he lies under a very criminal charge; every page of his book is enriched with quotations from the most classical poets and other authors."

Modern judges are less given to sarcasm, but the point that is so colourfully made is a serious one, which remains valid today: if there is to be perpetual property in literary works, civilised society, with its apparatus of criticism and discourse, would not be able to function; and the domain of public discourse would be swallowed up in private property, lawsuits and prosecutions. Perhaps there is a lesson here for those who would seek to erode the extent of fair use rights. .

Lord Coalston's judgement drew heavily on the idea that property had to be corporeal, or, if incorporeal, be in a corporeal thing - an analysis which may well not stand close scrutiny today; but, significantly, there is also a keen sense of the public domain - "the necessary consequence of publication is to make [the work] common to all mankind", and, later, he warns against the dangers of a perpetual monopoly.

Lord Alva was unable to see any trace of a perpetual right of literary property in Roman law¹⁷. Such a right did not form any part of the common law of Scotland. So far as it exists, it is a creature of statute:

"and therefore it can go no farther with us than it is carried by the statute; which I will gladly give force to, because it goes as far as, I think, justice and the encouragement of learning and industry, require. And I do not envy any other state or country, where either common law or statute may have carried it farther."

Lord President Dundas also could see no basis for a common law right of property distinct from

¹⁷ which heavily influences Scots law.

the temporary right conferred by the Statute of Anne.

The most eloquent opinion was that of Lord Kames, who, as we have seen, was an author himself and who was to go on to strike the great blow for liberty in *Knight v Wedderburn*.

His Opinion opens by underlining that the Court of Session is not bound by the decisions of the English Courts and then goes on to express the same doubt as Lord Coalston concerning property which was not itself either corporeal or in a thing which is corporeal. It is no more than a privilege or a monopoly. There is such a privilege or monopoly granted by the Statute of Anne, but it is a limited one:

"But to follow out the common law. The composer of a valuable book has great merit with respect to the public: his proper reward is approbation and praise, and he seldom fails of that reward. But what is it that intitles him to a pecuniary reward? If he be intitled, the composer of a picture, of a machine, and the inventor of every useful art, is equally intitled. Such a monopoly, so far from being founded on common law, is contradictory to the first principles of society. Why was man made a social being, but to benefit by society, and to partake of all the improvements of society in its progress toward perfection? At the same time, he was made an imitative being, in order to follow what he sees done by others. But to bestow on inventors the monopoly of their productions, would in effect counteract the designs of Providence, in making man a social and imitative being: it would be a miserable cramp upon improvements, and prevent the general use of them. Consider the plough, the loom, the spinning wheel. Would it not sound oddly, that it would be rank injustice for any man to employ these useful machines, without consent of the original inventors and those deriving right from them. At that rate, it would be in the power of the inventors to deprive mankind both of food and raiment. The gelding of cattle for food, was not known at the siege of Troy. Was the inventor entitled to a monopoly so as to bar others from gelding their cattle?"

These words look beyond copyright alone and eloquently justify why it is that patents (a development which then lay largely in the future), also require to be limited in duration, and, in these words, one sees the whole philosophical basis of Free and Open Source software: for society to function properly, there is need to share creative fruits. A perpetual monopoly would be a "miserable cramp on improvements and prevent the general use of them." The parallel with what tends to happen in the world of proprietary software is too obvious to need spelling out; and, perhaps by way of example, we might think of the strategic objectives of Microsoft in distorting the EU market through its anti-competitive practices such as the withholding of interfacing information, the bundling of browsers and the like.

Lord Kames continues:

"What shall be said of the art of printing? If the monopoly of this useful art was to be perpetual, it would be a sad case for learned men, and for the interest of learning in general: it would enhance the price of books far beyond the reach of ordinary readers. Such a monopoly would raise a fund sufficient to purchase a great kingdom. The works alone of Shakespeare, or of Milton, would be a vast estate."

And, of course, through its dominant position in providing operating systems, which are as critical to the running of computers as printing is to the publishing of books, Microsoft has traditionally commanded high prices and has "raised funds sufficient to purchase a great kingdom", or at any rate to have a great deal of influence in many kingdoms.

Lord Kames then proceeds to consider the effect that such a right of property would have on the freedom of movement of goods:

"It is a rule in all laws, that the commerce of moveables ought to be free; and yet, according to the pursuer's doctrine, the property of moveables may be subjected to endless limitations and restrictions that hitherto have not been thought of, and would render the commerce of moveables extremely hazardous."

This, too, remains relevant. The underlying basis in the EU Treaties for the powers exercised by the European Commission both in the *Café Hag* case¹⁸ and in its dealings with Microsoft over that company's anti-competitive practices lay in the extent to which those large and dominant enterprises interfered with the free movement of goods and services throughout the EU.

Then, Lord Kames turned to the wider business and economic realities:

"Lastly, I shall consider a perpetual monopoly in a commercial view. The act of Queen Anne is contrived with great judgment, not only for the benefit of authors, but for the benefit of learning in general. It excites men of genius to exert their talents for composition; and it multiplies books both of instruction and amusement. And when, upon expiration of the monopoly, the commerce of these books is laid open to all, their cheapness, from a concurrence of many editors, is singularly beneficial to the public. Attend, on the other hand, to the consequences of a perpetual monopoly. Like all other monopolies, it will unavoidably raise the price of good books beyond the reach of ordinary readers. They will be sold like so many valuable pictures. The sale will be confined to a few learned men who have money to spare, and to a few rich men who buy out of vanity as they buy a diamond or a fine coat. The commerce of books will be in a worse state than before printing was invented: at that time, manuscript copies might be multiplied at pleasure; but even manuscript copies would be unlawful if there were a perpetual monopoly. Fashions at the same time, are variable; and books, even the most splendid, would wear out of fashion with men of opulence, and be despised as antiquated furniture. The commerce of books would of course be at an end; for even with respect to men of taste, their number is so small, as of themselves not to afford encouragement for the most frugal edition. Thus booksellers, by grasping too much, would lose their trade altogether; and men of genius would be quite discouraged from writing, as no price can be afforded for an unfashionable commodity. In a word, I have no difficulty to maintain that a perpetual monopoly of books would prove more destructive to learning, and even to authors, than a second irruption of Goths and Vandals. And hence with assurance I infer, that a perpetual monopoly is not a branch of the common law or of the law of nature. God planted that law in our hearts for the good of society; and it is too wisely contrived to be in any case productive of mischief."

It is the way of would-be monopolists that their reach has always exceeded their grasp. By driving prices up, they limit the market. The more limited the market becomes, the less healthy it becomes, relying on a fickle base which moves on to the latest new thing, leaving in its wake orphaned works. There is not a healthy base for a vigorous market, so what is the point of creating any more? In the long run, monopolies bring contraction and decay in economic activity. To those who see this as their economic model, it is perhaps sufficient to caution them that they should be careful what they wish for, lest they get it.

18 For background, see C. Morcom, "Trademarks in the European Community after *Café Hag II*" 81 Trademark Rep. 534 (1991) pp534 - 553

Lord Kames concludes his Opinion by underlining the wider interests of society as a higher good than the short-term profit of the economically powerful, and asserts the calling of the Court to take that wider view:

"Our booksellers, it is true, aiming at present profit, may not think themselves much concerned about futurity. But it belongs to judges to look forward; and it deserves to be duly pondered whether the interest of literature in general ought to be sacrificed to the pecuniary interest of a few individuals. The greatest profit to them ought to be rejected, unless the monopoly be founded in common law beyond all objection: the most sanguine partizan of the booksellers will not pretend this to be the case. At the same time, it will be found, upon the strictest examination, that the profit of such a monopoly would not rise much above what is afforded by the statute. There are not many books that have so long a run as fourteen years; and the success of books upon the first publication is so uncertain, that a bookseller will give very little more for a perpetuity, than for the temporary privilege bestowed by the statute. This was foreseen by the legislature; and the privilege was wisely confined to fourteen years; a sufficient encouragement to men of genius without hurting the public interest. The best authors write for fame: the more diffused their works are, the more joy they have. The monopoly then is useful only to those who write for money or for bread, who are not always of the most dignified sort. Such writers will gain very little by the monopoly; and whatever they may gain at present, the profits will not be of long endurance; a monopoly would put a final end to the commerce of books in a few generations. And therefore, I am for dismissing this process as contrary to law, as ruinous to the public interest, and as prohibited by the statute."

Though in those remarks there is something of the disdain of the gentleman scholar for the jobbing writer, that was because Lord Kames was a product of his time. The underlying message that the courts, and, it may be suggested, the legislators, should always have before them as a critical consideration the wider good of society is a lesson which is eternal, though so often unheeded today.

5. Postscript

Boswell's report of *Hinton v Donaldson* was published in time to be cited by Donaldson's counsel in the Appeal to the House of Lords in the proceedings brought in England against him by Beckett¹⁹, where, as noted above, the lower courts had granted an injunction. Donaldson's Appeal in that case was successful, and the injunction was recalled²⁰.

Robert Forbes, Bishop of Ross and Caithness, recorded in his journal entry for 26th February, 1774, that, when the news of Donaldson's victory in the House of Lords reached Scotland, it occasioned:

"great rejoicings in Edinburgh upon victory over literary property; bonfires and illuminations, ordered tho' by a mob, with drum and 2 fifes²¹."

19 A full report is to be found in Cobbett, *Parliamentary History of England*, vol. XVII/KET (London, 1806-1820) and online at: <http://www.copyrighthistory.com/donaldson.html>

20 The decision in *Hinton v Donaldson* and the subsequent English decision in the House of Lords firmly shut the door on anyone trying to argue that there is such a thing as perpetual Intellectual Property. The fact that they are never cited, nor even particularly remembered today is, paradoxically, a mark not of their irrelevance but, rather, of their magisterial importance and continuing relevance.

21 Rev. Robert Forbes, A.M., Bishop of Ross and Caithness 1746-1775, (Henry Paton, ed.) *The Lyon in Mourning or a Collection of Speeches Letters Journals etc. Relative to the Affairs of Prince Charles Edward Stuart*, vol. 3, p. 294, (Edinburgh, 1896)

Perhaps we no longer live in an age when judicial decisions about copyright lead to bonfires and dancing in the streets, but if we could be confident that today's legislators would not allow themselves to be persuaded by narrow commercial interests, the promoters of which, in the words of Lord Kames, "aiming at present profit, may not think themselves much concerned about futurity" and, instead were to look forward and "duly ponder whether the interest of literature in general ought to be sacrificed to the pecuniary interest of a few individuals", then, and only then might there be cause for public rejoicing.

