The Open Source Legal Landscape

Brendan Scott[*]
Open Source Law
www.opensourcelaw.biz
inquiries@opensourcelaw.biz

March 2006

1. First Principles – Motivations of Copyright Law

1.1 Copyright is a legislative monopoly, created (in Australia) by the *Copyright Act 1968 (Cth)*. The purpose of the *Copyright Act* is to provide an incentive for the creation and distribution of works[1] the subject of copyright. It does this by granting monopoly rights to perform certain acts (such as reproduction), thereby permitting the vendor of a work to sell that work above the marginal cost of production of that work.

1.2 As such, copyright is a form of industry development scheme or Government subsidy aimed at promoting a particular end (ie the production and distribution of works). In 1984 this subsidy, which traditionally covered books and music, was extended to also cover software. Prior to 1984 it was legal to make electronic reproductions of software.[2]

1.3 Copyright operates by prohibiting numerous categories of action in respect of a copyright work. These prohibitions are, subject to some exceptions, absolute prohibitions and are independent of the means by which a person acquired possession of a work in respect of which the actions are carried out. In the absence of a permission from the copyright holder, there is an absolute prohibition on exercising any of the rights comprised in copyright in respect of that work.

1.4 For example, if a person buys both a screwdriver and some software from someone, they may do what they like with the screwdriver, including modifying it, improving it, renting it to others etc. However, the scope of actions that they can undertake with the software is strictly limited (subject to some qualifications[3]) to those things over which they have been granted permission from (ie licensed by) the holder of copyright in the software. This typically does not extend to modification, improvements, renting etc. Often this permission (ie the licence terms) only applies to the execution of the software in restrictive circumstances (such as tied to particular hardware, or particular classes of hardware, a limit on concurrent users etc).

[*] Brendan is the principal of Open Source Law, an ICT legal practice with a special focus on open source and customer copyright. Brendan has over 12 years of experience in ICT related legal issues. He is a director and founding member of of Open Source Industry Australia Limited and is on the steering committee of the Australian Service for Knowledge of Open Source Software, a national focal point for advice, management, governance, storage and dissemination of Open Source Software (OSS) for research and higher education.

[1] The *Copyright Act* uses concepts of "works" and "subject matter other than works". The term "work" is used in this document in its generic sense, not its Copyright Act sense . That is, "work" means some form of content to which copyright attaches, including subject matter other than works.

[2] *Computer Edge Pty Ltd v. Apple Computer Inc*. (1986) 161 CLR 171.

[3] "Strictly limited" overstates the position slightly. There are activities in relation to a work which are not prohibited by copyright, some prohibitions apply in a different fashion to different categories of work, and some activities may be rendered non-infringing in specific circumstances. This statement should be read subject to these qualifications.

1.5 The consequences of copyright infringement are broad and include a requirement to pay damages or an account of profits, and there may be criminal penalties attached to an infringement. An infringement can also result in the seizure of equipment used in the course of carrying out the infringement.

2. First Principles – Comments on Contract v Licence

2.1 There is a great deal of confusion in the area of open source licensing arising from a fundamental misunderstanding of the different roles of contracts and licences. It is all too common to see these concepts identified in the context of open source. A contract is "a legally binding promise or agreement,"[4] while a licence is a permission to do something. In the context of software licensing, a licence is a permission to do something which would otherwise be prohibited by copyright law (such as the reproduction of the software). A contract may include a licence, and a licence may form part of a contract. However, one does not imply the other.

2.2 For example, it is a trespass to enter onto someone's land without their permission. That said, people enter onto other's land all the time without those entries being trespasses – when you receive visitors, or when you walk into a shop are examples. This is because the owner or lessee of the land has either expressly (such as inviting someone) or by implication (shopkeepers) permitted them to come onto the land. If there was a big sign out the front of land[5] which states that anyone wearing a red hat can enter onto the land then, as a consequence, anyone wearing a red hat could enter the land without committing a trespass. If they do not comply with the conditions of the licence (by not wearing a hat, or wearing a blue hat), then they will commit a trespass if they enter the land because they have not met the preconditions to be entitled to the permission. In none of these circumstances is there a need for the exchange of promises (which traditionally characterises a contract) to occur with the entrant.

