KEMP LITTLE LLP

OPEN SOURCE SOFTWARE: AN INTRODUCTION

TABLE OF CONTENTS

A. BACKGROUND 2
1. Introduction 2

B. OSS: THE HISTORICAL CONTEXT 3
2. Origins of the OSS movement 3
3. The FSF and the four freedoms 3
4. GNU, the GPL and Linux 3
5. The OSI and the OSD 3

C. OSS: THE BUSINESS CONTEXT 5
6. Reasons for the rapid uptake of OSS 5
7. Reactions to OSS of established technology developers 5

D. THE OSS LICENCES 7
8. The GNU General Public License (GPL) 7
9. The GNU Lesser GPL (LGPL) 8
10. Other common OSS licences 8

E. THE CASES: OSS IN THE COURTS 10
11. The SCO-Linux cases in the USA 10
12. Other US litigation 10
13. gpl-violations.org: Sitecom, Fortinet, D-Link and Skype 11
14. Informal interventions by the FSF 11
15. FSF Europe and BT's Home Hub 12

F. CURRENT OSS LEGAL ISSUES 13
16. Key GPL legal issues: copyleft, 'contains' and GPLs v2 and v3 13
17. The Limits of copyleft: static and dynamic linking 13
18. The limits of copyleft: APIs and code shims 14
19. Key GPL legal issues: bare licence or contractual licence 14
20. Key GPL legal issues: 'Tivoisation' 15
21. Key GPL legal issues – patents 15
22. The limits of copyleft: Software as a Service 16

G. OSS INSIDE THE ORGANISATION 17
23. Defining an organisational policy 17
24. Inbound transactions - Source Code audits, etc 17

PANELS

1. The Open Source Definition. 4
2. GPL, Version 3: summary of key conditions 7
3. Software deployment: virtualisation, SaaS and SoA 16

OPEN SOURCE SOFTWARE: AN INTRODUCTION[1]

A. BACKGROUND

1. **Introduction.** A feature of the software world over the last ten years has been the rise of Open Source Software ('OSS'). From its earliest origins in US academia in the early 1970s[2], OSS emerged into the mainstream in the 1990s[3] and has continued to become increasingly widely used in the 2000s. Looking ahead, its scope and appeal show no signs of diminishing: as new software business models and licensing mechanisms[4] like virtualisation, distributed and grid computing and 'software as a service' gain traction, OSS will continue to become one of a range of options open to a CIO to meet the organisation's software needs.

In essence, OSS is software provided under licence granting certain freedoms to a licensee and should properly be seen as a range of associated licensing techniques: there are many different types of OSS licences, and they differ widely in clarity, length and legal effect. As a practical matter, OSS is frequently the CIO's boon and the GC's burden: whilst use of OSS by the technology group frees up scarce software development resources for other projects, it remains for the legal department to assess the risk that may arise, involving analysis of the OSS code and how it is used and assessing the applicable licences and their impact.

Sections B and C of this article briefly considers OSS in its historical and business contexts; Sections D and E overview key OSS licences and cases; Section F looks at a number of the knottier OSS legal issues that frequently arise; and Section G takes a practical look at OSS inside the organisation. OSS is a broad subject and much has been written about it[5], much of which is available on the internet, and a number of links have been provided in footnotes.

---

[1] By Mark Henley, Associate and Richard Kemp, Partner, Kemp Little LLP, London.

[2] A good description of the OSS movement's history is 'A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture' (Dennis Kennedy, http://www.denniskennedy.com/opensourcedmk.pdf). A good article on how OSS became mainstream is 'Open Source Software Monetized: Out of the Bazaar and into Big Business' (Mahony and Naughton, The Computer & Internet Lawyer, vol. 21. no. 10, October 2004).

[3] Leading Linux developer Red Hat Software released Red Hat Linux in 1995, IPO'd in August 1999, was added to the NASDAQ 100 in 2005 and moved to the New York Stock Exchange in late 2006.

[4] These techniques are outside the scope of this article, but see panel 3 below for a brief overview.

[5] A must read for anyone wanting to get into this subject in detail is 'Open Source Licensing, Software Freedom and Intellectual Property Law' (Prentice Hall, 2004) by Lawrence Rosen, a past General Counsel of the Open Software Initiative.

B. OSS: THE HISTORICAL CONTEXT

2. Origins of the OSS movement. The origins of the OSS movement can be traced back to a cultural attitude of opposition to the restrictive nature of exclusive rights under intellectual property laws in US academic circles, itself an echo of the counter culture of the 1960s. It was particularly antithetical to the commercial exploitation in the 1970s and 80s of the operating system UNIX developed by AT&T employees at Bell Laboratories.

3. The FSF and the four freedoms. In 1985 Richard Stallman[6], an ex-MIT academic, established the Free Software Foundation[7] ('FSF'), a non-profit body dedicated to the development of Free Software. The FSF would oversee the GNU Project, hold the copyright in the software created for it and enforce the licences. To be considered Free Software, the FSF identified four essential freedoms for any software licensee[8]:

- to run the software for any purpose;
- to study how the software works and adapt it;
- to redistribute copies of the software; and
- to improve the software, and release those improvements.