About the author

Iain G. Mitchell QC is Chairman of the Scottish Society for Computers and Law; Chairman of the Scottish Lawyers' European Group; Chairman of the Faculty of Advocates IT Group; Lecturer, Honorary Board of Lecturers, Institut für Informations-, Telekommunikations- und Medienrecht, Westfälische Wilhelms- Universität, Münster; and Freeman, Worshipful Company of Information Technologists.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Mitchell QC, Iain G (2009) 'BACK TO THE FUTURE: Hinton v Donaldson, Wood and Meurose (Court of Session, Scotland, 28th July, 1773)', *IFOSS L. Rev.*, 1(2), 111 – 122

DOI: [10.5033/ifosslr.v1i2.23](https://doi.org/10.5033/ifosslr.v1i2.23)

Copyright © 2009 Iain G. Mitchell QC.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Case law report: A look at *EDU 4 v. AFPA*, also known as the “Paris GPL case”

Martin von Willebrand^a

(a) HH Partners, Attorneys-at-law; Validos.

DOI: [10.5033/ifoossr.v1i2.17](https://doi.org/10.5033/ifoossr.v1i2.17)

Abstract

On 22nd September 2009 many Internet news sources noted a decision by the Appeal Court of Paris¹. The coverage generally described the decision as “an important case on GPL license enforceability”. This brief article examines the decision² and discusses its relationship to GPL license enforcement specifically and open source licensing in general. The account is based on review of the eight-page decision only and some legal conclusions are offered by considering the judgement in relation to the current situation in other jurisdictions, such as Finland.

Keywords

Law; GNU General Public Licence (GPL); Paris; France; Europe; Licensing; Free and Open Source Software; Contract

Info

This item is part of the [Case law reports](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

Introduction

The decision concerns a contract dispute regarding an IT project. In 2000 EDU 4 won a contract and agreed to deliver software to AFPA. From the Appeal Court decision, it is not entirely clear which party was the original claimant in the case, but the court of first instance had already ruled on substantive claims of both parties in 2004, even though the claims by AFPA may have been presented only for defence. In a series of many turns the parties presented their latest claims to the appeal court in April 2009. In essence, EDU 4 claimed not to have breached the IT project contract and that it was entitled to all payments, while AFPA claimed breach of contract and that the vendor was not entitled to any further payments and that early termination of the contract was justified. I will not look into the details of the contractual claims, since these claims are outside the scope of this article.

-
- 1 Ryan Paul, “Big GPL copyright enforcement win in Paris Court of Appeals”, *Ars Technica*, 22 September 2009, <http://arstechnica.com/open-source/news/2009/09/big-gpl-copyright-enforcement-win-in-paris-court-of-appeals.ars> (retrieved 14 October 2009); FSF France, “Paris Court of Appeals condemns Edu4 for violating the GNU General Public License”, 22 September 2009, <http://fsffrance.org/news/article2009-09-22.en.html> (retrieved 14 October 2009).
 - 2 Cour d’Appel de Paris, Pôle 5, Chambre 10, no: 294, issued on 16 September 2009, available at <http://fsffrance.org/news/arret-ca-paris-16.09.2009.pdf> (retrieved 14 October 2009).

No GPL License Enforcement

Not a single claim (by either party), as cited in the court’s decision, is based on (i) the GPL, (ii) any interpretation of the GPL or any of its terms as a contract, or (iii) any right established by law, such as the French Code on Intellectual Property (*Code de la propriété intellectuelle*), which contains the provisions establishing copyright, or the right of the author, in France.

The decision cites the GPL a number of times, mostly to describe VNC software which was included in the delivery. The decision also discusses whether the contract permitted inclusion of free software in the deliverables of the project.

The following argument is probably most relevant to the question of GPL enforcement (page 8):

“Considérant qu’il résulte de l’ensemble de ces éléments que la société EDU 4 a manqué à ses obligations contractuelles en livrant en décembre 2001, date à laquelle devait s’apprécier sa conformité, un produit, d’une part qui présentait pour les utilisateurs des EOF des risques d’atteinte à la vie privée, d’autre part qui ne satisfait pas aux termes de la licence GNU GPL puisque la société EDU 4 avait fait disparaître les copyrights d’origine de VNC sur les propriétés de deux fichiers en les remplaçant par les siens et avait supprimé le texte de la licence;”

A translation of the above citation into English follows:

“[The court] considers that it follows from all of these elements that the entity EDU 4 had not fulfilled its contractual obligations with its delivery in December 2001, the date on which the performance of EDU 4 was to be assessed, that on the one hand posed privacy risks to the users of EOF and on the other hand did not satisfy the terms of the GNU GPL license, since the entity EDU 4 had removed the original copyright notices of VNC from two files, replacing them with its own copyright notices, and since it had deleted the text of the license;”

One can conclude that the above paragraph does not concern enforcement of the GPL, but rather appreciation of fulfilment of the contract between EDU 4 and AFPA. To the extent that the decision relates to the terms of the GPL, it is very limited in reach. It may be noted that the removal of copyright and similar notices could be considered violations of law, and not merely violation of GPL license requirements.

User Claim Based on Non-GPL Contract

News coverage of the case made much of the fact that a software user had presented claims, rather than the copyright holder (as it is generally assumed that the GPL is enforceable by upstream licensors). However, these claims were not based on the GPL, but on a separate contract between the user and the distributor. Furthermore, the claims brought by AFPA had no relevance to GPL license requirements. Those claims were related to an IT project, i.e., whether the early termination of the project was justified or not.

Source Code to Modifications

The court notes at one point that EDU 4 did not deliver the source code to the modifications it had made to the software, although EDU 4 had committed to do so in one of its letters. It is not clear from the decision text whether EDU 4 explicitly mentioned delivery of the source code in its letter

or merely referred to a delivery satisfying the terms of the GPL. In the latter case, the court would have concluded that satisfying the GPL upon redistribution required delivery of the source of the modifications to the software. This is probably not so, since the conclusion, as such, would be slightly inaccurate. The license requirements of GPL version 2 can be satisfied without delivering the source code, for example by providing an offer to deliver the source code.

Open Source License Enforcement

As to the relevance of this decision to enforcement, the decision shows that the court considered the software to be licensed under the terms of the GPL and attached legal significance to the terms. On the other hand, the GPL and its terms play a very limited role in this decision, and thus I would hesitate to attach almost any legal significance to this decision as regards enforceability of the GPL. However, this view is based on my understanding that open source licenses are generally enforceable by their copyright holders.

Under Finnish law and other Nordic laws (Swedish, Danish and Norwegian), open source licenses can generally be considered enforceable. Under these laws the analysis is based on national Copyright Acts and the fact that nothing else, except the license, allows deviation from the monopoly granted to the author by the Acts. Since, for example, the right to modify the work and the right to distribute the work to the public are exclusively reserved to the author under applicable law, it is in the freedom of the author to decide on the manner of granting broader or narrower license rights to the work regarding modifications and distribution. In addition, there are no additional requirements set by law to dealings of private nature, such as a granting of a copyright license or entering into a contract, in order for them to attain enforceability.

There are differences in enforcement options, but that is beyond the scope of this article. My assertion that open source licenses are, in general, enforceable under Nordic laws does not mean that, in a particular case, each element of all licenses could be enforced. But this question might arise solely from the differences in expectations of the parties, as there are ambiguities in the formulations of some licenses, in which case it is no longer a pure question of eligibility for enforcement.

However, even a casual reader of open source-related legal writings on the Internet will encounter the question of whether open source licenses are enforceable and, if so, under what legal theory and through what practical means of enforcement. This discussion will persist, since my understanding is that some jurisdictions do impose requirements for legal transactions in order for them to become enforceable at all. This is probably most evidently the case for common-law jurisdictions, such as England and Wales³ and the federal and local jurisdictions within the United States⁴, in which some type of consideration seems to have a role in defining the legal nature of such transactions. The role and practical significance seems to vary by jurisdiction.

It might be due to the relatively frequent discussion of this common-law related question that commentators from other jurisdictions have begun to ask similar questions, even if they are not relevant to enforcement as such⁵.

3 See Mark Henley (2009) 'Jacobsen v Katzer and Kamind Associates – an English legal perspective', IFOSS L. Rev., 1(1), pp 41 – 44. Available at: <http://www.ifossr.org/ifossr/article/view/4>

4 See Lawrence Rosen, (2009) 'Bad facts make good law: the Jacobsen case and Open Source', IFOSS L. Rev., 1(1), pp 27 – 32. Available at: <http://www.ifossr.org/ifossr/article/view/5>

5 The relevance might not be purely enforcement-related, such as questions concerning license interpretation, or it might not be strictly open source or free software-related, such as questions concerning the enforcement of a very general provision on liability limitation.

Other Lessons

Open source or free software has gained significant ground during recent years. It should be noted that many open source projects have been widely used for a very long period of time, while there are open source projects that will never become mainstream due, e.g., to the limited scope of the projects or other issues. It is important to understand that it is not open source that makes good or bad software; it is the individual projects that are good or bad. The same is true of closed source projects: some are better, some are worse. But open source, as a licensing model, is becoming a mainstream model.

This licensing model should be handled like any other legal phenomenon: with a professional and unbiased approach.

Open source as a concept or phenomenon is already well known; what it means in practice and how it can be benefited from is less well understood. There are many incorrect beliefs that open source software is solely an area for hobbyists and amateurs. Although there are open source projects created by hobbyists and nonprofessionals, there are also many professional open source projects. The same applies to closed source projects.

This article has demonstrated that the news reported on the Internet concerning *EDU 4 v. AFPA* was, from a legal point of view, misleading⁶. It may be that the reporting suffered from erroneous understanding of the decision or was motivated by the desire for quick publicity and by an interest in the open source phenomenon. Some writers who reported this story undoubtedly strive to support the open source or free software phenomenon. But such support cannot be achieved by an unprofessional journalistic approach to matters of legal analysis. Unprofessionalism in such an area tends to reinforce the false belief that open source is solely a matter of non-professional software development. As a licensing model, open source should be treated like any other area of human life; legal analysis of open source issues should be guided by professionals.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

von Willebrand, Martin (2009) 'A look at *EDU 4 v. AFPA*, also known as the “Paris GPL case”', *IFOSS L. Rev.*, 1(2), pp 123 – 126
DOI: [10.5033/ifosslr.v1i2.17](https://doi.org/10.5033/ifosslr.v1i2.17)

Copyright © 2009 Martin von Willebrand.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



⁶ Of course, one might say that it is not at all unusual that legal questions raised on the internet are handled in a misleading, incorrect or even false way, whether they relate to open source or something else.

Book review: “Law and the Internet” - 3rd Edition, edited by Lilian Edwards and Charlotte Waelde

Andrew Katz^a

(a) Moorcrofts LLP

DOI: [10.5033/ifosslr.v1i2.18](https://doi.org/10.5033/ifosslr.v1i2.18)

Abstract

Andrew Katz reviews the Third Edition of Law and the Internet (ed. Edwards and Waelde, 2009), with particular emphasis on chapters 10, 11 and 12, covering the legal protection of computer software, free and open source software, and open access, respectively.