2.3 The same principles underly open source licensing. As mentioned above, the *Copyright Act 1968* (Cth) absolutely prohibits certain activities including the reproduction of software. Open source licences effectively exempt the permitted activities from copyright infringement, subject to compliance with certain conditions (which are different depending upon the licence). A failure to comply with the conditions in the licence will mean that the activities are no longer exempted from infringement of copyright. If the activity in question results in an infringement of copyright, then the copyright owner will have an action against the person engaging in the activity.

3. Old Model Licensing (Closed Source)

3.1 The old model of the exploitation of copyright for software involved granting licences in respect of the software in such a way as to prevent effective ownership of the software by buyers. The example given in paragraph 1.4 above demonstrates this in that, even though the purchaser buys both items, there are substantial restrictions on the use of the software, with no analogous restrictions on the use of the screwdriver. It is this lack of ownership rights under the old model which is used by vendors of works to secure an above market price.[6]

3.2 Historically the old model of copyright exploitation has worked to establish or entrench market power in those sectors of the economy where it operates. That entrenchment appears to occur through the control of distribution channels. Not only does this practice raise the price of software to the end user, it also creates substantial costs for new entrants seeking access to those channels, thereby reducing competition, with a compound effect on price. Further, the non-purchase costs of the old model (ie closed source) are comparatively high and introduce non-trivial transaction costs into every transaction in respect of the copyright work. These costs include the costs involved in finding and negotiating a licence and the costs of administering the licence to ensure that activities remain within the bounds prescribed by the licence terms.

3.3 These transaction costs have the effect of excluding from a work a spectrum of possible improvements (that is, those for which the benefits are less than the transaction costs involved in inclusion) and of

[4] Carter, JW, Harland, DJ, Contract Law in Australia, Third Edition, Butterworths, 1996 at [101].

[5] Technically, the sign would need to have been put there by someone who had the legal authority to authorise the entry.

[6] That is, a price above marginal cost of reproduction (in this case roughly $0). This is the price economics predicts for a competitive market -

preventing the maximal exploitation of the work (again, if the licence fees plus the transaction costs involved in acquiring and administering the exploitation of the work exceed the benefit from doing so, then the work will not be acquired).

4. Open Source as a New Model

4.1 Open source licensing is a customer driven market reaction to the high transaction costs and anti-competitive effects that the old model has produced. It effectively says that, through judicious use of copyright, customers can acquire software with rights analogous to ownership. In the example above, if the software is open source software, the person acquiring the software would have property-like rights over the use of the software in a manner analogous to the rights they have over the screwdriver.

4.2 The fundamental difference therefore between the old, closed source, model and the new, open source, model is that under a closed source licence, a customer acquires very restricted rights in relation to the software, whereas under an open source licence, a customer acquires very broad rights analogous to ownership of the copy they acquire.[7]

4.3 Another way of looking at this is that open source licensing attempts to treat software as a form of property, while the old model of licensing attempts to prevent such treatment. That is, open source is a form of deregulation of the software industry. Open source uses copyright to effect that deregulation.

5. Some Open Source Misconceptions

Open Source is Pro-Copyright

5.1 An open source licence is a licence over copyright granted by the copyright owner of a work which has certain characteristics (discussed further below). As a licence, it is only meaningful in the presence of the copyright regime. Open source licences are explicitly dependent upon the continued existence of copyright for their efficacy. As open source would not exist without copyright it is incorrect to assert that open source is opposed to copyright.

Complement of Commercial is Non Commercial, not Open Source

5.2 A corollary of section 4 above is that open source is a particular model for the commercialisation of software. It is a different model, but not a non commercial one. That said, there exists open source software which is made available on a non-commercial basis, just as there is closed source software which is made available on a non-commercial basis.

Complement of Open is Closed, not Proprietary

5.3 A corollary of paragraph 5.1 above is that the copyright in open source software is owned by someone, otherwise there is no basis on which a licence can be granted. As such to oppose the terms “proprietary” and “open source” software implies that the copyright in open source software is not owned by someone. This is incorrect. That said, this use of “proprietary software” is, unfortunately, widespread. If anything, the complement of proprietary software is public domain software. That is, software over which copyright does not exist or is not asserted.

6. Main Characteristics of Free Software and Open Source Definitions

6.1 Two important concepts in this area are the definition of open source software and the definition of “free software”. These concepts are related, but distinct.