4. GNU, the GPL and Linux. In 1983 Stallman had announced the GNU[9] Project, a mass collaboration to create free software whose first goal was to create a full operating system as a UNIX replacement. The licence adopted for the GNU Project's software was the GPL – the GNU General Public License (GPL). By 1992 all necessary components had been completed except the operating system kernel[10], then known as GNU Hurd. In 1992, this gap was filled when the GNU software was combined with a new kernel called Linux[11] to create a complete operating system. This combination, known as GNU/Linux, was licensed under the GPL.

5. The OSI and the OSD. By the late 1990s, some parts of the OSS community considered that the anti-IP sentiments of Stallman and others were inhibiting the widespread take up of OSS.

---

[6] Personal home page at http://www.stallman.org/

[7] www.fsf.org

[8] The second and fourth freedoms imply access to the source code.

[9] A recursive acronym for 'GNU's Not Unix'

[10] The kernel is fundamental to the operating system. Modern operating systems are typically built in layers, with each layer adding new capabilities. The foundation on which the rest of the operating system sits is typically called the kernel. The kernel will determine how much and the order in which processing time is allocated to each program. A kernel may also manage its own memory space, sharing it among system components and other programs that use the kernel's services. A kernel's services are requested by other parts of the operating system or by applications through a specified set of program interfaces sometimes known as system calls.

[11] Linux was developed by Linus Torvalds, a computer sciences specialist at the University of Helsinki. Incorporating elements from MINIX, a precursor Free/OSS operating system also related to UNIX (MINIX means 'minimal UNIX'), Torvalds and his collaborators coded a completely new operating system kernel known as Linus's MINIX, itself shortened to Linux. Linux was first publicly released in September 1991.

In 1998 Bruce Perens[12] and Eric Raymond[13], leading members in the OSS movement, established the Open Source Initiative[14] ('OSI') to promote OSS on pragmatic rather than ethical or philosophical grounds[15]. The OSI is the steward of the Open Source Definition[16] ('OSD') and part of its function is to review and approve licences conforming to the OSD. Many different licences satisfy the OSD, but the types of obligations they impose can vary quite widely. The OSD has developed ten criteria to determine whether a licence for software is open source (see panel 1).

Panel 1 - The Open Source Definition

1. **Free Redistribution:** software to be available for redistribution without payment.
2. **Source Code:** The software to be distributed with the source code or well-publicised access to it.
3. **Derived Works:** license to allow modification of the software and distribution of resulting derived works.
4. **Integrity of The Author's Source Code:** distribution of "patch files" used to recreate the derived work (rather than full source code) to be permitted.
5. **No Discrimination Against Persons or Groups**
6. **No Discrimination Against Fields of Endeavor:** for example, limiting to non-commercial purposes is not permitted.
7. **Distribution of License:** must be no need to execute extra licences for redistributed software.
8. **License Must Not Be Specific to a Product:** licence rights not to depend on the software being distributed with other specified software.
9. **License Must Not Restrict Other Software:** The license must not place restrictions on software distributed together with the licensed software.
10. **License Must Be Technology-Neutral**

---

[12] Personal home page at www.perens.com

[13] Personal home page at http://www.catb.org/~esr/

[14] http://www.osi.org

[15] The OSI's views were most memorably advocated in the essay by Eric Raymond entitled 'the Cathedral and the Bazaar' (http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/) where the cathedral represents the structured world of proprietary software development and the bazaar the looser communitarian approach.

[16] The full definition is at http://opensource.org/docs/osd. The OSD was written by OSI co-founder Bruce Perens and is based on his earlier work for the Debian Project – a project to create and maintain a free operating system based on GNU/Linux.

C. OSS: THE BUSINESS CONTEXT

6. Reasons for the rapid uptake of OSS. There are many reasons for the OSS model's popularity. For many projects, a sort of OSS eco-system develops around a community of programmers who work constantly to improve the code and provide bug fixes and additional functionality free of charge. Major OSS projects usually employ rigorous peer review mechanisms to marshal all contributions into a coherent, consistent and stable product. As a result, code adopted for an OSS product can be of at least as high a standard as software produced commercially. With source code readily available, OSS can be adapted to work on new hardware as and when available. Developers and users alike know that the OSS product will not become obsolete simply through the obsolescence of the original hardware platform (as might happen to proprietary software if maintaining a version for the obsolete platform was no longer commercially attractive).

Competitive pressures increasingly take time out of the product cycle – the period between the time when design starts and first customer availability. Using pre-made OSS components (particularly for routine, lower level tasks) shortens the development phase and frees up internal resources to develop higher level software that confers competitive advantage. This trend applies across most industry sectors where technology transforms business but is becoming particularly noticeable in the consumer electronics sector[17].