Keywords

Book review; Law; information technology; open access; internet law; Free and Open Source Software

Info

This item is part of the [Book Reviews](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

Hart Publishing, 2009. RRP £30

ISBN: 9781841138152

Since its first publication in 1997 (under the title *Law and the Internet: a Foundation for Electronic Commerce*), this book has experienced one intervening edition, expanded in girth and scope, and dropped the restrictive shackle of the subtitle “a Foundation for Electronic Commerce”. Although some chapters of the book have been updated, most are entirely new and in the same way that a 1997 VW Polo shares little in common with its 2009 descendant, *Law and the Internet* is, under the skin, far removed from its predecessors.

The 22 chapters in the current edition, while arranged in a logical order, are otherwise almost entirely disconnected from one another, and each can be read as a stand-alone essay by its author. Whether you find this inconsistency irritating or charming is a matter for the individual reader. I think it unlikely that any reader is likely to start at chapter one and work her way through the book linearly until the end, so in practice the lack of consistency is unlikely to be an issue. The chapters are of a generally high standard and well annotated.

The inconsistency does mean that some chapters read as textbook-like factual distillations, and some read more as academic papers exploring and arguing a point more rigorously. Some (like Andrés Guadamuz's chapter *Free and Open-Source Software*) manage to switch from one style to the other mid-stream. I do not regard this as a major issue.

The chapters most of interest to free and open source lawyers are chapters 10, 11 and 12, which deal with protection of computer software, free and open source software, and open access, respectively. It's arguable that they should not feature in a book on law and the internet at all, but in their absence there would be no review. In any event, the internet runs on an infrastructure powered largely by free and open source software, and the internet has been a critical element in the success of FOSS, so I'll allow the connection, at least to chapters 10 and 11, and move on.

Protection of Computer Software by cellist Arne Kolb, is interesting as an artefact from a parallel universe, possibly one in which Richard Stallman became a TV chef and never had the opportunity to interact with a computer. The first word of the title reveals the stance that Kolb takes throughout the rest of the piece, and indeed we learn that "...software developers...[want] strong legal protection for their own software...". The piece goes on to explore traditional views of copyright licensing, copyright infringement (including look and feel) and patent licensing. Whilst reasonably up to date, it takes a very narrow view of the subject. For example, it is apparently the case that "the developer...has an interest in preventing the creation of software with a similar 'look and feel'": this may have been true in the days of 80 column displays, but since the development of commercial GUIs (like the Mac in 1984), the proprietors of those GUIs (such as Microsoft and Apple) have been at pains to ensure a consistency of look-and-feel between apps running on their respective platforms, and have published style guides accordingly.

Free and Open-Source Software by Andrés Guadamuz is, in contrast, a well-researched, easily readable introduction to FOSS, including a history and a brief analysis of the philosophical differences between free software and open source. It explains licence ecology (although he describes the BSD licence as having an "assignment of rights" clause: I'm informed by Scottish colleagues that "assignment" is a synonym for "assignment", and there is no assignment in the BSD licence).

Other licences briefly covered are MIT and Apache 2.0. There is a reference to Creative Commons (although it would have been helpful to explain that this is really a suite of licences, many of which are not "free" or "open" at all). GPL v2 is covered in a little more depth, although Guadamuz misleadingly talks (with respect to section 2b) about "restrictions against using the software to create commercial software". If, by "commercial," he means what is generally called proprietary (or closed source) software, then this statement makes a little more sense than if he gives "commercial" the more common (if disputed) meaning used by the Creative Commons, for example. He also describes the LGPL as being virtually identical to the GPL, sans copyleft clause. There are pairs of copyleft/non-copyleft licences which could be described as virtually identical in this way (the Open Software Licence and the Academic Free Licence being an example), but not the GPL/LGPL: there are many differences in text between GPLv2 and LGPL v2.1. LGPL v2.1 does contain a copyleft clause, but it is weaker than that contained in GPL v2.

GPL v3 is also covered in greater depth (and justifiably criticised).

Guadamuz has written at some length about the consideration problem: namely, are FOSS licences contracts or bare licences? Six pages of the 32 page chapter are devoted to this issue. Since this is clearly a topic close to his heart, Guadamuz changes gear here and stops informing us about FOSS, and starts arguing his pro-contract view. I happen to disagree with him, but his arguments are interesting and worthy of further analysis. However, I'll restrain myself from doing such an analysis here for the same reason – lack of space – which Guadamuz should possibly have considered when devoting about a fifth of his allotted pagecount to this issue. Having said that, it's an important, and frequently overlooked topic, and a two page analysis of this topic would probably have been more in keeping with the remainder of the chapter than an out-of-place attempt

to persuade the Court of Appeals for the Federal Circuit that allowing the appeal in *Jacobsen v Katzer* was wrong. The four freed pages could have been devoted to some more analysis of patent and trademark issues.

The rest of the chapter reverts to Guadamuz's clear explanatory style. GPL-violations.org gets some coverage in the "enforcement" section, although it's inaccurately described as a non-profit branch of the Free Software Foundation – it certainly works closely with the Free Software Foundation Europe, but is independent of it, and Harald Welte is not so much a "main supporter" of GPL-violations.org, as its founder. Despite this, the reference to Welte and his work in this chapter is welcome. The SCO cases are briefly discussed, as is *Wallace v IBM*.

Finally, Guadamuz gives us some useful information to place FOSS in context from a practical perspective, and some indication of how he feels the future lies for FOSS.

In general, then, the chapter is an excellent and up-to-date introduction to FOSS and most of the main legal issues concerning it, and despite the factual quibbles, a lack of more commentary on patents and trade marks, and some over-analysis of the contract/licence issue, I highly recommend it to any lawyer or law student seeking a brief introduction to the legal issues behind open source.

Charlotte Waelde's chapter, *Scholarly Communications and New Technologies: the Role of Copyright in the Open Access Movement*, covers a very specific issue in greater depth: namely making scholarly articles and journals freely available, in both senses of the word "freely" with which the readers of this article will be familiar. The path of the open access movement parallels that of the free and open source movement (save that scholarly journals appeared a little earlier than software: apparently the first academic journal "Philosophical Transactions", was published in 1665, and remains in print to this day¹). The article suggests that the first significant date for Open Access was 1966, when ERIC – the Educational Resources Information Center was launched in the US to make bibliographic information relating to journal articles freely available. The chapter as a whole is a fascinating romp through the cultural, legal and economic issues affecting academic publishing, and whilst of little direct relevance to free and open source software, it's interesting to learn what fellow-travellers are up to.

Law and the Internet is the Las Vegas five-star buffet of internet law. The dishes are fresh and generally well-prepared and of high quality, although the breadth of choice does mean that there is a little inconsistency in execution and style from one to the other. However, all hungry readers will find that, if they select wisely, they will come away with a personal selection which will satisfy them well.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Katz, Andrew (2009) "Law and the Internet" - 3rd Edition, edited by Lilian Edwards and Charlotte Waelde', *IFOSS L. Rev.*, 1(2), pp 127 – 130

DOI: [10.5033/ifosslr.v1i2.18](https://doi.org/10.5033/ifosslr.v1i2.18)

Copyright © 2009 Andrew Katz.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Patents and their effect on Standards: Open video codecs for HTML5

Silvia Pfeiffer^a

*(a) Dr Silvia Pfeiffer is CEO at Vquence Pty Ltd Australia,
works within the non-profit open source software organisation
Xiph.Org, and has been a contractor for Mozilla on technical
issues around video for the Web*

DOI: [10.5033/ifosslr.v1i2.21](https://doi.org/10.5033/ifosslr.v1i2.21)

Abstract

HTML5 is an updated version of the hypertext markup language that has been empowering the World Wide Web for the last 20 years. One of the things that HTML5 introduces is a <video> element, which make video content as simple to include into Web pages as images. Similar to the issues that had to be overcome with the introduction of the tag in 1993, we are now facing the issue of a common baseline codec for the <video> element – a format that all browser vendors can implement from a rights point of view and will agree to implement. Ogg Theora/Vorbis has been proposed as a solution, but only Mozilla Firefox and Google Chrome have agreed to implement it. Right now, we are at an impasse as the patent situation around video codecs is unclear and different parties take a different stance. This article discusses the issues in more detail.

Keywords

Standards, patents, open source, open standards, submarine patents

Info

This item is part of the [Tech Watch](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

Introduction

On February 28th, 2007, Opera proposed the introduction of a video element into HTML5¹. The <video> element, similar to its counterpart the <audio> element, aims at vastly simplifying the use of audio and video content on the World Wide Web. When implemented, there will not be a need to install plugins into your Web Browser of choice because it will support the <video> element out of the box. Such an element will further allow new forms of publishing of audio and video in a more integrated fashion with HTML. An example is the projection of a video on the surface of a cube which is being drawn in a <canvas> element. A whole new world of online applications is enabled when such elements exist.

A few weeks after the initial proposal of the <video> element, Opera CTO Håkon Wium Lie stated

¹ See email to the WHATWG mailing list by Opera dated 28th February 2007, <http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2007-February/009702.html>

in a talk given at Google:

"I believe very strongly, that we need to agree on some kind of baseline video format if [the video element] is going to succeed. [...] We want a freely implementable open standard to hold the content we put out. That's why we developed the PNG image format. [...] PNG [...] came late to the party. Therefore I think it's important that from the beginning we think about this."²

Thus was born the need for a common "baseline codec" for HTML5 video (audio always implied). A baseline codec is essentially a video (and audio) encoding format that is supported and implemented by all browser vendors.

Håkon further stated requirements for the video element as follows:

"It's important that the video format we choose can be supported by a wide range of devices and that it's royalty-free (RF). RF is a well-establish principle for W3C standards. The Ogg Theora format is a promising candidate which has been chosen by Wikipedia."³

Ogg Theora is a codec developed by the Xiph.org Foundation as an open source codec⁴. Theora is a derivative of a codec that was earlier developed by On2 Technologies under the name VP3⁵ and released as open source in September 2001⁶. The VP3 source code was originally published on www.vp3.com. This site was closed down when the code and its further development were moved to Xiph.Org. On2 Technologies is currently in the process of being acquired by Google.

The license under which On2 Technologies published the VP3 source code reads as follows:

"On2 represents and warrants that it shall not assert any rights relating to infringement of On2's registered patents, nor initiate any litigation asserting such rights, against any person who, or entity which utilizes the On2 VP3 Codec Software, including any use, distribution, and sale of said Software; which make changes, modifications, and improvements in said Software; and to use, distribute, and sell said changes as well as applications for other fields of use."⁷

This implies that On2 Technologies asserts to never pursue its patents on the Theora codebase being an improvement to the On2 VP3 Codec Software.