6.2 Open source software is software which meets the requirements of the open source definition (sometimes referred to as the OSD). The open source definition is maintained by an organisation called the Open Source Initiative. The definition includes the following 10 elements:

(a) Free Redistribution – the code must be able to be redistributed without payment of a royalty;

(b) Source Code – the source code of the software must be made available;

(c) Derived Works – recipients must be permitted to modify the software and distribute modifications;

[7] However (consistently with the screwdriver example) this property only applies to the copy that the customer acquires. Where they provide a copy to another person, that person receives a like property in the copy they receive.

(d) Integrity of The Author's Source Code – licence may include some attribution restrictions;

(e) No Discrimination Against Persons or Groups;

(f) No Discrimination Against Fields of Endeavour;

(g) Distribution of Licence – licence applies automatically to recipients;

(h) Licence Must Not Be Specific to a Product;

(i) Licence Must Not Restrict Other Software;

(j) Licence Must Be Technology-Neutral.

6.3 The open source definition is available from: http://www.opensource.org/docs/definition.php. The definition includes more detail on each of these elements.

6.4 The free software definition is comprised of the following “four freedoms”:[8]

(a) The freedom to run the program, for any purpose (freedom 0);

(b) The freedom to study how the program works, and adapt it to your needs (freedom 1);

(c) The freedom to redistribute copies so you can help your neighbour (freedom 2); and

(d) The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3).

6.5 Where the licensing terms for software meet these conditions it is said to be “free software”. Note that the definition of free software does not require that the software actually be without cost. To repeat an oft quoted analogy, the “free” in free software is “free as in speech not free as in beer”. I would add here, “free as in market, not free as in beer”. The free software definition is maintained by the Free Software Foundation (www.fsf.org).

6.6 It must be kept in mind that these are technical definitions of the respective terms and it is not at all easy to deduce what the practical impacts of the application of these definitions from an abstract study of them.

7. Strong v Weak Licensing

7.1 One of the characteristics of open source licences is that they must “*permit*” the source code of modifications to the software to be licensed under an open source licence. The practical consequences of this permission are unclear. For example, a permission to license under a specific open source licence, is not a permission to license on any other basis. Some open source licences go further, not only permitting, but *requiring* that modifications of the software or other, related, software be licensed under an open source licence, typically under the terms of the same licence. Where a licence has this characteristic, and especially so where it attaches to other software, it is said to be a “strong” licence. That said, strong licensing should be considered as a spectrum, rather than as two discrete states. If the licence also requires that source code be distributed in conjunction with binary code, that will have practical consequences, thus increasing the strength of the licence. If the licence permits licensing under a different licence, that licence must also have an open source licensing requirement for the first licence to be considered a strong licence. The strong licensing requirement may be subject to a trigger condition (such as distribution of the modification). A licence which does not have these characteristics is known as a “weak” licence.

7.2 The GNU General Public License (GPL) is the best known, and perhaps the most widely implemented of the strong licences. It requires that if modifications to GPLed software are published or distributed, they must be licensed under the terms of the GPL. It further requires that works which: (a) include as part of them a modification of software licensed under the GPL; and (b) which are distributed must also be licensed under the GPL and that access to the source code of the software must also be provided. In this way the GPL may also attach to the distribution of the whole of a work which includes GPLed code rather (rather than being restricted to the modification per se). The GPL is the licence which

[8] http://www.fsf.org/philosophy/free-sw.html

covers the Linux kernel. The Mozilla Public License (MPL) is another example of a strong licence. At least one version of the MPL (version 1.1) permits the nomination of an alternative licence for modifications. Where the GPL is nominated the MPL will be an example of a strong licence which permits licensing under different licensing conditions.

8. The best known example of a weak licence is the Berkely Software Distribution License (BSD). Common wisdom is that the BSD licence permits the software to be modified and for those modifications to be licensed on a different basis, including under a closed source licence. One of the requirements of the BSD licence ("*Redistributions ... must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution*") can be read as requiring the BSD to attach to modifications. However, the absence of an express requirement in the BSD licence to distribute the source code of software in conjunction with a binary substantially weakens the licence. The BSD licence underlies the BSD family of operating systems (eg FreeBSD, NetBSD, OpenBSD).