Finally, in developing countries, the introduction of OSS can be used to establish a local software industry and as a result OSS is becoming widely adopted by governments in countries such as China, Brazil, India and South Africa. Customisation, integration and support services for software packages can be provided locally if source code access is available.

7. Reactions to OSS of established technology developers. Faced with the rise of OSS, the responses of established technology vendors differed. Software-only developers initially showed more hostility than vendors who could leverage hardware sales out of their OSS investment. Recently, increasing OSS mainstream use and 'monetisation'[18] of OSS has resulted in edgy detente if not rapprochement, although tensions remain under the surface, especially relating to patents.

For example, Microsoft is encouraging OSS vendors to enter into patent co-operation agreements and in November 2006 it entered into an agreement with Novell under which it promised not to assert its patent rights against customers purchasing SUSE Linux from Novell[19]. In mid 2007 it

---

[17] For example, mobile handsets: The Open Source Cellphone: Lead Users and Early Adopters in the Mobile Phone Industry, Anmol Madan, Media Lab, MIT (http://web.media.mit.edu/~anmol/351-madan-1.pdf)

[18] See Mahony and Naughton cited at note 2 above.

[19] http://www.microsoft.com/presspass/exec/steve/2006/11-02NovellInterop.mspx. Microsoft would also pay Novell for 350,000 "subscription coupons" for SUSE Linux which it could sell to its customers. Coupons could be redeemed from Novell for single- or multi-year subscriptions, upgrades and technical support.

entered into patent co-operation agreements with two more Linux distributors, Xandros[20] and Linspire[21].

In March 1999, IBM[22] established a Linux Technology Centre to work on developing OSS which now employs more than 300 Linux kernel developers. IBM has one of the largest technology patent portfolios in the world and in 2004 pledged not to use its patents against Linux.

Sun Microsystems[23] is a significant proponent of OSS and arguably the largest corporate contributor to the Open Source movement. In 2005 Sun open sourced its flagship operating system with the OpenSolaris project and it continues to employ hundreds of software engineers to work on OpenSolaris. The OpenOffice.org[24] project was created after Sun acquired the German software company StarDivision in 1999. StarOffice, StarDivision's office productivity suite, was released as OSS in 2000.

In addition, a consortium of companies formed the Open Invention Network[25] (OIN) to defend the OSS movement from patent infringement actions. OIN members include major technology companies such as IBM, Google, NEC, Novell, Philips, Red Hat, and Sony. Each agrees not to use its Linux-related patents against each other. The OIN has also begun purchasing relevant patents and provides free licences to all its members.

---

[20] http://www.microsoft.com/presspass/press/2007/jun07/06-04XandrosPR.mspx

[21] http://www.microsoft.com/presspass/press/2007/jun07/06-13LinspirePR.mspx

[22] http://www.ibm.com/developerworks/opensource/

[23] http://www.sun.com/software/opensource/

[24] http://www.openoffice.org/

[25] http://www.openinventionnetwork.com/

D. THE OSS LICENCES

There are many OSS licences, ranging from the intrusive, ‘copyleft’ GPL through to short licences containing virtually no express terms. As a practical matter, the key thing is invariably to identify the OSS concerned and the licence terms under which it is made available and then to assess whether the licence attaches any particular terms.

8. The GNU General Public License (GPL). The GPL was written by Richard Stallman and the FSF in 1989. Since then its underlying philosophy and the skilful way in which it turns intellectual property law against itself have made it popular. It is the licence for the GNU Project and GNU/Linux and for programmers subscribing to the FSF’s Free Software ethic.

The GPL protects the four essential freedoms identified by the FSF[26]. The legally radical mechanism to achieve this and the thing for which the GPL is best known was called ‘copyleft’ by the FSF. Copyleft is the idea that the freedoms guaranteed by the GPL would also apply to new works derived from the original GPL-licensed software. Copyleft would serve to expand the rights of the public, in contrast to the traditional role of copyright which grants exclusive rights to the author of new software.

Version 1[27] of the GPL was issued in 1989. Version 2[28] (GPLv2), the licence Linus Torvalds chose to apply to the Linux kernel, was issued in 1991. Version 3[29] (‘GPLv3’) was published in its final form and adopted on 29 June 2007.

Panel 2 – GPL version 3: summary of key conditions

Section 0 – definitions: GPL v3 uses the terms “propagate” and “convey” instead of the term “distribute” used in GPLv3. “Distribute” had different specific meanings in different jurisdictions.

Section 1 - source code: establishes the “corresponding source” for a program in object code form as including source code but also scripts for controlling compilation and installation of the object code.

Section 2 - basic permissions: clarifies that the rights under the licence are perpetual and irrevocable whilst the GPL conditions are met. The program may be made, run and propagated without regard the GPL conditions, as long as the work is not conveyed. “Conveying” means enabling other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

Section 3 - protecting users' legal rights from anti-circumvention law: prohibits using any program released under the GPL as a “technological protection measure” as defined by the US Digital Millennium Copyright Act or similar laws.