Further, Ogg Vorbis is an open source audio codec developed and published by Xiph.Org since about the year 2000. Vorbis was developed with a clear intention of only using techniques that were long out of patent protection. Vorbis has been in use by commercial applications for a decade now, including Microsoft software and many games.

After VP3 was published and turned into Theora, Ogg Theora with Vorbis as the audio track became the first unencumbered video/audio codec software.

2 See video of Håkon Wium Lie's Google talk at <http://video.google.com/videoplay?docid=5545573096553082541&ei=LV6hSaz0JpbA2AKh4OyPDg&hl=un>

3 See Håkon Wium Lie's page on the need for a video element at <http://people.opera.com/howcome/2007/video/>

4 See Xiph.Org's Website on Theora at <http://theora.org/>

5 See On2 Technologies' press release on the 24th June 2002 at http://www.on2.com/index.php?id=486&news_id=313

6 See On2 Technologies' press release on the 7th September 2001 at http://www.on2.com/index.php?id=486&news_id=364

7 See Xiph.Org source code repository at <http://svn.xiph.org/trunk/theora/LICENSE>

What exactly is an “open codec”?

The term "open codec" has been used in at least two substantially different environments: "open standards" and "open source".

In the open source environment, a firm requirement of open software is that it is published under a license that satisfies the open source definition⁸. In particular it contains the following sentence:

"The license shall not require a royalty or other fee <...>"

Thus, an open codec does not only consist of open source software, but also requires a royalty free license such that the code is actually usable in many different circumstances.

In the open standards environment, the definition of "open standard" is not as clear-cut⁹. Mostly, people regard a specification that has been developed by a committee of representatives from multiple organisations and published by a standards organisation as an "open standard", independent of whether it costs to purchase the specification or whether an implementation requires payment of royalties.

The World Wide Web Consortium (W3C), which is the standards body that is also publishing HTML, is an exception amongst standards bodies in that it seeks to issue only Recommendations that can be implemented on a Royalty-Free (RF) basis¹⁰. Thus, an open standard developed by other standards bodies may not be regarded as being open enough for the W3C. This is the case for the MPEG-4 H.264/AVC codec, which has been identified as a candidate for a video codec¹¹. H.264 has been approved as a standard jointly by the International Telecommunications Union (ITU) and the International Standards Organisation (ISO/IEC).

H.264 has a clear requirement for royalties to be paid on the large attached patent portfolio. At this point in time, publishers and distributors of H.264 content on the Internet are in a honeymoon phase where there are no royalties until 30th December 2010, but then, the royalty shall be “no more than the economic equivalent of royalties payable during the same time for free television”, which can reach up to \$5million annually depending on audience size¹².

While such licensing terms may seem reasonable and non-discriminatory (RAND) to ITU and ISO, many of the smaller Internet publishers, and in particular not-for-profit organisations, will find it hard to comply with such terms.

The honeymoon phase and the codec's high quality have made sure that H.264 now has a major share in the video formats that are being distributed over the Internet. However, with such license terms associated, it is not an acceptable format as a baseline codec for the HTML5 <video> element.

HTML5 baseline codec requirements

Opera's first experimental build of the HTML5 <video> element used Ogg Theora and Vorbis as

⁸ See Open Source Definition at http://en.wikipedia.org/wiki/Open_Source_Definition

⁹ See Open Standard Definition at http://en.wikipedia.org/wiki/Open_standard

¹⁰ See W3C RF requirements at <http://www.w3.org/Consortium/Patent-Policy-20030520.html#sec-Licensing>

¹¹ See W3C HTML Working Group Issue tracker, Issue #7 at <http://www.w3.org/html/wg/tracker/issues/7>

¹² See MPEG LA (License Authority) publication at http://www.mpegla.com/main/programs/AVC/Documents/AVC_TermsSummary.pdf

the video and audio codecs being inspired by Wikipedia's exclusive use of Ogg Theora/Vorbis. Consequently, the first specifications of video in the standard¹³ also included a suggestion of Ogg Theora/Vorbis as baseline codecs, with an expectation that this would encourage all browser vendors to make use of these open codecs:

“User agents should support Ogg Theora video and Ogg Vorbis audio, as well as the Ogg container format.”

However, by December 2007, it was clear to the editor of the HTML5 draft, Ian Hickson, that not all browser vendors were going to implement Ogg Theora and Vorbis support. He therefore replaced the should-requirement for Ogg Theora/Vorbis with the following:

“It would be helpful for interoperability if all browsers could support the same codecs. However, there are no known codecs that satisfy all the current players: we need a codec that is known to not require per-unit or per-distributor licensing, that is compatible with the open source development model, that is of sufficient quality as to be usable, and that is not an additional submarine patent risk for large companies. This is an ongoing issue and this section will be updated once more information is available.”¹⁴

This was supposed to bring the issue to the attention of the browser vendors and make them start solving it by clarifying the HTML5 requirements on a baseline codec. At that time, Ogg Theora was being criticised for its inferior quality over H.264, its lack of implementations on mobile devices, and its threat of submarine patents, which some vendors like Apple, Nokia, and later Microsoft used as an argument to not support it¹⁵.

So, Mozilla contributed some funding¹⁶ to have the core developers of Theora improve encoder quality, which has led to amazing progress - some of which still continues. With some currently deployed encoders – in particular with the YouTube encoder - Ogg Theora can now be regarded as almost on par with H.264 when it comes to video quality.

Further, initial implementations of Ogg Theora/Vorbis on mobile devices have emerged, proving the point that Ogg Theora is a much simpler codec than H.264 and therefore does not need as much special hardware support on small devices to make it usable. Many are still waiting for the day on which Ogg Theora/Vorbis video can be viewed on an iPhone or iPod, claiming these as the ultimate proof of portability. It is only a matter of market demand until a device vendor will step forward and offer Theora hardware decoding.

Theora now meets all of the requirements listed by the HTML5 editor bar the additional “submarine patent” risk. In contrast, the competing H.264 doesn't meet the license requirements and certainly the existence of submarine patents on H.264 also cannot be completely excluded.

The term “submarine patent” is being used rather loosely in the HTML5 community in reference both to patents that have not been identified yet as being infringed by a technology, as well as patents that have only been registered so recently that they are virtually unpublished and can

13 See Archive.org's June 2007 version of the HTML5 specification at <http://web.archive.org/web/20070629025435/http://www.w3.org/html/wg/html5/#video0>

14 See Ian Hickson's email in December 2007 to the WHATWG at <http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2007-December/013135.html>

15 See as an example this story in Apple Insider http://www.appleinsider.com/articles/09/07/06/ogg_theora_h_264_and_the_html_5_browser_squabble.html

16 See press release by Mozilla on 26th January 2009 at <http://blog.mozilla.com/blog/2009/01/26/in-support-of-open-video/>

therefore not be identified as being infringed. The latter is the real meaning of a “submarine patent” according to Wikipedia¹⁷, but both are being referred to as “submarine patents” in HTML5.

The risk of the existence of “submarine” patents on Theora is potentially slightly larger than on H.264, since many of the patent developing companies on video codecs were part of the H.264 development process and have their patents entered into the H.264 patent pool. The video patent developing companies, however, tend not to take part in W3C standardisation activities and therefore their patents are not covered by W3C. Thus, Theora supposedly poses a larger threat than any MPEG codec.

On the other hand, those patents are known and already licensed to Apple, Microsoft etc, so they would not pose additional risks to existing license holders. The number of as yet unknown “submarine” patents threatening Theora is possibly roughly the same as the ones threatening H.264. It is this risk, though, that continues to hold back Apple and Microsoft (and possibly others) from implementing Theora support in their hardware and software.

The current HTML5 video codec situation

With a goal of taking HTML5 to a Last Call status by the end of 2009, the HTML5 editor, Ian Hickson, had to clean out controversial aspects from the document. Seeing no progress on the baseline codec decision, on 29th June 2009, he removed the section on baseline codecs from the HTML5 specification altogether¹⁸.

By then, the situation had gotten worse: video element support had been implemented in Mozilla Firefox, Apple Safari/Webkit, Google Chrome, and Opera, but each browser vendor had done their own analysis of the situation at hand and different baseline codecs had been chosen. While Mozilla and Opera only supported Ogg Theora/Vorbis, Google decided to support both, Ogg Theora/Vorbis and H.264, and Apple decided to support only H.264¹⁹.

Mozilla shipped Ogg Theora/Vorbis with their release of Firefox 3.5 in June 2009 after having undertaken their own patent research and risk analysis²⁰. Unfortunately, the outcome of this research has not been published.

Google also did their own patent research, which Chris De Bona outlined in a post to the HTML5 mailing list²¹ on "Google's use of FFmpeg in Chromium and Chrome". FFmpeg in Chrome supports both, Ogg Theora/Vorbis and H.264. This is not a problem for Ogg Theora/Vorbis. However, it has been identified in that email thread that it seems to be a problem for people wanting video element support, but not having acquired their own license for H.264 support. It seems, Google themselves are protected by their license of H.264, but their users aren't. To the interested reader with a legal background, that email thread may be an informative read.

Opera continue development of Ogg Theora support and are shying away from H.264 for its license costs²². Their currently released browser does not support the video element, but it is

17 See Wikipedia article on Submarine patents at http://en.wikipedia.org/wiki/Submarine_patent

18 See email of Ian Hickson to WHATWG at <http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2009-June/020620.html>

19 See Opera blog post about video at <http://my.opera.com/ODIN/blog/video-on-the-web>

20 See Mozilla press release at <http://www.mozilla.com/en-US/firefox/3.5/releasesnotes/>

21 See Chris di Bona's email to WHATWG at <http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2009-June/020035.html>

22 See email to WHATWG at <http://lists.whatwg.org/pipermail/whatwg-whatwg.org/2009-June/020620.html>

expected that the next release will support it with Ogg Theora as the only codec²³.

Microsoft is a “fan” of the audio and video elements²⁴, but have early on in the discussion about baseline codecs stated that “Our legal people do not believe that Ogg Theora has much of any chance of being free from patent issues”²⁵. It seems they share the views of Apple and Nokia in this respect. They haven't spoken up about what baseline codec Internet Explorer is planning to support.

Apple asserts that Theora holds a larger “submarine” patent risk than any of the video codecs that have come through standards bodies and therefore refuses to implement native support in its applications and hardware devices for it²⁶. Apple has concerns because of issues raised by Nokia's video patent expert²⁷ and because the license that Xiph received from On2 may not apply to third party implementations of Theora²⁸. Nobody knows what the recent \$106M acquisition of On2 Technologies by Google²⁹ may change in this situation. Google has not made any statements about the codecs they are acquiring and what their intentions are.