8.1 The text of each of the licences mentioned above is available from:

http://www.opensource.org/licenses/

9. Licence Compatibility

9.1 One of the consequences of strong licensing is that care must be exercised when combining source code from two or more different projects which are licensed under different licences. To take the examples above, if two software products, one of which is licensed under the MPL, the other under the GPL are combined the resulting product must be licensed consistently with the requirements of both the MPL and the GPL. If the requirements of these licences are *per se* inconsistent then there is no legal basis on which the output product can be licensed. Compatibility between two licences refers to whether or not software licensed under those licences may be combined to produce a work which can be licensed consistently with the requirements of both licences. Typically it is important to know whether a licence is "GPL compatible" so that software the subject of it can be combined with software the subject of the GPL.

9.2 The Free Software Foundation has a page setting out the compatibility of various licences with the GPL here: http://www.gnu.org/licenses/license-list.html.

10. Dual Licensing

10.1 The fact that a person has granted a permission in a specific form does not mean that that person cannot grant further permissions in other forms. For example, saying someone can come into your yard and use your pool on Mondays and Wednesdays if they bring a towel doesn't prevent you from also permitting them to do so on Fridays and Sundays if they bring a beach ball.

10.2 The same issues underlie the concept of dual licensing. Under a dual licence, a software vendor grants a licence over software on the terms of an open source licence. They will also make it known that if someone wants to acquire the software under different terms, then they are prepared to allow that as well. Typically, the acquirer of the software will want to avoid some aspect of the open source licence and will pay a premium to do so.

10.3 The common candidate for a dual licensing is the GPL (see discussion in section 7). As discussed above, the GPL is a strong licence because it requires that modifications to the software which are published or distributed must also be licensed under the terms of the GPL. So a person might not want to use some code the subject of the GPL to incorporate into their own software because then that combination would need to be licensed under the GPL. By offering the code under a dual licence scheme the dual licence vendor may charge other vendors to allow them to include the software in their own products under a licence other than the GPL. Dual licensing is therefore unlikely to work effectively with a weak licence.

Examples of Dual Licensing

10.4 The database software vendor MySQL (http://www.mysql.com/) uses a dual licence model for the distribution of its namesake database product MySQL. Most of MySQL was licensed under the GPL, although some parts were licensed under a licence called the Lesser GPL (LGPL). However, a few years ago they chose to use the GPL for those components as well. MySQL claims to have one of the largest installed bases for database software in the world. For more details about MySQL's licensing see:

http://www.mysql.com/company/legal/licensing/

10.5 The open office suite OpenOffice.org also used a dual licence structure for the version 1 code tree, with the software licensed under the GNU Lesser GPL (LGPL) and the Sun Industry Standards Source Licence (SISSL). For the version 2 code tree, only the LGPL has been used. See:

http://www.openoffice.org/FAQs/faq-licensing.html#1

11. How Risks Relating to Open Source Can be Categorised

11.1 Many open source risks are very similar, or identical, to risks experienced in relation to closed source, although the greater freedom afforded by open source means that an organisation may play a different role in relation to the software and therefore is exposed to different categories of risk. For example, an acquirer of software under a closed source licence is usually prohibited from distributing that software so risks associated with distribution will not arise in practice. However, under an open source licence that acquirer will be free to distribute that software if they choose to and, as a result, may be exposed to risks related to that distribution.

11.2 Open Source risks can be broadly separated into three categories:

(a) acquisition risks;

(b) supply risks; and

(c) interlinked risks.

11.3 As one might expect, acquisition and supply risks have obvious meanings. Acquisition risks relate to the risks which are involved in the acquisition of software from a third party. Supply risks relate to the risks involved in the supply of software, whether your own or someone else's, to a third party. Interlinked risks are risks which involve an element of both of these aspects – for example, the acquisition of some software for the purpose of modification and on supply of the software as modified.

12. Overview of How Risks can be Addressed

12.1 Open source risks are addressed through proactive action and risk management plans. Those plans involve steps such as: the identification and classification of the risk, an assessment of the likelihood of the risk occurring and of the consequences if it does occur and taking proactive steps to prevent or address the occurrence of the risk.