---

[26] See section B above.

[27] http://www.gnu.org/licenses/old-licenses/gpl-1.0.txt

[28] http://www.gnu.org/licenses/old-licenses/gpl-2.0.html

[29] http://www.fsf.org/licensing/licenses/gpl.html

Section 4 - conveying verbatim copies: clarifies that a party may charge money for each copy it conveys and may offer support or warranty protection for a fee.

Section 5 - conveying modified source versions: conveying "a work based on the program" or "the modifications to produce it from the program, in the form of source code" requires that certain conditions to be met, including licensing the entire work under the GPL.

Section 6 - conveying non-source forms: specifies how source code should be provided.

Section 7 - additional terms: deals with additional terms that may be used to supplement the GPL terms (to permit wider compatibility with other OSS licences).

Section 8 – termination: confirms there is no right to use the program except by the licence, and that failure to comply with its conditions will result in termination of the licensee's rights. Termination under this section will not affect the licences of parties who have received program copies under it. It also introduces a process for reinstating the licence on cessation of violations.

Section 9 - acceptance not required for having/running copies: but copying, distributing or modifying the program indicates acceptance of the licence (without the rights granted by the GPL these actions would constitute copyright infringement).

Section 10 - automatic licensing of downstream recipients: clarifies that downstream recipients of the program are automatically licensed by the original licensors. No right to sublicense.

Section 11 – patents: (see section E).

Section 12 - no surrender of others' freedom: states that conditions imposed by a third party contradicting the GPL do not excuse the licensee from its conditions.

Section 13 - use with the GNU Affero General Public License ('AGPL'): permits the linking or combination of code licensed under the GPL with code licensed under the AGPL, which relates to networked use of software.

Section 14 - revised versions of this licence: clarifies that some versions of a program may be covered by one or more versions of the GPL.

Sections 15 and 16 – broad disclaimer of warranty and limitation of liability: these provisions also recognise that a warranty or liability may be accepted in return for a fee.

9. The GNU Lesser GPL (LGPL). The LGPL[30] is the second of the FSF's two main licences and, as the name suggests, is less intrusive than the GPL. It is frequently used for software libraries that will be utilised by proprietary programs.

10. Other common OSS licences. The OSI publishes the licences that it has approved as conforming with the OSD on its website at http://www.opensource.org/licenses/alphabetical. At August 2007, 51 licences were listed of which 41 are in active use. In contrast to the lengthy and technically complex GPL, there are licenses which are relatively short, very permissive and which

[30] http://www.gnu.org/licenses/lgpl.html

impose few, if any, conditions. Two of the most commonly used are the MIT and BSD Licenses. The MIT License[31] is little more than a copyright notice with a broad permission to deal (and sub-license) the software and a broad warranty and liability disclaimer. The BSD License[32] also consists of a copyright notice, a broad permission to redistribute and use the software (with or without modification) and a warranty and liability disclaimer and also adds three conditions: (i) that redistributions of source code retain the licence terms; (ii) that redistributions of object code include the licence terms in accompanying documentation or other materials; and (iii) that the name of the licensor should not be used to endorse or promote products derived from the software without permission.

[31] http://www.opensource.org/licenses/mit-license.php

[32] http://www.opensource.org/licenses/bsd-license.php

E. THE CASES: OSS IN THE COURTS

11. The SCO-Linux cases in the USA. The highest profile examples of OSS in the courts stem from the SCO-Linux series of cases[33]. This is a series of US cases between SCO Group (SCO) and various GNU/Linux end users and distributors and Novell.

In 1990 AT&T, which had created the UNIX operating system in 1969, sold all its rights in UNIX to Novell. In 1995 Novell sold certain parts of its UNIX business to the Santa Cruz Operation and in 2000 Santa Cruz Operation sold its entire UNIX business to Caldera Systems, which then changed its name to SCO Group.

SCO began making statements that it owned the rights to UNIX and that it would seek royalties from GNU/Linux users and distributors. Novell then claimed that the copyright in UNIX had not been assigned in the 1995 sale and began registering the copyrights to certain UNIX products. SCO filed suit, claiming slander of title[34]. In the meantime, SCO had begun suits against two other GNU/Linux distributors, IBM (alleging that IBM was in breach of a contract with SCO when IBM provided source code for the GNU/Linux code base[35]) and Red Hat[36]. SCO also sued some of its users (including AutoZone[37] and DaimlerChrysler[38]) who had stopped using SCO's UNIX in favour of GNU/Linux. On August 10, 2007, the federal district court judge decided that Novell had retained the copyright to UNIX and not assigned it to SCO[39]. This ruling is likely to prove fatal to the other cases brought by SCO and on September 14, 2007 SCO filed for Chapter 11 bankruptcy protection.