Further, Apple is worried about the additions that Xiph.Org made to VP3 to turn it into the current version of Theora, since these could be covered by real submarine patents³⁰. The changes are all publicly listed³¹ and consist to a large extent of bug fixes, though some more interesting extensions have been contributed. A patent analysis of the major changes has not been published.

Apple also claims that because video codecs influence all of Apple's business, it would be a big target for patent trolls, while Mozilla's Firefox and Google's Chrome businesses are both too small for a patent troll to be encouraged to step forward at this stage. Now, if YouTube converted from H.264 to Ogg Theora that would be a different situation. Interestingly, several other video hosting sites, amongst them Dailymotion³², Wikimedia³³, and Archive.org³⁴ already support Ogg Theora/Vorbis and many smaller community video sites have also decided to go with this simple and usable solution.

Apple would prefer for Theora and Vorbis to be ratified by a standards body with involvement from the main video and audio codec patent holders. However, such a process would introduce so much delay as to exclude Theora and Vorbis as codec solutions in HTML5.

Unfortunately, the W3C is not able to employ lawyers to undertake an independent patent assessment for any codec, to expose “submarine” patents. However, the W3C could issue a call for a contribution of baseline codecs and go through a short but formal selection process, which would require patent holders to step forward. This would require inclusion of other standards bodies to ascertain that everyone had the chance to speak up in due time. We will see if such a process is possible and if indeed it will satisfy Apple, Nokia and Microsoft.

23 See tweet from Opera video element developer <http://twitter.com/foolip/status/6923376494>

24 See email to W3C as discussed and linked at <http://www.osnews.com/story/22182>

25 See W3C HTML Working Group Issue tracker, Issue #7 at <http://www.w3.org/html/wg/tracker/issues/7>

26 See W3C HTML Working Group Issue tracker, Issue #7 at <http://www.w3.org/html/wg/tracker/issues/7>

27 See Nokia submission to a W3C workshop on video for the Web at

<http://www.w3.org/2007/08/video/positions/Nokia.pdf>

28 See email from Apple to Xiph.Org at <http://lists.xiph.org/pipermail/theora/2009-July/002415.html>

29 See discussion of On2 Technologies acquisition by Google at <http://www.techcrunch.com/2009/08/05/google-acquires-video-compression-technology-company-on2-for-106-million/>

30 See email from Apple to Xiph.Org at <http://lists.xiph.org/pipermail/theora/2009-July/002415.html>

31 See Xiph.Org version control system at <http://svn.xiph.org/trunk/theora/CHANGES>

32 See Wikipedia article on Dailymotion <http://openvideo.dailymotion.com/en>

33 See Wikipedia article on Wikimedia <http://commons.wikimedia.org/wiki/Category:Video>

34 See Wikipedia article on Archive.Org <http://www.archive.org/details/movies>

In the meantime, open source software developers continue to bridge technology gaps. It is currently possible to run Ogg Theora/Vorbis content in every browser using the Cortado³⁵ Java applet. This is a similar approach to a Flash plugin with the advantage that the applet is loaded automatically and there is no installation process. A better solution is available for Safari/Webkit with a QuickTime plugin called XiphQT³⁶. Once installed, Ogg Theora/Vorbis will be supported in Safari/Webkit. For Internet Explorer, an ActiveX control is in development that will also be usable in the same way.

Conclusion

It seems that in the near future, all technical challenges for universal support of Ogg Theora/Vorbis as a baseline codec for HTML5 video and audio will be solved – either through native browser support or through extensions. Whether the legal situation will be solved in parallel is doubtful. It is, however, clear that unless the H.264 patent holders radically change their stance on licensing, H.264 will not be a viable contender as a royalty free baseline codec for HTML5.

Further changes to the situation could come from currently proprietary closed source codecs - e.g. if On2/Google contributed a new codec such as VP8 to the discussion by open sourcing it and making its patents available under a royalty free license – or further even indemnify all users of the codec from and against any and all liability.

In the meantime, the HTML5 editor is still waiting for the browser vendors to come to an agreement on a baseline codec that satisfies the requirements. Until such a time, the patent situation with video codecs continues to hold back the standardisation and broad uptake of a fundamental section of HTML5.

Disclaimer

The views expressed in this article are my personal views, not my company's, Xiph.Org's or Mozilla's. I am not a lawyer and this is not a legal analysis. It is only provided to give a background on the situation of the HTML5 standardisation work for video.

About the author

Dr Silvia Pfeiffer is CEO at Vquence Pty Ltd Australia, works within the non-profit open source software organisation Xiph.Org, and has been a contractor for Mozilla on technical issues around video for the Web.

³⁵ See Xiph.Org site at <http://www.theora.org/cortado/>

³⁶ See Xiph.Org site at <http://www.xiph.org/quicktime/>

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Pfeiffer, Dr Silvia (2009) 'Patents and their effect on Standards: Open video codecs for HTML5', *IFOSS L. Rev.*, 1(2), pp 131 – 138

DOI: [10.5033/ifosslr.v1i2.21](https://doi.org/10.5033/ifosslr.v1i2.21)

Copyright © 2010 Dr Silvia Pfeiffer.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Five Reasons To Care About Mobile Cloud Computing

Fabrizio Capobianco^a

(a) CEO, Funambol Inc., Free Software entrepreneur

DOI: [10.5033/ifossr.v1i2.24](https://doi.org/10.5033/ifossr.v1i2.24)

Abstract

Funambol Inc.'s Fabrizio Capobianco explores the technical and market opportunities presented by the intersection of cloud technologies and rapid advances in mobile computing for Free & Open Source software. In doing so he exposes the unique needs of this new market segment, and the legal challenges which may arise from its development.

Keywords

Cloud computing, mobile computing, AGPL

Info

This item is part of the [Tech Watch](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

Mobile cloud computing represents an opportunity for the free/libre open source software movement that is just as big and radical as cloud computing, maybe even more so.

The term 'cloud computing' has been defined by the U.S. Government's National Institute of Standards and Technology¹ as “a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.”

There are ongoing efforts to standardize cloud computing (Open Cloud Manifesto) but they seem to lack use cases about mobile computing. Cloud computing becomes mobile when a mobile device tries to access a shared pool of computing resources on-demand. There are at least 5 reasons why mobile cloud computing is important for free and open source software.

1. Mobile cloud computing is big in size

At the end of 2009, mobile phones were four billion. By 2013, that number is projected to grow to 6 billion. That is many times the number of personal computers. And when we start including in the mobile world other Internet capable devices, like ebook readers, photo frames, printers, photo and video cameras, personal navigators, the numbers go way up. Small portable devices that can

¹ Full document of the NIST Cloud Computing <http://csrc.nist.gov/groups/SNS/cloud-computing/>.

access information are already part of everyday life for hundreds of millions of people in the developed world. Also, many hints point to the fact that developing countries will be using the mobile cloud before they get to the 'regular' one.

Just as Free/Libre Open Source Software played a major role in the growth of the Internet and cloud computing, sparking issues about openness and freedom, the Free Software movement has the potential to provide a similar yet different impact on mobile cloud computing.

2. Mobile cloud computing is a need – form factor and other needs

By definition, mobile devices that access the Internet are performing mobile cloud computing: handsets need to borrow storage and computing power from the cloud because of their limited resources or because it makes more sense. For example, consider modern wireless car navigators, like the Dash: these devices not only can store locally the maps and calculate routes, but they rely on the cloud to get real time information about traffic conditions and plan the routes accordingly. Accessing data in the cloud from mobile devices is becoming a basic need.

3. Mobile cloud needs interoperability

Mobile cloud services are largely dominated by vendor specific walled gardens, and debate is not as intense as one would expect given the numbers of cell phone users. Probably this is due to the fact that not only Free Software powered mobile phones are still a minority, but also installing new software on phones was not an option for the mass market until recently. After iPhone and Android, with more and more 'application stores' emerging, the issue of mobile users' freedom is showing up. Users of one handset, for example, may want to get their email from a provider but sync pictures with another. Or if they buy music from a digital store from the desktop computer, they want to sync their playlists with any phone.

A minimum requirement is interoperable services² implementing open standards, because users' data must be preserved at all costs. Proprietary walled gardens create small monopolies that sometimes grow big and take away personal data from the users. Consider these recent cases that demonstrate that users of mobile cloud services are exposed to serious problems.

Palm Pre owners cannot access the music that they bought and stored in Apple's iTunes: Apple still wants to own the music it sold its users and keep their data hostage. Similar risks are run by owners of Amazon Kindle, who had their purchased books deleted too easily by Amazon from the devices.

The recent fiasco with Microsoft losing people's data is the opening act of how we've all learned that data is not necessarily safe in a proprietary cloud. If one of the world's premier software companies cannot be trusted to keep people's data safe, who can be trusted? Furthermore, do people really want Microsoft, Google et al to access all their data? With the cloud in general and in particular, the mobile cloud (because you want your mobile data backed up), it is more important than ever that people have the full ability to access and preserve their data, which means the open mobile cloud. These are just visible signals of proprietary services battling to own user data. If iTunes and Microsoft used interoperable and open standards, which could be safely implemented in free/libre open source software, their users would not face these problems.

2 <http://maffulli.net/2009/05/19/locked-devices-gplv3-and-the-path-to-mobile-freedom/>

4. Mobile cloud largely depends on locked-down devices

Network operators don't want users to be too free, so most of them prevent users to run applications that are not digitally signed. RIM, Apple and to some extent, Symbian devices are locked down, which renders users' freedom in the mobile cloud a balancing act: on one hand a developer needs to obey the rules dictated by network operators and device manufacturers; on the other hand the same developer needs to find ways to deliver freedom to users. In Funambol's case, for example, the official iPhone client can only sync contacts because the official Apple SDK only allows that. Nonetheless, the Funambol client for iPhone can also sync calendars by accessing directly the sqlite database, but this version cannot run on the device unless it is unlocked (breaking Apple's warranty).

5. Mobile cloud is an opportunity for free software providers

With so many new mobile devices hitting the market, billions of new users have the issue of freedom for the software on the device and freedom in the mobile cloud.

The Free Software community has the opportunity to participate in the mobile cloud debate and shape this new environment. Ignoring the issues posed by the mobile cloud risks excluding a large number of digital citizens from the benefits that free software has brought to other computer users. The mobile cloud is an open territory where many vendors³ are already fighting to lock-in users.

Resting on the cloud and network services, free and open source software should rely on licenses that prevent abuse. Fortunately, the Free Software Foundation has contributed a very good tool to bring freedom to the cloud. By extending its reach to interaction over a network, the Affero GPL v3⁴ (AGPLv3) is very effective at bringing copyleft to the services offered by cloud computing. Some people have had the chance to use open source software to offer services to the public, without returning anything to the community. That's taking open source software as free beer. It is just not being honest with the community, to the people who sweat to write the code to see someone running away with it and not contributing anything. Using the AGPLv3 for all software that can be used over a network is a smart way to start building a mobile cloud that respects user's freedom.