12.2 Proactive measures include the adoption of policies to be followed when acquiring open source software, including the conduct of a due diligence check in much the same way as a credit check would be carried out on a closed source vendor. Where the open source software is acquired under a contract, they can also include contractual provisions which address specific aspects of the risks.

12.3 In addition, establishing a process to identify and restrict the likely uses of open source software also assists in addressing risks. For example, by restricting activities in relation to open source software to acquisition and use automatically excludes supply and interlinked related risks. The policy may require special permission to be sought if the relevant employee proposes to have the organisation engage in (eg) a supply of open source software. A policy position might include different due diligence processes to be followed depending on whether the proposed activities included acquisition, supply or interlinked aspects respectively.

12.4 The effects of open source licences can occur immediately on the occurrence of a trigger condition (such as, the distribution of a modification to the software). The most important aspect of risk management in this area is therefore a proactive approach to open source acquisition and use.

13. Interlude - Wrapping Things Up

13.1 There are several dozen open source licences approved by the Open Source Initiative. Of these, one, the GNU GPL, accounts for roughly two thirds of all projects on the Sourceforge software repository. We conclude this paper with an overview of two of the best known licences – the GNU GPL and the BSD licence and provide a number of short case studies of open source related matters that OSL has been involved with. These case studies give some insight into the breadth of issues which can arise and some of the means of addressing them.

14. GNU General Public License

14.1 The GNU General Public License (GPL) is the licence of choice of the GNU Project (www.gnu.org) and of the Free Software Foundation (www.fsf.org – note, this is the same site as www.gnu.org). This paper relates to version 2 of the licence, available from: http://www.fsf.org/licenses/gpl.html.

14.2 The GPL grants broad usage rights. Where a person is granted a GPL over software, they may copy and distribute the source code of the software verbatim (clause 1) subject to the following:

(a) they must include an appropriate copyright notice;

(b) keep intact all notices referring to the GPL and the absence of a warranty;

(c) give other recipients of the program a copy of the GPL;

14.3 The licensee may also copy and distribute the object code of the software provided that they comply with the qualifications set out in paragraph 14.2(a)-(c) above and simultaneously distribute or provide effective access to the source code (the licence sets out specific options in relation to the distribution of the source code) (clause 3).

14.4 The GPL also includes an express disclaimer of warranties and an exclusion of liability.

14.5 The GPL permits the making of changes to the software and does not require the distribution of changes made. However, if you do distribute those changes, and they are "derived from" the software, you must distribute those changes on the terms of the GPL. This makes the GPL a strong licence. It is not clear from the wording of the licence, but this licensing requirement is unlikely to apply to "internal" distributions within the organisation.

15. BSD Licence

15.1 The BSD (Berkeley Software Distribution) Licence was originally included with the distribution of the Berkeley Unix in the early 90s. There are two versions of the BSD licence, distinguished by "the advertising clause". The advertising clause was removed from the standard BSD licence in 1999. The BSD licence template is available from: http://www.opensource.org/licenses/bsd-license.php.

15.2 The BSD licence permits "redistribution and use in source and binary forms, with or without modification" provided that:

(a) copyright notices are retained;

(b) the licence conditions form part of the distribution (for distributions of binary code, the conditions may be set out in documentation or other materials provided with the distribution);

(c) all advertising materials mentioning features or use of the software must display a specific acknowledgement. The pro forma acknowledgement refers to the University of California, Berkeley, although other people who adopt the old style BSD licence insert their own details. This requirement is only present in "old" (ie pre 22 July 1999) BSD licences and licences which take their pedigree from old BSD licences;

(d) no use is made of the University of California to endorse or promote products derived from the licensed product.

15.3 The BSD licence also includes a generic disclaimer which attempts to exclude all liability resulting from use of the software.

16. Case Studies

16.1 NSW Department of Commerce – Linux Panel Contract

The NSW Department of Commerce was in the last stages before issuing a tender for Linux related goods and services. While they had arrived at a policy approach on a number of important issues, they were unsure of how to modify their standard terms and conditions to accommodate open source and their desired policy approach.

In negotiations subsequent to the submission of tenders they were faced with pushback to some clauses from a tenderer.

OSL Response:

We revised their standard terms to:

(a) create a rigorous definition of "open source software";

(b) make specific adjustments relevant to the use of open source software (eg no need for escrow agreement if source is provided);

(c) ensured that closed source components were permitted, but were given treatment different to that of open source components (eg escrow).