12. Other US litigation: Considering the widespread popularity of OSS and the GPL in particular, aside from the SCO litigation there has been very little in the way of US judicial authority on the enforcement of OSS licences against licensees[40]. However, the dearth of reported cases may be misleading; the FSF and, more recently, the Software Freedom Law Center ("SFLC") have taken active roles in the US in ensuring compliance with the GPL outside the courts. The SFLC provides legal representation and other law-related services to protect and advance OSS. It is chaired by Eben Moglen, former General Counsel of the FSF. In September 2007 the SFLC

[33] A good, comprehensive resource for further information on the SCO litigation is http://www.groklaw.net/

[34] http://www.groklaw.net/staticpages/index.php?page=20040319041857760

[35] http://www.groklaw.net/staticpages/index.php?page=20031016162215566

[36] http://www.groklaw.net/staticpages/index.php?page=20031017044328636

[37] http://www.groklaw.net/staticpages/index.php?page=AZ-Timeline

[38] http://www.groklaw.net/staticpages/index.php?page=DC-Timeline

[39] http://www.groklaw.net/pdf/Novell-377.pdf

[40] However one relevant case which has caused a stir in the OSS community is ***Jacobsen v Katzer and Kamind Associates, Inc*** [Case No. C 06-01905 JSW]. Robert Jacobsen, a developer of OSS model train software, has alleged that Matthew Katzer and Kamind Associates Inc. infringed his copyright by using his software outside the scope of the OSS licence. The US District Court for the Northern District of California made a controversial decision in denying an interim injunction (see section 19 below and http://www.thelen.com/tlu/JacobsenVKatzer.pdf) which has since been overturned by the Federal Circuit Court of Appeals (see http://jmri.sourceforge.net/k/docket/cafc-pi-1/08-1001.pdf).

launched the first US copyright infringement lawsuit based on non-compliance with the GPL. It has filed six further lawsuits since then and most have settled out of court.

13. gpl-violations.org: Sitecom, Fortinet, D-Link and Skype. Gpl-violations.org was founded by Harald Welte[41] in Germany in January 2004. Most of Welte's successes have been achieved informally outside the courts, but he has brought a handful of cases in Germany.

In 2004 the German lower regional court of Munich[42] confirmed a temporary injunction enjoining the distribution of OSS in breach of the GPL's requirements. The defendant, Sitecom, had used netfilter/iptables without providing access to the source code. The court ruled[43] that Sitecom was not entitled to use the netfilter/ip-tables code for its proprietary products and prohibited it from distributing them. The court, upholding the decision made at the interim hearing, held that the GPL licence terms had been validly agreed between the parties by way of standard licence terms and conditions and that the defendant was in breach of the licence.

In April 2005, gpl-violations.org brought an action against Fortinet UK for using GPL software in firewall and anti-virus products and trying to conceal the fact using cryptographic techniques. The Munich district court granted a preliminary injunction against Fortinet Ltd., banning further distribution of those products until they complied with the GPL[44].

In 2006, gpl-violations.org brought a claim in the Frankfurt district against D-Link for selling a data storage device that used the Linux kernel. It claimed D-Link had sold the device without enclosing the GPL text, disclaiming any warranty or disclosing the source code. The court in its July 2006 ruling[45] treated Section 4 of the GPL as a condition subsequent which, when broken, revoked D-Link's licence to use the software.

In July 2007 it was reported[46] that Skype had been found in breach of the GPL by a regional court in Munich. The breach was said to involve the manner in which Skype had distributed a VoIP handset which used an embedded Linux kernel, having failed to supply the source code with the handset. Welte reported in May 2008[47] that Skype had accepted that judgment and withdrawn its appeal.

14. Informal interventions by the FSF. Although the FSF, like gpl-violations.org, has made informal approaches to users of GPL-licensed software allegedly breaching the licence conditions, these approaches do not tend to attract publicity. The first step towards enforcement generally

---

[41] Welte wrote the netfilter/iptables software used to provide firewall functionality in the Linux kernel which gives him standing to bring actions for copyright infringement against Linux kernel users who disobey the GPL's conditions.

[42] LG München, dated May 19 2004, Az. 21 O 6123/2004

[43] An unofficial translation can be found at http://www.jbb.de/judgment_dc_munich_gpl.pdf

[44] http://gpl-violations.org/news/20050414-fortinet-injunction.html

[45] An unofficial English translation of the case (6.9.2006 no. 2-6 O 224/2006) is at http://www.jbb.de/judgment_dc_frankfurt_gpl.pdf

[46] http://yro.slashdot.org/article.pl?sid=07/07/24/174240&from=rss

[47] http://laforge.gnumonks.org/weblog/2008/05/08/

comes when OSS activists inspect a software or hardware product looking for signs that GPL software has been incorporated. If they believe that GPL software has been used in breach of the GPL conditions, they will generally post the details to their blogs or to the online forums of gpl-violations.org or the FSF. Other activists will then re-analyse the products to verify the initial findings. The negative publicity that this starts to create within the OSS community may itself be sufficient to encourage GPL compliance. If not, gpl-violations.org or the FSF may enter into a dialogue with the company allegedly in breach. There are very few examples of informal dialogue failing to resolve the issue, but in principle the final stage could involve the matter being brought before the courts. As at August 2008, we are not aware of any court actions that FSF Europe has brought in the UK.