About the author

Fabrizio Capobianco is an Italian entrepreneur now based in the Silicon Valley where he has founded Funambol Inc, the company behind the largest open source project in wireless, with over a million downloads, providing over the air data synchronization to a large number of mobile devices.

3 <http://blog.internetnews.com/mmegna/2009/06/how-do-mobile-cloud-sync-servi.html>

4 <http://www.fsf.org/licensing/licenses/agpl-3.0.html>

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Capobianco, Fabrizio (2009) 'Five Reasons To Care About Mobile Cloud Computing', *IFOSS L. Rev.*, 1(2), pp 139 - 142
DOI: [10.5033/ifosslr.v1i2.24](https://doi.org/10.5033/ifosslr.v1i2.24)

Copyright © 2009 Fabrizio Capobianco.

This article was first published in the International In-house Counsel Journal.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Open Source Policies and Processes For In-Bound Software

Karen F. Copenhaver^a

(a) Choate, Hall & Stewart LLP

DOI: [10.5033/ifosslr.v1i2.27](https://doi.org/10.5033/ifosslr.v1i2.27)

Abstract

Virtually all companies use some amount of software that has been made available to the company under an open source license. The re-use of these valuable software components is inevitable and should not be discouraged. In addition to reducing the time and cost of development, re-use of components that have withstood the tests of time and the many eyeballs of critical peer review provides assurance of quality and reliability. As companies begin to recognize the enormous value of this software, they come to the realization that they must implement policies and systematic processes that assure compliance with the terms of open source licenses. This includes both managing the use of open source software within the company and external obligations arising under relationships with open source communities.¹ This article focuses on policies relating to in-bound code.

Keywords

Law; Free and Open Source Software; Governance; Compliance

Info

This item is part of the [Platform](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

The most successful compliance programs are not replicated from a form or copied from another company. They are organically developed within an organization to fit within that company's existing internal control mechanisms. The exercise of developing the policy and processes is an important part of the organization's preparation for adoption and deployment. As a threshold matter, companies must understand the critical distinction between policy and process. The term "policy" refers to a set of company values that should not change over time. Such values are aspirational in nature and need to be supported by processes that implement the core values. Processes do, and should, change as frequently as necessary to reflect the development and growth of the organization's business.

Participation in the development of the policy and process assures personal and organizational buy-in and results in an efficient process that does not die of its own weight. It makes it possible to optimize the process before it is rolled out, rather than relying on iterative fixes following serial failures that may result in improvements but at the same time undermine confidence and

¹ Such relationships typically include reciprocal contributions of software under an open source license, employee participation in open source projects, and commitments to open source communities.

commitment.

Open Source Policies

I divide compliance programs into two parts – a policy statement and a written business process to manage the company’s compliance with the policy. This section discusses the design of company policies; their implementation through company processes is discussed in the following section. Think of your “policy” as the part of your company’s written vision and value statement that will absolutely not change with time. Separating the changing from the unchanging gives more power to the permanent commitment and avoids the appearance that compliance is something akin to situational ethics. Here is an example of a policy statement for a software vendor that could be issued and remain unchanged over a very long period:

XYZ respects the intellectual property of others and expects others to respect its intellectual property. All company personnel must operate within established procurement processes for the acquisition of intellectual property assets, including software [and trademarks], for use in XYZ’s products or internal operations.

XYZ’s proprietary software and intellectual property are key assets that contribute unique value to the company. The impact of the introduction of any code licensed from external sources into XYZ’s propriety software must be fully considered before its incorporation into the development process and, if incorporated, compliance with the terms of the applicable license must be achieved. It is required that XYZ maintain full accounting for all licensed materials that are included in products commercially distributed and sold by the company.

This policy applies to all licensed materials, regardless of the method of procurement. Software that is available for download requires the same level of review and consideration as software acquired from a commercial company pursuant to a formal contracting process.

Any questions or concerns regarding compliance should immediately be addressed to _____ [Title such as the VP of Engineering].

The rest – anything that is fluid – is all process. Why the distinction?

- Because companies should not violate their policies. Company policies are commonly intended to be immutable concepts that provide fundamental guidance in the form of “thou shalt nots.”
- A company undermines the effect of all of its policies if any policy is not consistently applied.
- If you are in litigation, you do not want the other side to demonstrate that your actions were in violation of your own policy. You don’t want to be “hoisted with your own petard.”
- An overly dogmatic policy may also be poorly received by developers.
- Because a change in a policy implies that the policy was wrong to begin with and because most companies’ use of open source will change over time, the policy should anticipate a changing set of internal and external norms and requirements.
- A “one-off” decision should not be a violation of company policy; it should be an anomaly that is specifically supported by the process as implemented.

A particularly illustrative example of a statement that should *not* be included in any policy is something like: “It is company policy not to use any code made available under the GPL.” Here, the point is not whether the GPL is good or bad. Indeed, the company may, for whatever reason, prefer to choose other alternatives when all else is equal, but that does not amount to a flat-out prohibition against using GPL code. And even putting aside the thorny issue of using GPL code directly in the company’s proprietary code base, there are too many invaluable tools and programs made available under the GPL that can be used to facilitate internal development appropriately and in full compliance with the applicable license obligations to impose a blanket prohibition without an exception process. Almost all companies that develop software use tools, such as GCC, the GNU Compiler Collection, that are made available under the GPL, and most companies use GPL licensed code such as the Linux operating system in their internal operations. Issuing a policy that prohibits all use of code licensed under the GPL is highly likely to be violated on the date that it is issued. Thus an issue has been created because the company has a policy violation even if the company is in full compliance with the license. Rather, the important point is that policies adopted without thoughtful consideration of the issue will fail almost immediately.

Essential Elements of Successful Open Source Processes

1. Assignment of Responsibility for Decision Making

Probably the most important indicator of whether a process carrying out a policy will be successfully implemented in a company is whether there is a clear statement of personal responsibility for every part of the process. While you are reading below the details of structures and decisions, I am sure you will be thinking: “Why would you spend so much time dividing the analysis into all of these different parts?” Your own analysis may differ, but, in my experience, the answer to why a process is not working is somewhere in these structural issues – even though the structural issues are often only reflected in personal conflicts.

Here is an example of a problem that, in reality, is caused by a lack of structural clarity but often manifests itself in the context of a personality clash. Let us say an engineer named John is asked to gather information about an open source project and its potential use and to analyze its pros and cons. John works with development management, and a careful decision is made in favor of using the project. But that decision gets rehashed and remade all the way up the chain by people who do not know or understand the technical analysis that went into the original decision or who incorrectly feel it is their responsibility to make an in-depth analysis. John feels angry and dismissed, and complains to everyone who will listen that he will never take the process seriously again.

If we assume that, in the end, it was the right decision not to use the project, how would you fix the problem? I suggest that the best solution is to make it clear that everyone in the chain is not actually rehashing the same decision, even if that appears to be the case. Rather, what is happening is that everyone is making *different* decisions based on their own knowledge and expertise. When the responsibility for making different parts of the decision is not clearly divided and assigned, it is likely that everyone in the chain will re-visit all of the different parts of the analysis and will be unnecessarily risk-adverse. Frequently, everyone in the chain is authorized to say “no” and no one is comfortable saying “yes.” Therefore, identifying everyone’s specific role in the process can be essential to empowering everyone involved in that process to make an affirmative decision. Furthermore, doing so makes it clear that everyone’s contribution to the process is respected. For example, the lawyer should not be re-making technical decisions about component selection if the development manager has been clearly assigned responsibility for that task. Likewise, the development manager should not be making a decision based on his

assessment of the likelihood of a patent claim if that role is reserved to individuals tasked with corporate-wide risk management relating to patent matters.

There are many potentially successful structures, but decisions are commonly made either through a vertical or a horizontal process. A vertical process starts with a decision and recommendation made at the operational department level and specifies an orderly sequence through a series of chairs to confirm or overturn that decision. Each step in the sequence involves a single individual; there are no groups or boards that meet to consider the question. The last person in the sequence verifies that all the others have signed off and gives the final approval. The original request by the department is either confirmed or denied. No one in the chain exercises their own judgment regarding an alternative path. If the answer is no, the department goes back to work to develop another proposal and request. For example, the following is a common vertical process in a small organization:

- A non-management developer initiates a request to use an open source component. He or she has the responsibility to gather information about the open source project, the applicable license, etc. He or she provides that information to the development manager.
- The development manager assesses the usefulness and quality of the project, the time savings that would be achieved using that code, etc., and decides whether the choice is appropriate from a development perspective.
- The development manager consults with in-house or outside legal counsel who reviews the license, the proposed use and the business objectives, and determines that the license is appropriate for the use. The legal counsel's review is limited to matters within his or her expertise as a lawyer and confirmation that the company's internal processes have been followed.
- The development manager gets business unit management approval – probably without a meeting or presentation - based on confirmation that company policies and processes have been followed.

A vertical process is possible when issues are well-defined and when the impact of a decision is limited to a single business unit. The decision making and the process is largely in the control of the department making the request.

In contrast, a horizontal process provides for decisions to be made based on simultaneous input from multiple stakeholders. In a horizontal process the operational department requesting permission is not the decision maker. The impact of the decision goes beyond that department. Others are empowered to impose an alternative solution that is preferable for the organization. The answer to the original request may not be yes or no, it may be an entirely different plan of action. For example:

- The development manager gathers information and makes a business case to an Open Source Review Board on the pros and cons of using an open source component.
- All of the stakeholders across departments in the decision are represented on the Open Source Review Board, and the decision is made through the back-and-forth of committee deliberations. The committee reaches a consensus around the committee's recommended plan of action – which may be entirely different from the plan proposed by the department.

While both structures can work, processes that are not clearly defined are neither or both and the result is increased frustration at the requesting department level. The individuals in the requesting department have made a decision and know the facts regarding the request better than anyone else

in the organization. They want that decision confirmed unless there is a specific identified corporate policy that requires denial. They will assume the process is vertical unless expressly told otherwise. But an undefined process is likely to be conducted as though each rung in a vertical process is akin to an informal committee meeting resulting in a full review by participants acting outside of their expertise. A clearly defined horizontal process avoids this frustration by putting the department on notice that they are advocates for their request but not the decision makers. It also establishes a formal committee process with clearly defined areas of responsibility for everyone included in the meeting.

2. Who Initiates a Request and Who Gathers What Information?

Designers of effective processes need to clearly delineate the discrete tasks of gathering information and acting on that information. As discussed above, frequently the individuals tasked with collecting and presenting information believe—incorrectly—that they are responsible for making decisions based on their findings. If that is indeed the intended process, the process description should state so explicitly. Otherwise, those whose task is to gather and present information will provide only their conclusion and the information that supports that conclusion.