During negotiations we identified that tenderer's response did not properly understand the operation of the clauses (or were using open source as a smoke screen to avoid obligations).

Outcome:

Client successfully negotiated Linux contracts with a number of tenderers. Protected client against muddy thinking on open source issues.

16.2 NSW Department of Commerce – Open Source Issues Advice

The client wanted to properly understand the open source legal landscape and issues which they may encounter in respect of open source.

OSL Response:

Provided detailed advice covering, among other things:

(a) methodology for identifying existing open source usage and licences;

(b) overview of effect of open source licences accounting for 95% of projects against agreed measure;

(c) overview of risk categories for open source;

(d) specific examples of risks in each category;

(e) guidance on managing these risks;

(f) outline of business models for commercialisation of open source software;

(g) employment related issues for open source;

(h) template open source policy.

Outcome

Advice will form the basis of an open source information resource for the whole of NSW State Government.

16.3 Open Source Friendly Contracting Agreement

The client wished to engage contractors to do specific work involving the use of open source. Contractors baulked at standard terms, considering that they may be inconsistent with the licensing requirements imposed upon them through the use of open source software.

OSL Response:

Rework standard contract to make it open source friendly. The contract included specific provisions to ensure that the use of open source was subject to the informed consent of the customer, with optional clauses for the open source licence back of material if agreed or required. The contract also included a specific procedure for identifying open source inputs. We also helped highlight to contractors that there is no inconsistency between IP ownership clauses (vesting IP in the customer) and open source licences.

Outcome:

Revised contract used for open source related projects.

16.4 Hybrid Licensing Approach

This client was interested to know what consequences were involved in the adoption of a specific licence for software that they were planning to release. They had heard a lot about the GPL and were wondering about how the use of the GPL would constrain them in the future. The client was concerned that the presence of a GPL version would undermine the revenue stream for the product.

OSL Response

We reviewed the licence terms and their application to the particular circumstances. As the client would own all of the copyright in all of the relevant material the client would need to honour its existing licences in respect of third parties, but was not bound by the requirements of the GPL in respect of future modification or distribution.

Outcome

The client ultimately settled on a delayed licensing approach, with the release of a GPL version “lagged” behind a closed source version (by two versions).

Notes:

Lagged or versioned releases can lead to difficulties in maintaining two code bases which may have a tendency to diverge. The model also requires careful treatment of community contributions, as special care is needed for the inclusion of those contributions under the closed version of the product.

16.5 Licence Compatibility

A client wished to include a number of packages licensed under different licences into a single installer package which could be downloaded as a single file and installed with a single click on a target computer. One of the packages was licensed under the terms of the GPL.

OSL Response:

On our analysis the archive document was a work which included the whole of a work the subject of the GPL – and therefore the archive needed to be licensed under the terms of the GPL. On a conservative reading, this meant that other packages in the archive may need to have GPL compatible licences.

Without disclosing the identity of the client, we initiated informal contacts with the relevant project owners about the executable archive approach and whether, if it was caught by the licence, they would be willing to make an exception. The project owners were not willing to make such an exception.

Outcome

The client chose to adopt a conservative approach and separated out differently licensed components for end users to acquire separately.

Notes:

Do not discount the possibility of a direct approach to the relevant copyright owner. In this case it was not successful.

It is important in all open source cases to be proactive not reactive. There is no negotiation process as it is ordinarily understood, and there is not necessarily any continuing client relationship or any interdependency between licensee and copyright owner. There is little or no opportunity to correct a mistake after the fact.

16.6 Licence Compatibility – Archiving

A somewhat worrying corollary of the licence compatibility study presented above is that if an organisation is running backups and those backups include GPLed software and those backups are sent to a third party for off site storage, then the archive tapes may be required to be licensed under the terms of the GPL! This of course depends on a number of variables including the manner of storage. However, the risk is there.

OSL Response:

We raised this with Eben Moglen of the Free Software Foundation shortly after Linux.Conf.Au in 2005.

Outcome:

I understand there are provisions in GPL v 3 which address this.

Notes:

The paranoid may care to have their backup procedures reviewed by legal.