15. FSF Europe and BT's Home Hub. One case that is reported to have attracted some notice, however, was the intervention by FSF Europe over BT's Home Hub[48], a network device that utilises the Linux kernel. BT was challenged by FSF Europe for distributing the Home Hub without making available the firmware source code. BT admitted that the Home Hub used Linux kernel version 2.6.8.1 under GPL v2, but stated that it also used proprietary software which it claimed was not subject to the GPL. In January 2007 FSF Europe notified BT of its claim that it was breaching the GPL. Since then BT has published source code for certain parts of the firmware[49] but FSF Europe argued that some of the necessary code was still missing, saying that the GPL required publication of a top level Makefile (a file used to assist compilation), the scripts that would be used to generate a firmware image and also a script or file containing configuration information for certain library files. In spite of FSF Europe's claims, no further action appears to have been taken against BT.

[48] http://www.theregister.co.uk/2007/01/29/bt_says_enough_gpl/

[49] See http://www.btyahoo.com/broadband/adhoc_pages/gplcode.html

F. CURRENT OSS LEGAL ISSUES

16. Key GPL legal issues: copyleft, 'contains' and GPLs v2 and v3. Section 2b of GPLv2 has been the subject of considerable debate and the source of a significant amount of confusion. It sits at the heart of the copyleft mechanism and reads as follows:

> "2 You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work… provided that you also meet all of these conditions:
>
> b) You must cause any work that you distribute or publish, that in whole or in part **contains or is derived from**[50] the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License."

The confusion arises from the ambiguous use of the word "contains." The concept of a derivative work is well understood in US copyright law, but "contains" is open to widely varying interpretations. The static vs dynamic linking debate will be discussed below, but "contains" is a fundamental part of this problem – if software is linked to a GPL program, when should they be regarded as together forming a new program that "contains" the GPL program? When should they be regarded as remaining separate and independent works? The FSF, Linus Torvalds and other professionals from the IT industry have all formed their own individual views of the answer. Fortunately, in GPLv3 the reference to "contains" has been omitted which should reduce the scope for confusion and misunderstanding.

17. The limits of copyleft: static and dynamic linking. A programmer rarely needs to write an application entirely from scratch and will usually take advantage of code already written and stored in libraries by placing in his program links to the other programmers' code for the software compiler to use. If the compiler is instructed to bind the code from the library permanently into the executable for the new program, that is called "static linking." If the code remains separate, but the executable (i.e. compiled code) draws from the library when it is running, that is called "dynamic linking."

There is a strong argument that static linking creates a work that "contains or is derived from" (in the language of GPLv2) the GPL-licensed work. Where static linking takes place the compiler binds together the GPL-licensed library with the new program in a single piece of executable code. It is not difficult to see why this new piece of executable code is considered by many to be a work which "contains or is derived from" the GPL-licensed work. Conversely, where dynamic linking is present there is no permanent binding together of the GPL-licensed work and the new program; only transitory copies which form in the computer's memory when the new program is run. The case for creation of a single work containing or deriving from the GPL-licensed work is weaker but not unarguable. The view of the FSF[51] is that what constitutes combining two parts into one program depends both on the mechanism of communication (how intimately the two are connected) and the semantics of the communication (what kinds of information are interchanged).

---

[50] emphasis added

[51] http://www.gnu.org/licenses/gpl-faq.html#MereAggregation

18. The limits of copyleft: APIs and code shims. An issue may arise where proprietary device drivers are used with OSS such as GNU/Linux. Device drivers tend to be loaded by the Linux kernel as kernel "modules." These modules communicate intimately with the kernel which, for the reasons given in the preceding paragraph, has led many in the OSS community to claim that Linux kernel modules must also be licensed under the GPL. However, some hardware vendors, especially the developers of the most advanced graphics cards, permit their hardware to be used in GNU/Linux machines but decline to publish the source code for the drivers required for that use.

This has led to the development of what may be called 'acceptable Linux development practice' (although that is perhaps overstating it) for both sides where developers use APIs[52] or code shims[53] to buffer or insulate their own code from the 'copyleft' effect of the GPL under which GNU/Linux is licensed. The Linux kernel communicates via an API or code shim, which in turn communicates with the proprietary device driver. In the absence of settled case law on this issue[54], this practice cannot be regarded with certainty as having its intended effect although many IT industry professionals accept that the proprietary device driver is not a derived work of the Linux kernel which must be licensed under the GPL. As a practical matter, particularly where a shim is used with GNU/Linux, care needs to be taken to ensure that the shim is not simply a device to get around the effect of the GPL with no other useful purpose.

19. Key GPL legal issues: bare licence or contractual licence. A common question about the GPL is whether it constitutes a bare licence or a contractual licence. The distinction is important because the terms of a contractual licence may be enforced against the licensor. A bare licensee, on the other hand, cannot bring a claim against the licensor. Contractual terms may also be implied or disallowed by the courts in certain circumstances. Third, if the GPL is a contract then it is possible that the remedy of specific performance might be granted by a court and this would be a powerful remedy if used to compel a licensee to provide access to source code.

The publicly expressed views of the FSF[55] are that the GPL is a bare copyright licence with conditions attached and that breach of those conditions automatically results in the loss of the licence to distribute. However, some commentators characterise the GPL as a contract. Initially a unilateral contract – an offer made to the world by the author to use his/her software in compliance

---

[52] Colloquially called 'hooks and handles' an API (application programming interface) is an interface in the source code that an operating system (or library or other application) provides in order to be able to respond to requests for service from another computer program.

[53] A software shim a small piece of software providing a bridge between two other programs. It takes its name from a shim in engineering, where a thin piece of material is used as a spacer to fill small gaps between objects to provide a better fit. The code shim between the Linux kernel and the proprietary code is typically provided under a licence that is compatible with the GPL, but which has no copyleft provisions. The FSF maintains a list of licences that it believes are compatible with the GPL at http://www.gnu.org/philosophy/license-list.html#GNUGPL.

[54] There has been reticence on the side both of the OSS community and the proprietary software world to fire the first shot over whether copyleft reaches past APIs and code shims. This is understandable as, quite apart from the time and costs involved in protracted litigation, anything other than outright victory for the FSF side would diminish the effectiveness of their informal intervention approach (see Section E above) whilst an adverse ruling for proprietary developers would have major consequences for future software and hardware development. Furthermore, success by the FSF might lead to hardware vendors withdrawing their support for GNU/Linux machines.

[55] http://www.gnu.org/press/mysql-affidavit.html at paragraphs 18 and 22.

with the GPL conditions – where the normal requirement to communicate acceptance is waived by the licensor[56] so that when the software is modified or distributed, the offer is accepted by conduct and a bilateral contract is created. If the GPL is characterised in court as a copyright licence only (not a contract), which would seem likely given the FSF's position on the subject, then the relief available in a UK court would be damages, an account of profits (or a hybrid of the two) and an injunction to restrain further infringement. Of considerable concern to the OSS movement was ***Jacobsen* v *Katzer and Kamind Associates, Inc*.**,[57] when, at an interim hearing, the US District Court for the Northern District of California found that breach by the licensees of their obligations under a non-exclusive OSS licence gave rise to a claim for breach of contract rather than a claim for copyright infringement. Leading members of the OSS community including the OSI, SFLC, Linux Foundation and Creative Commons Corporation submitted as amici curiae a brief supporting an appeal from the District Court's decision. On 13 August 2008, the US Court of Appeals for the Federal Circuit vacated the District Court's decision, finding that the licensees' obligations were conditions limiting the scope of the licence and not independent contractual covenants. By failing to comply with those conditions, the licensees had acted outside the scope of the licence and therefore an action for copyright infringement could be brought by the licensor. The injunction application was remanded back so that the District Court could make a finding on the licensor's likelihood of success on the merits.

20. Key GPL legal issues: 'Tivoisation'. 'Tivoisation' is the term given to a technical method used by Tivo, Inc.[58] for preventing modifications to its set top box (STB). It is a controversial practice because the Tivo STB is also built on the Linux kernel, for which the GPL guarantees (amongst other things) the freedom to make modifications. Although Tivo publishes the STB software in a high level programming language, it has used Digital Rights Management (DRM) effectively to remove the ability to make modifications. The hardware uses DRM techniques to check whether its software has been modified and refuses to run it if a certain signature is not present. The algorithm for producing a signature is not published so, effectively, only Tivo can modify the STB software.

Section 6 of GPLv3 seeks to outlaw 'Tivoisation' by requiring a licensee distributing software with a consumer product to accompany the source code with "Installation Information", meaning any methods, procedures, authorization keys, or other information required to install and execute modified versions of the software in the consumer product. The installation information must be sufficient to ensure that the mere act of modification does not cause the consumer product to stop working.

21. Key GPL legal issues: patents. GPLv3 goes further than GPLv2 in taking steps to defend OSS from further threats of patent infringement claims and Section 11 aims to prohibits future deals similar to the one struck by Microsoft and Novell in November 2006[59]. It holds that if a patentee

---

[56] Section 5 of the GPL version 2 states "…by modifying or distributing the Program (or any work based upon the Program), you indicate your acceptance of this License to do so…"

[57] See note 40.

[58] www.tivo.com

[59] See note 19 above.

distributes GPL-licensed software and grants a patent licence to some of the parties receiving that software, then the patent license will automatically extend to all software recipients and to any works based on it.