3. Who Makes What Decisions?

Generally, there are at least four components of any decision to use licensed code in your product.

- First, is it appropriate to have a dependency on code that is not owned by the company for the purposes of this product? The first decision is whether the company wants, or needs, to control the functionality to be provided by the code. Relevant considerations in that decision include: security, creation of a dependency on a format or standard, possibility of acquisition of the code owner by a competitor, difficulty or expense of removing the code after development on top of it has begun; and impact on product roadmaps or design decisions.

The tolerance for using licensed code has grown considerably over the years. In fact, an assumption has evolved that re-using assets is a wise business decision—and that assumption will only become stronger over time. But whether using open source code is a thoughtful decision or just a practical assumption, the first decision that the company is making is that using code that is not completely controlled by the company is appropriate in this instance.

This is the first step in the open source risk analysis. It is based on the company's lack of control or the absence of any confidentiality. It is not based on events that affect the project in general or any other users of a project. It is a determination of whether developing a dependency on this code creates a vulnerability for this particular company and product plan.

- Second, assuming that using code you do not own is appropriate, is this the right code? Does it provide the right functionality for today and is it extendable to provide the required functionality for tomorrow? Is it good code? Is it documented? Is support available, and, if so, from whom? Is there a compelling commercial reason for the support to be maintained? For example, the risk of using an open source project that is hosted by a commercial software company may be different from the risk of using code developed by an unincorporated project that is supported by many companies. The sole supporter may lose enthusiasm and there may be no other enthusiasts to pick up where the company left off. And if it is likely that support will be unavailable in the future, does it make business sense for the company to maintain the code in-house?

- Third, are the terms of the license for the code aligned with the company's business objectives? Can the company achieve its business objectives while complying with the license obligations? This is the lawyer's domain, where counsel can help the organization understand and prepare for the legal ramifications of business decisions. Here, most in-house lawyers have to conduct two lines of analysis: a legal interpretation of the license, and an analysis based on community consensus, or lack thereof, on the applicable license obligations.

Lawyers also have to determine (i) whether there is a process in place that will enable the company to stay in compliance in the future; (ii) the likelihood of inadvertent failure to comply with the license terms; and (iii) the impact of a compliance failure on achieving the company's objectives. If compliance is dependent on a specific set of facts, will a flag be raised if those facts change? For example, if compliance is dependent upon use of the code without modification, will a review be triggered if the code is modified? If achievement of business objectives, while remaining in compliance, is dependent upon limiting usage to internal application only, will a review be triggered if the code is distributed? This is the second step in the open source risk analysis, and involves internal risk that can be managed, rather than risk that arises from matters outside of the company's control. Companies with robust processes for managing compliance have more options for dealing with this exposure than companies that cannot be sure that implemented compliance policies will be maintained.

- Fourth, is the use of this code consistent with the company's tolerance for risk, its risk management practices, or both? This combined legal and business analysis is the third step in the risk analysis, based on matters outside of the company's control. This assessment is very similar for all code regardless of its origins. While the risk tolerance for open source in general has grown as open source has become more mainstream, risk analysis should always be project-specific. For example, the risk of using a project that is also used by many other industry leaders is very different from the risk of using a project that was long ago abandoned. Obviously, an established project is much more likely to have widespread support and resources if continued availability of the code and support is jeopardized.

This type of risk is not based on the applicable license, which means that companies that make decisions based entirely on the license will miss this part of the decision process. Who, if anyone, will assist in the defense of this code in the event of a patent or copyright infringement claim? What is the governance structure for the project? Has anyone reviewed the code for internal license conflicts? For example, since open source code does not come with any indemnification for intellectual property claims, is there some additional review of the code that should occur to make sure that the same diligence is applied to the open source code as is applied to internally developed or commercially licensed code provided without a viable indemnity?

4. Assignment of Responsibility for Information Gathering

This step is perhaps the most important element of identifying places where a process stalls or breaks down completely. It is very easy to create an appearance of progress by bouncing a request back and forth, but it is much harder to gather all the information necessary for the decision making described above. For instance, it is common for an original requester to complain about delays in making a decision, while the process is on hold awaiting information necessary to perform the analysis from a person tasked with obtaining the data. This problem is often compounded by a lack of feedback as to the effectiveness of information gathering—itself a sign of poor process.

Of course, the amount of information required for a decision varies significantly from organization to organization. Some companies get no more than the basic facts: project name; license; and some description about support options. Other companies want something approaching the Business Readiness Review (“BRR”) process originally proposed and developed by Carnegie Mellon West, O’Reilly CodeZoo, SpikeSource, and the Intel Corporation.² The BRR looks at the following characteristics in assessing a project’s maturity:

- **Functionality:** does the software meet user requirements?
- **Usability:** is the software intuitive, easy to install, easy to configure, and easy to maintain?
- **Quality:** is the software well designed, implemented, and tested?
- **Security:** how secure is the software?
- **Performance:** how does the software perform against standard benchmarks?
- **Scalability:** can the software cope with high-volume use?
- **Architecture:** is the software modular, portable, flexible, extensible, and open? Can it be integrated with other components?
- **Support:** how many sources of support are available?
- **Documentation:** is there good quality documentation?
- **Adoption:** has the software been adopted by the community, the market, and the industry?
- **Community:** is the community for the software active and lively?
- **Professionalism:** what level of professionalism does the development process and project organization exhibit?

Some companies do a full review of the source code before using it. Even though the project indicates that it is made available under a certain license, the file headers within the project may indicate other licenses that may or may not be compatible with the declared license for the project. Code scans can find other code that may have been copied from other projects that are made available under a license that is not indicated. Furthermore, some companies have very different review processes for code to be shipped in a product and code to be only used internally.

5. Assigning Responsibility for Follow-Through

Follow-through refers to an explicit mandate for ensuring ongoing compliance with the review process once initial approval has been granted. This is the most difficult step to implement in the entire process, as illustrated by the example of a company that had developed a very specific plan to ensure compliance with certain license obligations. The plan had been documented, and development had commenced. However, the details of the plan had not been communicated to the right people on the product management team and was ignored.

Best practices for implementing follow-through programs that ensure ongoing compliance with the process include tightly coupling review processes with product development cycles, usually in the form of automatic triggers when potential issues arise. It is important to avoid the all-too-common pitfall of follow-through plans: merely keeping a file in the lawyer’s office with information about a particular component decision without ensuring that this information is actually reviewed as the code travels throughout the development process.

6. Education

To be successful, a process should place a heavy emphasis on ongoing education.

- Education at the time the policy is rolled out. The tone of the communication is

2 See Home – Open BRR, <http://www.openbrr.org/> (last visited Jan. 22, 2010).

important. If it sounds as though the lawyers are “crying wolf,” the process will do more harm than good. Generally, the best communication will portray the controls being imposed as positive steps enabling more efficient use of open source, where consistent with the company’s goals. It is equally important to select the right people to deliver the communication. If a lawyer is sent to a department meeting to talk about the risks associated with using open source code, it is unlikely that anyone will long remember what was said. An enthusiastic and supportive department manager explaining the policy and process will find a much more receptive audience. And a dose of reality may help: the best education session I have attended was conducted by a manager who was new to the company and had just survived a remediation effort at his prior employer, which he described as the most tedious and frustrating experience of his working life. This personal account was a helpful illustration of what could happen if the introduction of code into the product was not actively managed.

- Education of all new employees. Some employees, especially those fresh from school, can sometimes develop bad habits that are very hard to break. For example, one company hired a brilliant programmer only to find out six months later that he had contributed massive amounts of code to the company’s code base—amounts that were not humanly possible to write in his short tenure at the company. The employee had dutifully gone through all of the new employee training and had signed a statement that he understood the policies. Then how could he have thought that cutting and pasting massive amounts of code from Internet sites was appropriate? Simply, it was the way he had learned to code. The idea that he should start every project with a fresh sheet of paper seemed so preposterous to him that he had assumed that the company did not mean it. What is more, new employees may bring with them incorrect assumptions about open source code from their previous employers, or even from what they read on the Internet.
- Ongoing training regarding process improvements. Hearing and responding to criticism and implementing suggestions is key to a successful program. It is important to let the employees know that someone is listening and responsible for the success of the process.

7. Open Source Review Boards

To capture the benefits of horizontal processes described above, many organizations establish a cross-discipline group of individuals, often called the Open Source Review Board, who meet and decide as a group on all open source usage. Usually, this group assumes responsibility for all aspects of the decision because all of the stakeholders in the decision-making process are represented on the board.

Many companies rotate executives, who are in the best position to share knowledge across the entire organization, on the open source review board to spread their support of and confidence in the process across the company. At the same time, many organizations prefer to keep one or more individuals on the Open Source Review Board for several years to ensure continuity of learning. It is often these board members who are best able to change established precedent, because they remember the basis for the original decision and know when it is no longer applicable or appropriate to the facts of the situation.

Depending on organizational preferences, the role of an Open Source Review Board can be limited to confirmation of decisions already vetted through a vertical process. Alternatively, the company can adopt a largely horizontal process, with the board acting as a court of first impression. Either choice can be efficient, depending on the commitment of the members of the board to engage in the process and to make attendance at the meetings a high priority. If the board members do not make the board a high priority, then the role of the review board should be limited to confirming the results of a prior vertical process.

In my experience, Open Source Review Boards work well in larger organizations but not as well in small companies. In smaller companies the people on the board wear too many hats and expect others to attend the meeting when they are stretched thin. The in-house counsel frequently relies on the committee meeting for verification that the issues have been fully vetted and the counsel's work is delayed when the board meeting has to be rescheduled because the required decision makers are not in attendance. And precisely because a process already exists – to convene the committee – there is no alternative process to support thoughtful analysis within the management chain and outside of the review board.

8. Open Source Compliance Officers

Because the establishment of an Open Source Software Compliance Officer (“OSSCO”) has been a component of many well-publicized settlements of litigation based on allegations of non-compliance with open source licenses,³ many companies are proactively considering the establishment of a position with a similar title.

Beyond the title, there is a lot of variation in what the job description or mission statement for an OSSCO looks like. In a large organization, the job is probably more akin to an open source ombudsman who maintains some degree of a separation between the day-to-day business processes for open source approval, and is available to discuss concerns from individual employees concerned about the company's fulfillment of commitments to the communities from which the company benefits. A single compliance officer could not personally be involved in every decision, so the focus is at the process level and on specific issues that arise out of the ordinary course of business.