16.7 University Technology Project – Dealing with Government Grant.

This client was a University technology commercialisation arm. They were in the process of developing a specific product which used open source inputs. The funding grant for the work imposed specific requirements on the licensing of work outputs, including the licensing of third party components. The grant conditions were technically inconsistent with the terms of the third party input. The project also had a third party collaborator which would be providing (non open source) inputs.

OSL Response

We identified the issues raised by the grant terms, highlighted the inconsistencies and argued that the objectives of the grantor would be fulfilled through the substitution of the open source licence terms in favour of the grant terms. Grantor was satisfied that its interests were adequately protected. Renegotiated the grant terms. We also advised on appropriate terms for the provision of input from the third party collaborator.

Outcome

Grant terms were successfully changed to permit the use of open source components.

16.8 Major Hardware Vendor – Preinstalled Open Source

This client is involved in the production of computer systems. They had, over some period of time, received customer feedback to the effect that the preloading of a specific configuration of open source packages would be desirable. The customer's head office/IP section was concerned about what would need to be included in contractual provisions, and what risks would be involved.

OSL Response

We:

(a) provided a briefing to the CEO on issues including suggested approaches;

(b) provided an overview of what competitors are doing in the market;

(c) commented on the availability of insurance and the pricing of risk;

(d) provided a menu of legal conditions that could be used when providing this package. It was important that the terms be able to be interfaced with existing contracts and should appropriately quarantine liability.

Outcome

Package in the process of being taken to market.

Notes:

The market may have different contractual requirements of the provision of open source compared to the provision of closed source. Not recognising this may result in incorrectly costing a product, or making promises which cannot be fulfilled.

16.9 Major Corporate – Adoption of Open Source Package

This client wished to implement well known and long standing open source code in a product, but were concerned about risk. The licence terms for this code were non standard but probably open source compliant.

OSL Response:

We supported the client in making an informed decision on the implementation including:

(a) identifying the appropriate licence terms;

(b) determining whether the licence terms were sufficiently broad;

(c) providing guidance on evaluating the bona fides and credentials of both the package and the person or project from which the package was acquired.

Outcome:

Client weighed risks, decided to adopt package.

Notes:

Much the same process can be used when adopting open source packages.

Creating an open source acquisition policy is a quick and easy first step to take on addressing open source within your organization.

16.10 Major Corporate – Use of Open Source against Infringing Competitor

Client has reason to believe a competitor is infringing an open source package in the competitor's product.

OSL Response:

We have:

(a) helped identify means of determining infringement;

(b) outlined possible consequences (one of which is nuclear);

(c) provided an overview of the process of bringing the infringer to account.

Outcome

Not for disclosure at present.

Notes:

Infringing an open source licence can result in the issue of an injunction restraining the distribution of the product; the award of substantial damages; the disclosure of the infringer's IP and criminal sanctions.

16.11 Multiple Clients – Multiple Issues

Siege Mentality

Typically most organisations are already using open source software in some form or another. Unfortunately, a common process for open source to enter an organisation is as follows (and all of this probably occurred several years ago):

(a) an engineer wants certain functionality which is provided by an open source product;

(b) the engineer puts in a requisition to buy a closed source equivalent;

(c) requisition is rejected or takes too long to approve;

(d) the engineer notices that open source has $0 acquisition costs, so not covered by acquisition policy;

(e) the engineer raises the issue of open source informally with their manager;

(f) manager is mildly to violently opposed, may or may not consult legal, forbids use of open source;

(g) (optional) lawyer provides fire and brimstone advice to the effect that open source is responsible for third world poverty and will make you go blind;

(h) the engineer considers this, and especially the lawyer's response, as irrational;

(i) the engineer acquires open source anyway, satisfies need, no one is any the wiser.

The method of acquiring open source immediately places the engineer on a siege footing. Should management raise open source in the future, the engineer must deny all knowledge. When legal asks the engineer about open source they will avoid the issue or prevaricate. When legal issues an edict that all open source needs to be approved by legal, the engineer interprets this as a blanket prohibition (which it may or may not be).

Why the engineer rejects management and legal's recommendation (or direction) is not entirely clear. However, we can speculate that it is a combination of some of the following factors:

(a) the typical engineer's formidable self confidence;

(b) their refusal to accept arguments from authority (rather than logic);

(c) the comparative unimportance of the software being acquired;

(d) the comparative expense of the closed source alternative;

(e) the need for some result to be achieved quickly;

(f) the lawyer's advice was, all things considered, excessive, over the top, uninformed and, in some places, bordering on the hysterical. What's more it was inconsistent with the engineer's own knowledge of open source and appeared purely reactionary;

(g) the engineer placed their trust in one or more unqualified domain specialists (see below)

Notes:

Absolute prohibitions unsupported by reasoning may be counterproductive in the long run. This has probably already happened within your organisation.

Unqualified Domain Specialists

Another common occurrence (and something of a corollary of the acquisition scenario identified above) is that knowledge of open source licensing and the possible consequences are provided from within the engineering section – by engineers. The basis of this advice is comments by other engineers, and commentary available over the internet. In the great majority of cases this commentary is by another engineer or by some other person who has no qualifications as a lawyer, and, in fact has nothing even remotely resembling legal training.

The engineer probably has a functional knowledge of the open source landscape and are well aware of other, similar, deployments which appear to be working well. They don't pay particular attention to the specific circumstances which are involved, and may not be aware of critical differences (licence wise) between those and their own implementation. As such, when presented with the opinion of a lawyer to which says (effectively) that open source is evil, it is easy for the engineer to dismiss the view as uninformed (going on what is reported in the media it would not surprise me if they were *in fact* uninformed).

The value of the bush lawyering which is present in the open source community is roughly equivalent to that in other areas (although is perhaps more prevalent), which is to say, it is not very valuable. Some common misconceptions include:

BSD permits me to license the code under my own license	Wrong
GPL requires me to distribute my modifications	Wrong
Using contractors to work on “internal” open source projects is no problem	Wrong

OSL Response

In these circumstances it is important to be part of the solution, rather than part of the problem. When faced with these circumstances we have:

(a) provided training to engineers, by way of presentation with an emphasis on practical risks (and judicious use of the views of specific unqualified domain specialists or organisations with moral weight within the open source community);

(b) put together an acquisition policy for open source;

(c) helped to audit and identify existing open source usage;

(d) provided guidance on what to consider to make informed decisions on open source acquisitions.

The good thing about engineers is that they are amenable to persuasion by reasoned argument and exposition of the real (not just theoretical) risks. It is important to make it clear that open source acquisitions will be treated roughly equivalently to closed source acquisitions. This is not all that hard as the issues are largely the same, but the risks have been internalised in the case of closed source acquisitions.

Outcome

Engineering adopted more responsible approach to open source usage.

Notes:

Don't call your lawyer to plan your network topology.

Don't call your technicians to give you legal advice.

Community Acceptance

We often receive inquiries from businesses who want to get involved in open source, and want to do the right thing, but are not clear on what is right and what is wrong in the circumstances. That is, there is a desire to comply with both the letter and the spirit of the law.

OSL Response

These queries can often be resolved with a short phone call. OSL maintains contacts with many leading figures in the open source industry, both within Australia and overseas. We are able to put people in contact with peers who may have an informed view and to make informal contacts on a client's behalf.

17. Counter-studies

17.1 Well Known Organisation – Non Open Source Touted as Open Source

This third party (not a client!) chose to use the term “open source” in a marketing campaign in respect of a licensing initiative. This was motivated by the fact that a key component of the initiative was the use of software licensed under terms which were not OSI approved, but were claimed by their vendor to be compliant with the open source definition. The marketing material clearly signalled a problem to the open source community as it referred to open source in one breath, then described a discriminatory licensing regime in the next (open source licences cannot be discriminatory).

OSL Response:

We were not involved!

The outcomes were easily foreseeable. Had we been asked, we would have told them to:

(a) not market the initiative as open source; or

(b) require the vendor to have their licence OSI approved.

Outcome

They immediately came under fire from a number of directions, and could not justify their material. They had to keep their head down until things blew over. Now, no one mentions the war.

Notes:

It may well be that the licence for the software met the open source definition. However, this was not immediately obvious and you don't want to have to put yourself in the position of having to argue the subtle definitional issues involved in a public forum.

Be cautious about claims to openness. Insist on an OSI Approved licence.

The open source community is wary of imposters. If unsure, find an appropriate sounding board to consult about open source related initiatives prior to announcing them.

18. Further Info on Infringements

See http://gpl-violations.org/.