22. The limits of copyleft: Software as a Service. Software as a Service (SaaS) describes where software is hosted by a company and made available to users indirectly via a web browser. There has been considerable controversy over whether the source code for OSS hosted by a SaaS provider must be made available to the users. Under the wording of current OSS licences (except the GNU Affero General Public License), the hosting of OSS software by a SaaS provider would not appear to fall within the definition of redistribution. Indeed, Section 0 of GPLv3 notes that mere interaction with a user through a computer network, with no transfer of a copy of a program, is not conveying and as a result, the obligations to publish source code may not be triggered.

Panel 3 – Software deployment: virtualisation, SaaS and SOA

Lawyers are well used to reviewing software licences, whether the software is licensed directly or indirectly, and whether perpetually or periodically. OSS is the first to hit the mainstream of a number of overlapping new techniques for deploying software with which lawyers will need to become familiar, including:

- **Virtualisation.** Where software is used to run one or more Operating Systems on a host computer including Operating Systems not written specifically for that hardware (platform virtualisation) or to reach the computing resources that ordinary software cannot reach by aggregating a large number of individual computing resources into a smaller number of more powerful resources (resource virtualisation). The successful IPO of VMWare in August 2007[60] shows the increasing demand for virtualisation solutions, which can improve infrastructure optimisation, business continuity, software lifecycle management and desktop management. Legally, the novel problems that virtualisation creates include the difficulty in applying and enforcing traditional licensing parameters (for example, when new "virtual" computers can be created or destroyed by employees instantly in software) and in defining and measuring performance metrics across a heterogeneous and distributed hardware platform.

- **SaaS: software as a service.** IT lawyers are familiar with ASP (application service provision), effectively an internet version of the old bureau services/remote host model. SaaS goes further in harnessing the ubiquity of broadband and the greater sophistication of web browsers and scripting languages. The SaaS supplier develops, hosts and maintains software applications and its customers experience a service using only a web browser. SaaS has become an umbrella term subsuming ASP and some aspects of on-demand or utility computing or rental-based software delivery models. Legal issues are concerned with the move away from pure licensing and towards service provision, with risk transfer to the SaaS provider, service levels, etc.

- **SoA: service oriented architecture.** Effectively an evolution of distributed computing, this technique enables an application to be built from various software modules, each providing specific services. The application can be constructed by reference to the business processes required, rather than by reference to the compatibility of the different software components or their platforms.

[60] http://www.vmware.com/company/news/releases/IPOpricing.html

G. OSS INSIDE THE ORGANISATION

23. Defining an organisational policy. Implementing an organisational policy on OSS use is critical for ensuring that employees do not use OSS in a way which will bring adverse unintended consequences. Any use of OSS should have an express determination by management that it will not conflict with the company's business model. There is a common misconception amongst programmers that OSS is "free" in all senses of that word. It is sometimes confused with "public domain" software, where the author has disclaimed any intention to enforce his/her rights against users of the software. In fact, uncontrolled use of OSS could oblige the organisation to share proprietary source code or else face an injunction removing its products from sale. Another risk introduced by uncontrolled use of OSS is of third party IP infringement claims. Code of unknown or dubious origin can bring risks of third party claims and by default OSS licences expressly disclaim any warranties or indemnities.

An organisation should consider establishing a policy for the types of OSS code that will be accepted for its products. It should consider when and with what level of legal supervision the company will accept OSS. In particular, it should consider whether the contributors to a particular OSS project have been authorised to take part by their employers and the terms on which they have been allowed to make their contributions. It may be that the contributors purport to contribute software without the full authorisation of the owners of the relevant IP, i.e. their employer. There are a number of other issues relating to the interaction between employees and the OSS community. In particular, there is a risk that employees inadvertently disclose the company's confidential information or license the company's valuable intellectual property. There is also a risk that an employee will contaminate the company's IP with third party IP acquired from the OSS community. In the event that the company ever needed to enforce its IP rights against a third party, it would first have to establish a clear chain of title, unravelling the effects of any third party IP contamination in the process.

24. Inbound transactions - Source Code audits, etc. Overlapping with the organisational policy, organisations should carefully consider treatment from the OSS perspective of inbound transactions (for example, company acquisitions or software purchases where Intellectual Property Rights are assigned) from the OSS perspective. This applies whether the organisation is acquiring or disposing of companies whose assets include software, and whether that software is owned by or licensed to the company: in reality this will be relevant for most company acquisitions of any size. As well as corporate transactions, organisations should also consider their procurement operations to make sure they have an effective way of verifying that code they buy or license in does not contain unexpected OSS. Black Duck Software[61] and Palamida[62] are two companies that provide automated audits of OSS source code. They provide consulting services for specific development projects and are able to generate a report which identifies the origins of the code included in any particular project's code base. They also identify the applicable OSS licences for management or the legal department.

Kemp Little LLP (MSH/RHK), August 2008

---

[61] http://www.blackducksoftware.com/

[62] http://www.palamida.com/