In smaller organizations, the description of the OSSCO's duties is closer to that of a one-person Open Source Review Board. The officer is involved in every decision that the company makes regarding development, use or distribution of open source software. To the extent the officer serves as a champion of open source software within the company, his job description should ensure that his recommendations are attuned to legitimate business needs of the company. Here is a proposed job description that was provided by Karen Sandler of the Software Freedom Law Center:

The OSSCO should be available and responsive regarding issues relating to free software license compliance. The OSSCO should undertake best efforts to resolve all such issues as quickly as possible. In cases where violations have been identified, the OSSCO should on a periodic basis provide to the copyright holders a written report of the scope and manner in which the company is redistributing the software and complying with the applicable licenses. The OSSCO should also be responsible for reviewing all of the company's products before they are offered to the public to ensure that they are in compliance with all applicable free software licenses.⁴

9. Timing and Tools

Most processes that are perceived by developers as successful have some element of guaranteed turn-around time. This does not mean that there is a promise that all issues will be resolved in a given time period, as that only guarantees a negative answer if the assessment has not been

3 See, e.g., BusyBox Developers Agree To End GPL Lawsuit Against Verizon, Mar. 17, 2008, <http://www.softwarefreedom.org/news/2008/mar/17/busybox-verizon/> (discussing the terms of the settlement between Verizon Communications Inc. and the Software Freedom Law Center acting on behalf of two BusyBox developers).

4 Private correspondence between Karen Sandler and Karen F. Copenhaver. For another excellent discussion of OSSCO, see also Posting of Stormy Peters to <https://fossbazaar.org/?q=content/job-description-open-source-compliance-officer> (Mar. 12, 2008) (offering thoughts on an OSSCO job description).

completed by the deadline. But a promise that there will be a response within a specified period provides confidence that open source issues will not hold up development. With a dedicated group sitting on an Open Source Review Board, institutional learning grows, and issues quickly begin to sort themselves into those that are relatively easy to answer and those that will take more time and work to bring to a conclusion. Timing commitments also tend to encourage automation of the process and the creation of forms and guidance documents that make for more efficient operation. Below are several thoughts on recommended best practices.

- Request forms and templates that make sure that all required information is provided with the initial submission avoid an inefficient back-and-forth information gathering process.
- Guidance documents that provide insight into a typical Open Source Review Board's analysis filter out requests that are unlikely to be approved.
- Email approval processes that make sure that the requests are circulated to the right people save time and frustration. Consider using dedicated mail accounts and distribution lists.
- Process management tools that provide status reports upon request and automatically remind decision-makers of approaching deadlines make the process more efficient.

To belabor an obvious point, a process for managing the use of open source software will benefit greatly from automation. Automated tools can:

- provide evidence of unintended open source usage (which often is the very first step in convincing management of the need for a formal process);
- deliver timely reminders or decision triggers by identifying open source components as they are added to a source tree;
- gather and organize the information necessary for decision making;
- provide a record of analysis and decision making;
- maintain a bill of materials for any code base that travels with the code or is available as part of the product checkpoint process; and
- assist in identifying all possible sources and available licenses of discovered open source code.

What Is the Right Policy and Process for a Company?

Of course, the right policy and process for a particular company will depend on many factors: the reason for implementing the process (e.g., what is the immediate issue?), the level of sophistication of the company's employees about open source software and communities, and the existence of a compliance issue that has already involved third parties in the process.

The main reason for implementing an open source review process is compliance. But the real answer is more nuanced, and requires understanding of events that triggered the introduction of a policy. Some examples follow.

- Customer demand. The number one reason for implementing a compliance program is a request from customers. For example, a software company has very little experience with open source but responds to customer demands for a version of their product that works on open source platforms, or to demands for a list of all open source code used in the product. The company intends to hire developers who are familiar with those platforms, and wants to educate both their existing and new developers on a consistent approach to controlling the introduction of open source code into their development environment.

- Reverse procurement policy. A company that had a prohibition on using open source software has decided to change that policy. To address the issue of gaining control of the process in its early stages, the company's policy addressed the fact that the company's employees had little or no experience with open source.
- Push the work off the lawyer's plate. The company's lawyer is getting requests for approval of licenses, and the requests arrive without any of the information required to make a decision. Frustration on both sides necessitates the creation of a process for using the attorney's time wisely and obtaining a commitment from the attorney for a target turn-around-time. Because this situation demands difficult compromises from both sides, it is important to employ well-respected internal champions who can move the process forward.
- The painful moment. A company suffers a rude awakening when it discovers a compliance failure, perhaps in the form of notices from third parties or even undesirable media attention.⁵ The resulting remediation efforts are disruptive, and the company executives want to make sure that they are never in this position again. While the nature of the policies that arise in these circumstances is usually relatively ponderous, nonetheless the policies tend to work well *as designed* because they have executive backing at the highest level and are implemented quickly.
- Anticipating a merger or acquisition. A company anticipates an acquisition as its exit strategy and it wants to be prepared for due diligence inquiries from the acquiror. Here, the anticipated acquiror's counsel serves as the "bad guy" to whom all internal frustration can be transferred. Under these circumstances, developing an open source policy and process can be complemented by reviewing the company's code base for existing open source usage.
- Closing condition for round of financing. After a round of due diligence in relation to a financing, developers are instructed by a company's board to "get the code clean and to keep it clean." This incentive coming from the highest levels is likely to assure success of the process *regardless* of the nature of the policy adopted.
- Consistency across groups. A company has good procedures for most of its business operations but little or no procedures for certain part of its business (e.g., a recently-acquired small company whose core product contains significant open source code). While developing a global approach that works for all groups can be difficult, the existing disparity between divisions creates a feeling that the failure to adopt similar policies in the non-compliant segment of the business is intentional or willful. Moreover, this failure in the non-compliant portion destroys the value of the significant investment that has been made to bring the rest of the company's operations to compliance.
- Open sourcing own code. A company decides that it wants to contribute a code base to an open source project and to try to form a community around that code base. This investment in the release of the code and in the development of the community around it will be severely undermined if the company fails to comply with its open source obligations to other communities.

Conclusion

Companies should establish their core values with respect to open source usage, and formalize their analysis in a nuanced policy that responds to current issues and anticipates future challenges. Once a policy is in place, it should be operationalized by a well designed and articulated formal process. Regardless of the motivation for implementing the compliance program, an efficient

⁵ See, e.g., GPL Violations homepage – The gpl-violations.org project, <http://gpl-violations.org/> (last visited Jan. 22, 2010) (listing past and present infringers of the GPL).

process with clearly identified responsibilities is important to gain the necessary support of all of the stakeholders within the organization.

About the author

Karen F. Copenhaver is listed in *The International Who's Who of Internet & e-Commerce Lawyers*, *Chambers USA*, *Best Lawyers in America* and as a *Massachusetts Super Lawyer*. She was also named in *The Legal 500* for technology transactions. Ms. Copenhaver is only the 5th lawyer ever to receive Mass High Tech's prestigious "Mass High Tech All-Stars Award," which honors the thought leaders and innovators throughout the New England technology sector. She has been chosen by *Intellectual Asset Management* magazine as one of the world's top IP strategists in their feature "IAM 250 — A Guide to the World's Leading IP Strategists." Ms. Copenhaver is also director of intellectual property strategy for the Linux Foundation.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 1, Issue 2 (December 2009). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Copenhaver, Karen F. (2009) 'Open Source Policies and Processes For In-Bound Software', *IFOSS L. Rev.*, 1(2), pp 143 – 154

DOI: [10.5033/ifosslr.v1i2.27](https://doi.org/10.5033/ifosslr.v1i2.27)

Copyright © 2009 Karen F. Copenhaver.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.





International Free and Open Source Software Law Review

CALL FOR PAPERS

The 'International Free and Open Source Software Law Review' (IFOSS L. Rev. or IFOSSLR) is a peer reviewed biannual journal for high-level analysis and debate about Free and Open Source Software legal issues. It is published by an independent Editorial Committee and seeks submissions from qualified authors in a variety of research areas, blending the strengths of legal rigour and outstanding technical and community awareness.

The topics covered by the publication include copyright, licence implementation, licence interpretation, patents applicable to software and business methods, standards applicable to software, case law, statutory changes, license enforcement, competition law applicable to software, economics of Free and Open Source Software, Free and Open Source Software business models and due diligence of Free and Open Source Software.

IFOSS L. Rev. is the first legal publication to focus specifically on issues facing Free and Open Source Software, a highly successful community-driven development paradigm that rewards cooperative production.

Full-length articles

IFOSS L. Rev. accepts full-length researched articles on legal issues of interest to the publication's readership. All accepted submissions are peer-reviewed. No fixed length is prescribed, but 2500 – 5000 words is a useful indicative range.

Legislative Review

This section contains briefs on recent or upcoming changes to legislation relevant to Free and Open Source Software. Articles discuss the main implications of the legislative change, set out the context of prior law, and indicate how governmental policy and/or international law will affect implementation. All accepted submissions are peer-reviewed. No fixed length is prescribed, but 1500 – 3000 words is a useful indicative range.

Case Law Reports

The focus of this section is reports on recent developments in case law that offer commentary on the legal and policy implications of the decisions. Articles are peer-reviewed where appropriate. No fixed length is prescribed, but 500 – 3000 words is a useful indicative range.

Book Reviews

IFOSS L. Rev. accepts reviews of recent literature relevant to Free and Open Source Software. Book reviews are normally 500 – 2000 words in length, and should critically analyse the main arguments of the literature under examination. Further, they should indicate the reviewer's assessment of the literature's usefulness to IFOSS L. Rev.'s readership, its novelty of approach, its factual accuracy and its relevance to current debates.

Tech Watch

IFOSS L. Rev. prides itself on outstanding industry awareness and cross-disciplinary reach. Tech Watch is therefore offered as a forum to set contemporary technical and social issues in the Free and Open Source community into their legal context. These discussions may form the basis of subsequent detailed examination by other contributors. Tech Watch columns take a in short essay form, and are expected to be between 500 and 2000 words in length. They are usually written by non-lawyers.

Submissions process

Submissions are now being sought for publication in 2010. Articles can be submitted to IFOSS L. Rev. at: <http://www.ifoossr.org>

Submissions should adhere to the publication's editorial and style guidelines listed at: <http://www.ifoossr.org/index.php/ifoossr/about/submissions#authorGuidelines>

Any administrative and bibliographic enquiries can be directed to admin@ifoossr.org

"IFOSSLR may be hard to pronounce, but it is an easy to read, informative new publication. I look forward to future issues."

-- William Patry, Senior Copyright Counsel, Google Inc.

"Exchange of ideas is the life principle of free software. This journal is a crucial part of the exchange of ideas making free software law."

-- Eben Moglen, Software Freedom Law Centre

Thanks to our Sponsors

IFOSS L. Rev. is published by its Editorial Committee. The Committee gratefully acknowledges the sponsorship of its many supporters and donors, including:

