The emerging role of open-source software in merger analysis

By

Scott Sher
Charles Biggio
Ramsey Shehadeh
Jonathan Lutinski

***Reprinted from* ECLR**
Issue 7, 2011

Sweet & Maxwell
100 Avenue Road
Swiss Cottage
London
NW3 3PF
(Law Publishers)

SWEET & MAXWELL

The emerging role of open-source software in merger analysis

Scott Sher

Charles Biggio

Ramsey Shehadeh

Jonathan Lutinski

Comparative law; EU law; Licensing; Mergers; National competition authorities; Open-source software; United States

Introduction

Open-source software is having an increasingly material impact on software markets. A recent International Data Corporation (IDC) study found that open-source software revenues are growing at a 22.4 per cent compound annual growth rate, and will reach $8.1 billion by 2013.[1] One expert explained:

> "The open-source software market has seen a strong boost from the current economic crisis … [it] is increasingly a part of the enterprise software strategy of leading businesses and is seeing mainstream adoption at a strong pace."[2]

Open-source software can be a substantial market disruptor. One of the most famous examples is Linux, which serves as the operating system for over 40 per cent of the websites on the internet.[3] Another is Apache HTTP Server, which serves approximately 54 per cent of the websites.[4] On the consumer software side, Firefox, developed by Mozilla, is the number two internet browser. These examples are important not only because of their obvious commercial significance as complex products that can run at the core of an enterprise, but also because they offer credible competition to proprietary software vendors, and in some cases displace them entirely.

Despite its commercial significance, the role that open-source software plays in merger market analysis remains ill-defined. There is little precedent at the enforcement agencies in the United States, Europe and elsewhere regarding open-source software in merger analysis. As a result, merging parties are left wondering whether, how, and to what extent the Department of Justice (DOJ) and Federal Trade Commission (FTC) (or foreign regulatory authorities, most notably, the European Commission (EC)) consider open-source software in their competitive analyses. In this article, we examine several scenarios in which open-source may be relevant to the antitrust analysis of a software merger:

- Where open-source software is available in the market and two leading providers of proprietary software merge, leaving few or no commercial providers of alternative proprietary software.
- Where a firm with a market-leading proprietary product acquires a firm that provides a successful open-source alternative.
- Where two firms with competing open-source software solutions merge, leaving no additional open-source software alternatives in the market, but there are competitive closed-source alternatives.

In this article, we first discuss the nature of open-source software to allow for proper assessment of the competitive effects of mergers where open-source software is the target of the transaction or a potential alternative to the merging parties' products. Following that discussion, we analyse how the US antitrust agencies evaluate the competitive import of open-source software under the Horizontal Merger Guidelines,[5] and how the EC may assess the role of open-source competition differently.

Overview of open-source software

What is open-source software?

The short answer is that open-source software is "free". Here, "free" has two meanings. First, open-source software is available without payment to the source-code owner. This does not mean the open-source software is costless to the user, as there usually are expenses associated with implementing and maintaining the software and internal costs borne by users in the form of IT personnel hired to administer the open-source application. Of course, proprietary software (especially its source code) is not free either; users often pay substantial sums to use the software. Secondly, and perhaps more importantly, the source code in open-source software is not secret, but rather freely available to anyone to use, modify, or incorporate into a new derivative product. Conversely, the source code of proprietary

[1] BusinessWire, *Open Source Software Market Accelerated by Economy and Increased Acceptance From Enterprise Buyers, IDC Finds* (July 29, 2009), available at *http://www.businesswire.com/portal/site/home/permalink/?ndmViewId=news_view&newsId=20090729005107&newsLang=en* [Accessed May 5, 2011] (citing Michael Fauscette, *Worldwide Open Source Software 2009–2013 Forecast* (Int'l Data Group, 2009)).

[2] BusinessWire, *Open Source Software Market Accelerated by Economy and Increased Acceptance From Enterprise Buyers, IDC Finds*, July 29, 2009.

[3] Netcraft Ltd, *Operating System Software Used at Sites in All Locations January 2009* (January 2009), available at *http://news.netcraft.com/ssl-sample-report/CMatch/oscnt_all* [Accessed May 17, 2011].

[4] Netcraft Ltd, *February 2010 Web Server Survey* (February 3, 2010), available at *http://news.netcraft.com/archives/web_server_survey.html* [Accessed May 5, 2011].

[5] US Department of Justice & Fed. Trade Comm'n, *Horizontal Merger Guidelines* (rev. 1997), reprinted in 4 Trade Reg. Rep. (CCH), paras 13,104, available at *http://www.justice.gov/atr/public/guidelines/horiz_book/hmg1.html* [Accessed May 5, 2011] (hereinafter "*Horizontal Merger Guidelines*" or "*Guidelines*").

software is a closely-guarded secret not generally available for manipulation by licensees. However, "open-source" does not always mean that the code is in the public domain; invariably, somebody owns the code and related intellectual property, e.g. a trade mark. Rather, as with any other software intellectual property, the source code is made available pursuant to a licence arrangement under which the owner of the code defines how the code may be used, modified and redistributed by licensees. Thus, open-source software is "free", but sometimes has (significant) strings attached.

The Open Source Initiative ("OSI") lists 10 criteria an open-source software licence should meet, including free distribution, a well-publicised means of obtaining un-obfuscated source code, the ability of licensees to make modifications and create derived works together with the ability to redistribute the modifications and derived works on the same terms as the licence to the original code, non-discrimination principles, and automatic licensing upon redistribution.[6] In addition, numerous open-source software licensing schemes have been deemed to be acceptable "open-source" licences by organisations like OSI and the Freedom Software Foundation ("FSF"). Those schemes combine commercial considerations and the ideological principles of the open-source community in varying proportions.

Open-source licensing schemes

A key aspect of these licences is the extent to which (if at all) the licence permits redistribution of open-source code in proprietary products. For example, one of the most widely-used open-source licensing schemes, the GNU General Public Licence ("GPL"),[7] drafted by the FSF, requires that any redistribution of the source code, including any modifications to and derivative works from the original code, be subject to the same open-source licensing terms as the licence to the original code. Open-source licences with this attribute are sometimes called "copyleft" licences. The GPL and other copyleft licences prevent a licensee from incorporating code into a product that is redistributed for a fee.[8] Moreover, they have the so-called "viral" effect of requiring any licensee's proprietary code used in creating a derivative work to be open-sourced under the terms of the copyleft licence if redistributed. The same is true of the Mozilla Public Licence, another open-source licensing scheme that has a viral effect on derivative products. While these types of open-source licences are designed to preserve the original source code's openness and to extend that openness to any improvements and modifications (including derivative works) made by any licensee, they are sometimes characterised as "restrictive" in the sense that a licensee may not create a proprietary, derivative product offered for a fee.

On the other hand, other open-source licensing models, such as the Apache Licence 2.0 drafted by the Apache Software Foundation and the BSD licences authored by the Regents of the University of California, permit licensees to incorporate software released under these licences into proprietary products. For example, Apple's highly proprietary OS X has its roots in open-source code licensed under a BSD-style licence. Google licenses the Android platform for mobile devices under Apache Licence 2.0.[9] Because licences like the BSD do not prevent a licensee from closing off access to the modifications it makes to open-source (which can have the practical effect of nullifying the value of the original open-source code), they are sometimes characterised as "permissive," since they permit a licensee to charge for the modified code and keep the modifications secret.

Thus, the differing licensing schemes affect the ability of licensees to monetise products that are based on open-source code. If the licensee will not be redistributing the open-source code (either on a standalone basis or as part of a new product), then the restrictions imposed by GPL-style licences will not have much effect, since the "restrictions" on commercialisation apply only if the code is "resold" by the licensee. For example, web-based enterprises have used GPL-style open-source products (e.g. databases and web servers) extensively without adverse economic consequences because these enterprises do not redistribute any source code when consumers interact with their websites. Likewise, developers can publish commercial software that interfaces with open-source software, such as commercial application software that operates on Linux. But, if the licensee is a software OEM that would like to incorporate open-source code into the products it creates for sale, the GPL code could not be used because it would "infect" the entire derivative product, preventing the OEM from charging a fee and requiring the OEM to open-source on GPL terms any of its own proprietary code comingled with the licensed open-source code.

Whether source code is licensed under the GPL or the BSD, the owner of the code does not receive royalties from its open-source licensees. Firms that own software that they have open-sourced under copyleft licences such as the GPL (as well as licensees of copyleft code) stay in business by selling support, services, and ancillary products.

Firms using open-source software may lower their development costs by relying on a development community broader than their own programmers. For example, although Linux is a open-source product, many companies, most notably Red Hat Inc, earn hundreds of

[6] Open Source Initiative, "The Open Source Definition", available at *http://www.opensource.org/docs/osd* [Accessed May 5, 2011].

[7] The text of the GPL may be found at *http://www.gnu.org/licenses/gpl.txt* [Accessed May 5, 2011].

[8] Monty Program Ab, *Request to Protect Disruptive Innovation in the Overall Information Technology Sector* (2009), p.21, available at *ftp://ftp.askmonty.org/secret/COMP_M.5529_Req_to_protect_disruptive_innovation.pdf* [Accessed May 5, 2011].

[9] Ryan Paul, "Why Google chose the Apache Software License over GPLv2 for Android" in *Ars Technica*, November 6, 2007, available at *http://arstechnica.com/old/content/2007/11/why-google-chose-the-apache-software-license-over-gplv2.ars* [Accessed May 5, 2011].

millions of dollars selling support and maintenance services. In addition, firms that own source code available under the GPL or similar licences can offer the same code for a fee under a proprietary licence. This "dual licensing" model, pioneered by MySQL, gives the owner of the code the means to earn revenues from licensees (like the software OEM noted above) who would prefer to pay a royalty rather than be subject to the requirements of an open-source licence, while providing an open-source licence to those who are not affected by restrictions on redistribution. It is important to note that the ability to "dual license" is available only to the owner of the code.[10] Absent a specific licence granting such rights, an open-source licensee would not have the right to offer the source code, including its modifications to that source code, on a dual-licence basis.

Regardless of the licence form employed, once code has been made available on an open-source basis, it cannot be "taken back" by the owner. At most, an owner can decide to make its new developments proprietary, but it cannot prevent others from using or modifying—or making derivative works from—code already available under an open-source licence. Thus, a key attribute of open-source code is that it persists in the marketplace, available for use and modification by anyone, with no legal ability on the part of the owner to exclude anyone from using or distributing the code. In other words, the owner cannot later exclude firms that have developed or will develop products derived from the owner's open-sourced code, so long as the redistribution requirements of the open-source licence are observed.

The open-source community and the creation of derivative products

An important consequence of the ability of anyone to use and modify open-source software is that the "community" of users and developers plays an important role in the ongoing development of the code. This development includes everything from bug fixes to the contribution of new features and functionality. Although anyone can be a member of an open-source community, invariably, some members are more committed to and have more expertise with developing a given open-source project. Thus, with the development freedom made possible by open-source comes the possibility of confusion if no organised structure evolves to make sense of the community's activities. Some institution has to determine which bugs to fix and which new features to incorporate. Usually, a hierarchy develops in which a core group of developers organises which (and when) new features and functionality will be included in updated versions of an "official" release of the software. The hope is that a balance will emerge between the freedom of the community to contribute to the development of the code and the order necessary to produce a coherent product that users and developers can recognise as "standard issue".

In many cases, a commercial entity takes charge of developing the code and provides a framework for the community's development activities. For example, EnterpriseDB, a firm that markets and supports an open-source database product called PostgreSQL, provides resources and direction for PostgreSQL development.[11] At the same time, an organised community of developers actively develops PostgreSQL.[12] Depending on how development resources are allocated, the most significant concentration of developers for an open-source product may be employed by the significant "commercial" vendors of the product (that is, vendors that sell support and services for the otherwise free software).

Invariably, disputes arise between members of the community and the course charted by the steward. Sometimes splinter groups organise around a vision for an open-source product that differs from the vision of the main development group. When this happens, a new "fork" of the open-source product emerges, with its own decision-making group and followers. Thus, open-source licensing offers the possibility for any number of competing products to emerge from an original embodiment of open-source code. While there is no legal impediment to prohibit many forks emerging from a single branch, there are practical limits to how many forks can be successful. For example, there may not be enough developers to sustain multiple forks (e.g. where most of the significant developers work at the commercial steward of the product), even though a small group in the community would like to see its vision adopted.[13]

Another limitation on forks might be that a fork's sponsors would not have the right to dual license the forked version of the product, thereby limiting their ability to tap into what may be a source of revenue necessary to sustain its existence. Or, the consumers of an open-source product may become confused when an incoherent proliferation of competing iterations of an open-source product emerge. Such confusion might in turn degrade the market acceptance of some or all of the various iterations of the product.

Thus, while open-source products provide opportunities for entry and limit the ability of even the owner of the code to remove derivative products from the market, practical commercial considerations must be taken into account when assessing the significance of open-source products in a merger analysis. As described below, the ability of the owners and licensees of open-source software to generate revenues and for community members to sponsor a successful fork are important aspects of the antitrust analysis, both on the demand side

[10] The owner of the code is typically either a developer (e.g. IBM), or a foundation created to manage the IP (e.g. the Mozilla Foundation).

[11] EnterpriseDB, *http://www.enterprisedb.com* [Accessed May 5, 2011].

[12] PostgreSQL, *http://www.postgresql.org/community* [Accessed May 5, 2011].

[13] Both proprietary and open-source software developers face the same trade-off between developing general products that can serve many customers (less consumer value but at lower cost) and specialised products that more closely meet the needs of smaller groups of customers (more consumer value but at higher cost).

(e.g. does a GPL product compete for OEM customers?) and supply side (e.g. is entry likely when the open-source licence in effect limits a firm's commercial viability?).

Why is open-source software important in technology markets?

Open-source software is important in facilitating product improvement and competition. It provides a platform for software developers and users to diagnose and remedy problems with current software as well as create and adopt new open-source solutions to accommodate changing needs. Because open-source software (unlike commercially-available proprietary software) allows access to the source code, users can improve it, fix bugs, and produce new software products derived from the open-source code. The increasing availability and adoption of open-source software is having an impact in several significant software markets and can pose real competitive threats to commercial proprietary vendors, particularly where a significant support community emerges to develop the code. As early as 2003 Microsoft recognised the potential threat that open-source software posed to its OS monopoly. CEO Steve Ballmer noted:

> "Noncommercial software products in general, and Linux in particular, present a competitive challenge for us and our entire industry, and they require our concentrated focus and attention."[14]

A recent Seventh Circuit opinion by Judge Easterbrook lauded the benefits of open-source software for innovation in the market for operating systems. There, the court described how the general public licence to Linux, which ensures the availability of any Linux-based source code, "precludes … reduction in output" and "facilitates production of new derivative works".[15]

Low development costs

The production of complex software products requires substantial development costs. As such, software production requires significant up-front investments in product development while short-term variable costs of production are virtually zero.[16] Thus, market segments for complex software products tend toward just a few suppliers at any given time.[17] Consequently, open-source software projects can pose a competitive threat to proprietary software products. In particular, development costs may be diffused across a development community rather than borne by a single development company, and substantially reduced when entering via a fork.[18] For example, among server operating systems, Linux has thrived in large part because the costs for developing the platform were largely carried by community programmers contributing the fruits of their programming efforts to the Linux Foundation.

However, while open-source development costs may be lower, an open-source developer/owner faces two other challenges. First, the return on investment may also be lower because the developer cannot charge a licence fee for the software. Actual investment returns typically depend on the complementary products the developer offers. Secondly, the open-source developer may face a co-ordination cost in developing a critical mass in the user and development community. Because the developer/owner typically relies on community development, at least in part, the developer must facilitate community formation and attempt to minimise free riding by community users (i.e. users who do not contribute back to the community's development efforts). Without sufficient community scale for community development and contribution, the software will not succeed in either attracting enough users or continuing to innovate.

Open-source as a strategy to increase scale

Open-source can also serve as a viable competitive strategy where proprietary alternatives have failed or floundered. Take, for example, the Integrated Development Environment ("IDE"), which is a software application that computer programmers use together with a programming language to develop new software. In 2001, IBM donated $40 million of software code to found the Eclipse Open Source project, which included IBM's proprietary efforts in this market, called VisualAge. IBM developed VisualAge to compete head-to-head against Microsoft's Visual Studio, which was the dominant IDE at the time. The Open Source Eclipse project was extremely successful in attracting a development community to support it.[19] Given that the software was free, it forced Microsoft to react. Microsoft offered Visual Studio Shell for free and substantially lowered the price of its Visual Studio suite. Why did IBM decide to open-source its IDE? In part, IBM must have concluded that its efforts to challenge Microsoft's Visual Studio dominance would be less successful as a proprietary offering. By open sourcing the project, IBM was able to attract a substantial number of application developers to the open platform, which helped build a substantial ecosystem for the project. This success provided the IDE with the scale needed to compete against Microsoft. At the same time, IBM was able to sell services and more

[14] David Becker, "Ballmer Memo Targets Linux" in *CNET*, June 4, 2003, available at *http://news.cnet.com/Ballmer-memo-targets-Linux/2100-1016_3-1013124.html* [Accessed May 5, 2011].

[15] Wallace v Int'l Bus. Machs., 467 F.3d 1104 at 1107 (7th Cir. 2006).

[16] See, e.g. Carl Shapiro and Michael L. Katz, "Antitrust in Software Markets" in *Competition, Innovation, and the Microsoft Monopoly: Antitrust in the Digital Marketplace* (Kluwer Academic Publishers, 1999); Jurgen Bitzer, "Commercial Versus Open Source Software: the Role of Product Heterogeneity in Competition" (2004) 28 Econ. Sys. 369, 370.

[17] Shapiro and Katz, "Antitrust in Software Markets" in *Competition, Innovation, and the Microsoft Monopoly*, 1999.

[18] Shapiro and Katz, "Antitrust in Software Markets" in *Competition, Innovation, and the Microsoft Monopoly*, 1999.

[19] The Open Source Eclipse Project was formed in November 2001 by industry leaders Borland, IBM, MERANT, QNX Software Systems, Rational Software, Red Hat, SuSE, TogetherSoft, and Webgain. By 2003, it had grown to over 80 members: The Eclipse Foundation, "History of Eclipse", available at *http://www.eclipse.org/org/#history* [Accessed May 5, 2011].

advanced, proprietary software applications to work with the open-source IDE, creating a viable business model centered on an open-source strategy.[20]

These examples demonstrate that open-source can challenge market incumbents. At the same time, countless examples demonstrate that where open-source does not gain significant market traction, it represents no more than a hobbyist's alternative to enterprise-grade software products. The code could be insufficiently developed; the developer community supporting the project could be too small; the market segment that the open-source product tried to serve could demand substantial post-installation service, support and customisation; or insufficient third-party infrastructure could exist to make the open-source platform a competitive threat. In such instances, the existence of an open-source alternative may not represent any competitive threat to market incumbents.

Commercial viability of code

There are countless open-source projects; indeed, there is likely an open-source project equivalent in every software market. But that does not mean that open -source is a competitive alternative to proprietary software solutions in every instance. Some open-source projects relate to university research projects and are designed to target very specific (and oftentimes non-commercial) areas of research. Although freely available for the market to use, expand, or improve, in many instances, the resulting code has little commercial viability, since it would take as much time to make the code commercially viable for use in an enterprise (e.g. sufficiently scalable, fast enough to perform the required functions, secure enough to deploy in a network, etc.) as it would to develop it from scratch.

Availability of support and service

Even if the code itself is, or with relative ease could become, commercially competitive with proprietary solutions, that does not mean that the open-source project necessarily will gain broad market acceptance. Take, for example, a relatively simple software deployment. The code, in such an instance, may not be the differentiator in the market. It could be, for example, an application that works in conjunction with complex hardware, and the service of the software is provided by that hardware vendor. An open-source alternative may develop that is as strong as, or even marginally stronger than, the proprietary code offered by the hardware vendor. But, if that code requires post-installation support and service by the hardware (or other third-party) vendor, the availability of an open-source alternative will not ensure that the product is a commercially feasible alternative for all customers. If, in the absence of third-party support, an IT department is required to service and support the software post-installation, then corporate users will evaluate the total cost of ownership ("TCO") of the software, accounting for the costs that will be borne internally by the company. In the case of individual users, the open-source alternative may not develop into a viable alternative to the proprietary application unless a third party sponsors the software to service customers' post-installation needs. The Firefox internet-browser is an example of a successful open-source software product that is widely used by individuals and does not require. the same level of external support as a product like Apache, which is only used by an "expert" community. Without sufficient demand for the software or complementary products, and consequently sufficient revenue potential, even the availability of strong open-source software will not provide an adequate constraint on the proprietary alternatives in the market.[21]

Open-source software in merger analysis

Antitrust authorities face unique challenges in analysing markets where open-source software has or may become a competitive threat. In the next section, we address some of these challenges, particularly with respect to market definition, market entry, and situations where open-source software is part of the merger or acquisition transaction.[22] We also discuss, in the context of the Sun/Oracle transaction, how open-source competition may be assessed differently in merger review in the US versus merger review at the European Commission.

At first blush, the availability of open-source software suggests that entry barriers are low and that a new product could quickly come to market in response to a merger that otherwise might lessen competition. But the open-source nature of a software product is just one aspect of the question of whether entry is viable. Other considerations are often equally important. For example, do products created from open-source code have the functionality and support to compete with the performance capabilities of the products offered by the merging parties? Are there sufficiently focused development efforts to make sure an open-source product incorporates improvements and new features in a way that satisfies the requirements of demanding commercial users? Is there

[20] Martin LaMonica, "New Age Dawns at Eclipse" in *CNET*, January 28, 2004, available at *http://news.cnet.com/New-age-dawns-at-Eclipse/2100-7344_3-5149102.html?tag=mncol* [Accessed May 5, 2011] (noting industry perception that IBM has used Eclipse primarily to push sales of its own Java software); see also Martin LaMonica, "Eclipse to Split from IBM" in *CNET*, January 20, 2004, available at *http://news.cnet.com/Eclipse-to-split-from-IBM/2100-7344_3-5143421.html?tag=mncol* [Accessed May 5, 2011].

[21] For an illustrative discussion of the benefits of community adoption in the case of Linux, see, e.g. Carl Shapiro and Hal Varian, *Linux Adoption in the Public Sector: An Economic Analysis* (December 1, 2003), available at *http://people.ischool.berkeley.edu/~hal/Papers/2004/linux-adoption-in-the-public-sector.pdf* [Accessed May 11, 2010]. New revenue models, such as revenue derived from contextual advertising associated with hosted software use continue to expand opportunities for return on open-source software development.

[22] The agencies have faced similar challenges in addressing in-house development of software solutions by corporate users in competition with third-party suppliers. For a general discussion of the consideration of software mergers by the agencies, see, e.g. Carl Shapiro and Michael L. Katz, "Antitrust in Software Markets" in *Competition, Innovation, and the Microsoft Monopoly: Antitrust in the Digital Marketplace* (Kluwer Academic Publishers, 1999).

adequate service and support for an open-source product to drive widespread market acceptance by commercial users? In short, the lower intellectual-property barriers made possible by open-source software are but one aspect of the entry analysis with which antitrust authorities must wrestle.

Open-source also raises special questions on the demand side of merger analysis. These questions fall into two main categories. First, as in all merger analyses, the authorities must determine whether the features and functionality of the open-source product make it a sufficiently close substitute for proprietary software. This question is really no different from questions asked in any merger case. Secondly, putting to one side the functionality of the product, the authorities must ask whether something about the open-source nature of the product in question makes it more or less attractive to users in comparison to the proprietary alternatives. For example, open-source licensing terms may make an otherwise perfectly substitutable open-source product unattractive to a user who does not want its proprietary code to be made subject to the commercial restrictions imposed under a GPL licence. On the other hand, open-source software is “free” (i.e. there are no licence fees and the source code can be modified by the user) and this freedom may make an open-source product a close substitute to proprietary software that has more features and functionality. A case in point is Oracle’s acquisition of Sun Microsystems, in which the EC evaluated whether the open-source nature of Sun’s MySQL database product made it an especially important competitive constraint on Oracle’s very expensive, proprietary database.[23]

Open-source software as a consideration in market definition and market concentration

Open-source software can broaden the product market and serve as a competitive constraint on the merging parties, but this may not always be the case. Before the market significance of open-source may be assessed, open-source software must first be identified. An open-source product may be distributed through channels other than typical retail channels that may not be captured in industry reports. Because such reports usually include only market share attributed to revenues from the sale of commercial products, free products may not be captured. The source code may be available from any number of sources, making it difficult to identify all the “suppliers” of the product. It is also often difficult to measure downloads of the code or other indicia of its use by consumers. In other words, assessing the role of open-source products as potential competitors in the market is likely more challenging than identifying competitors of proprietary systems.

Beyond measuring market share, the authorities face the potentially challenging determination of whether open-source presents a close enough substitute to be considered a viable alternative in the relevant market. Here, analysis of the product’s functionality is essential. For example, Open Office is a free open-source office software suite for word processing, spreadsheets, presentations, graphics, and databases. However, just because Open Office is an alternative to Microsoft Office does not necessarily mean that it represents a true competitive threat. As the Seventh Circuit succinctly noted in *Wallace*, “[p]eople willingly pay for quality software even when they can get free (but imperfect) substitutes”.[24] Moreover, open-source must be sustained by an infrastructure of maintenance and support to be an effective competitor. If not, large enterprises may be reluctant to invest in software serving core functions such as word processing—even if doing so lowers costs—unless they can be confident that bugs will be fixed, necessary upgrades will be available and timely, and any other issues or problems with the software will be easily serviceable.

Another important issue affecting the commercial viability of open-source—and therefore its potential as a substitute—is how new features get incorporated into an open-source product. If additions to the software are unregulated and unmaintained by an organised community of users, then the software may very well become “bloated”, affecting its performance. Open-source products that are splintered into multiple forks by the development community may not have a sufficiently clear identity to achieve wide adoption among commercial users. As noted above, businesses are unlikely to sacrifice product performance or a coherent development path for the low initial costs of licensing open-source software. Moreover, free software that is maintained by a community of users may have inherent security limitations associated with it—with a “recipe” available to the world, an open-source product may be particularly susceptible to hacks, which would be especially dangerous if installed in certain components of an enterprise.

Finally, if the authorities conclude that open-source software is a viable substitute for proprietary enterprise-ready software and, therefore, part of the relevant market, they must attempt to quantify what portion of the market open-source represents. Since open-source software does not generate revenue from the sale of its source code, conducting traditional market-share analysis based on revenue may be extremely difficult or even impossible. Though a company that sells derivative products of open-source software earns income from those products, it is unlikely that this type of revenue stream could be used in comparison with proprietary software sales to generate a useful market share analysis.[25] Accordingly, the best way to calculate market share in

[23] Press release, European Commission, *Mergers: Commission Opens In-depth Investigation into Proposed Takeover of Sun Microsystems by Oracle* (September 3, 2009), available at *http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/1271* [Accessed May 5, 2011].

[24] *Wallace*, 467 F.3d at 1107.

[25] Similarly problematic would be any type of market-share calculation based on capacity, since the potential capacity of licenses for an open-source product is unbounded.

markets with open-source competition may be based on a measure of volume—e.g. number of licences installed, number of processors, or number of users—or on a measure of total cost of ownership.[26] It will likely be difficult to track how many licences are "in use" by customers in the market, and the number of instances that the code has been downloaded may not represent its market penetration. For example, it is impossible to know whether any of the free downloads actually were deployed by the individual or enterprise that downloaded it, and the marginal cost of downloading and potentially experimenting but not actively using the product is low. To the extent that it is impossible to calculate market shares based on the number of users due to the unavailability of data, authorities may look at anecdotal evidence that shows customers using open-source as leverage in negotiations.[27] While this may not provide exact figures for a market-share analysis, it can serve as a proxy for the agencies in determining whether a critical number of users license open-source software such that it will likely constrain a small but significant non-transitory price increase post-acquisition.[28]

Open-source facilitating entry

Market entry is also likely to be a significant issue in markets where open-source software represents a potential competitive threat. For example, under the Horizontal Merger Guidelines,[29] entry can be a mitigating factor if it is, "timely, likely, and sufficient in its magnitude, character and scope to deter or counteract the competitive effects of concern".[30] In such a market, entry can be facilitated by forking or building upon existing open-source code. While in many instances, firms relying upon open-source software are likely to be in a position to enter a market in a timely manner (because the code is freely available to the market), issues related to the *likelihood* and *sufficiency* of such entry are more complex for antitrust enforcers.

Timely

Entry in software markets may be more likely to be timely than entry in other markets as a result of open-source. Availability of the basic source code could significantly reduce the expenditure of sunk costs required to enter. Because significant investment in the code—both in terms of money and time—has already been made, companies looking to "fork" or modify the open-source software for commercial purposes can do so more quickly than firms in industries where new entrants must start from scratch. The bigger challenge may involve forming a community around a fork, and developing the institutions to manage the fork.

Likely

What becomes trickier with open-source software is determining whether entry will actually occur and whether it will be sufficient to constrain the exercise of market power. Under the Guidelines, minimum viable scale is the smallest average annual level of sales that the committed entrant must achieve for profitability at premerger prices. Entry is likely if the minimum viable scale is less than the likely sales opportunity available to an entrant. While entry costs for an open-source competitor may be low, corresponding revenues may be low as well, increasing the minimum volume necessary for a new entrant to become profitable. Simply put, third parties with the ability to support and maintain open-source software may not find the potential revenue-stream enticing enough to enter and compete with proprietary software vendors.

As another example, the potential sales opportunity for a new entrant may be limited due to the presence of network effects. There is value to the consumer, particularly with respect to software where standardisation is crucial, in using the same product as everyone else. This is particularly the case in software, such as document processing software, where the exchange of information is essential to the product's usefulness.[31] If the network effects are strong enough, sales may not be diverted to a new entrant even if its product is functionally comparable and priced at a significant discount to the existing proprietary product.

Sufficiency

Finally, the sufficiency of competition offered by the new entrant may be questionable. The entrant's product must be close enough, in character and scope, to the existing proprietary software that the merged firm will suffer sales losses to the open-source competitor.[32] As discussed earlier, in order to entice enterprise customers to switch

[26] Gustavo Olivieri and Laura Marchegiani, "Open Source Software and Technological Innovation: Competitive Issues" (2005) 95 *Rivista di Politica Economica* 47, 54.

[27] For example, a proprietary software vendor's win/loss data may demonstrate that it loses sales to open-source.

[28] Similarly, as may be necessary for assessing market definition and competitive effects, calculation of margins associated with distribution of open-source software can be challenging, particularly when the value derived from that distribution is related to sales of complementary products facilitated by use of the open-source software.

[29] *Horizontal Merger Guidelines* para.3.0. While the Guidelines apply to the US antitrust agencies only, their principles on the viability of new entry are universal, and therefore applicable in the context of any merger analysis.

[30] Notably, the US antitrust agencies are currently in the process of considering significant revisions to the *Horizontal Merger Guidelines*. Particularly interesting for the present analysis are the proposed revisions related to entry. For example, under the revised guidelines, "timeliness" is now a flexible, fluid standard depending on whether entry is perceived to be fast enough to deter any anticompetitive effects resulting from the transaction. In addition, under the proposal, the agencies will give substantial weight to the existence of an "an actual history of entry" in determining the likelihood of entry. Both of these factors have the potential effect of making the demonstration of "entry," whether or not it involves issues of open-source competition, more difficult. US Department of Justice & Fed. Trade Comm'n, *Horizontal Merger Guidelines for Public Comment* (April 20, 2010), available at *http://www.ftc.gov/os/2010/04/100420hmg.pdf* [Accessed May 5, 2011].

[31] Alexandre Gaudeul, *Competition Between Open-source and Proprietary Software: The (LA) TEX Case Study* (EconWPA 2004), available at *http://129.3.20.41/eps/io/papers/0409/0409007.pdf* [Accessed May 5, 2011].

[32] *Horizontal Merger Guidelines* para.3.4.

to open-source, the software must be functionally comparable to proprietary software, offering comparable features, interface, and product support and maintenance.

Disruptive market dynamics

Open-source software may also be relevant to merger analysis because it has the potential to disrupt stagnated software markets characterised by little competition and high entry-barriers. This disruptive innovation creates a new market dynamic focused on providing useful, affordable, and more customer-friendly alternatives. For example, in some markets, "low-end" (i.e. less functionality and/or less complexity than is generally required for mission critical applications) open-source disruptors enter the market by meeting low-end needs purposely neglected by incumbents focused on high-margin segments. Then, the open-source products move up-market and become increasingly consequential to entrenched firms with significant high-end market share. This sort of market dynamic is significant: if successful with its strategy, the disruptive open-source software product either will act as a significant competitive constraint on the entrenched market incumbent(s) or will become an acquisition target for the entrenched incumbent that is seeking to displace the firm that is steadily climbing up the curve and penetrating more lucrative, higher-end market segments.[33]

Open-source as part of the transaction

Another interesting competition issue posed by open-source software is the appropriate way to analyse a merger where open-source software is combined with proprietary software, and how open-source competition may be viewed differently by antitrust authorities in the United States and Europe. One view is that where open-source is a viable competitor in a relevant market, it represents a unique form of competition that cannot be eliminated or significantly diminished through merger. In its review of the acquisition of Sun by Oracle, the EC launched an in-depth investigation into Oracle's acquisition of Sun amid concerns that the combined entity could have the ability to create higher prices or reduced choice for European consumers.[34] The EC queried, in its initial objections to the merger, whether harm would flow from, "the world's leading proprietary database company … tak[ing] over the world's leading open source database company".[35] Ultimately—after an extended review of the deal—the EC cleared the acquisition.[36] In doing so, the EC found it significant that other actors in the database market could, if necessary, fill the void in competition left open by MySQL if the viability of MySQL were adversely affected by the merger.

First, the EC found that another open-source database, PostgreSQL, offered a credible alternative to MySQL.[37] While PostgreSQL's success thus far had been limited (in part because of the overwhelming success of MySQL), PostgreSQL had the ability to replace the "competitive force … exerted by MySQL on the database market", particularly under circumstances in which Oracle restricted the use or functionality of MySQL post-transaction.[38] Secondly, because MySQL is open-source, developers could create derivative versions of the database by forking it, and these derivative products could serve as a competitive constraint on Oracle.[39] The Commission noted that several forks of MySQL, such as MariaDB, Percona, and Drizzle, already existed and additional forks were likely to develop and enter the market if Oracle affected the performance of MySQL following the merger.[40] Finally, the EC noted Oracle's public pledges to release future versions of MySQL under the GPL and its steps, "to allow third parties to continue to develop storage engines to be integrated with MySQL and to extend the functionality of MySQL".[41] As a result of these public announcements, the EC concluded that Oracle was, "unlikely to have the ability or incentive to eliminate MySQL after the merger".[42]

Despite clearing the transaction, the EC appeared to take the position that open-source competition may represent a unique form of competition to proprietary versions of software in the same market. In the case of MySQL, it found that the competitive constraint exerted by Sun was insufficiently unique such that its acquisition of Oracle would diminish competition. Nonetheless, the EC tipped its hand and exposed its belief that in some instances open-source competition is "special", and transactions that eliminate a substantial supporter of open-source deserve closer scrutiny.

The EC's view that open-source competition is a unique competitive constraint is best summarised in para.661 of the Commission's decision:

> "Due to its business model and open-source nature MySQL appears to exert a specific competitive constraint which seems to be different from the constraint that can be exerted by proprietary database vendors. After the merger Oracle will of course

[33] Paula Rooney, "Is the Open Source IPO a Pipe Dream" in *ZDNet*, January 29, 2008, available at *http://blogs.zdnet.com/open-source/?p=1947* [Accessed May 5, 2011] ("M&A activity does not threaten the integrity of open source … [r]ather it simply reflects how disruptive technologies alter the business models of incumbents.").

[34] Press release, European Commission, *Mergers: Commission Opens In-depth Investigation into Proposed Takeover of Sun Microsystems by Oracle* (September 3, 2009), available at *http://europa.eu/rapid/pressReleasesAction.do?reference=IP/09/1271* [Accessed May 5, 2011].

[35] Press release, European Commission, *Mergers: Commission Opens In-depth Investigation into Proposed Takeover of Sun Microsystems by Oracle*, September 3, 2009.

[36] Press release, European Commission, *Mergers: Commission Clears Oracle's Proposed Acquisition of Sun Microsystems* (January 21, 2010), available at *http://europa.eu/rapid/pressReleasesAction.do?reference=IP/10/40&format=HTML&aged=0&language=EN&guiLanguage=en* [Accessed May 5, 2011].

[37] Press release, European Commission, *Mergers: Commission Clears Oracle's Proposed Acquisition of Sun Microsystems*, January 21, 2010.

[38] Decision 142/2010 declaring a concentration to be compatible with the common market and the functioning of the EEA Agreement (COMP/M.5529—Oracle/ Sun Microsystems) [2010] OJ C142 para.677, available at *http://ec.europa.eu/competition/mergers/cases/decisions/m5529_20100121_20682_en.pdf* ("EC Decision").

[39] Decision 142/2010 paras 678–750.

[40] Decision 142/2010 paras 679, 714.

[41] Press release, European Commission, *Mergers: Commission Clears Oracle's Proposed Acquisition of Sun Microsystems* (January 21, 2010).

[42] Decision 142/2010 para.658.

> continue to face strong competition from other proprietary database vendors such IBM, Microsoft, Sybase and others whose database offerings were presented in section 1.2.3. However, to the extent that, as explained in section 4.3.4.4, MySQL also appears to potentially exert a particular competitive constraint not only on Oracle but also on other proprietary database vendors, the Commission's assessment focuses on the potential for another open-source database vendor to replace such a competitive constraint on Oracle and other proprietary database vendors."[43]

Despite "strong competition from other proprietary database vendors", the EC was clearly concerned that the "specific" and "particular" competitive constraint exerted by open-source competition would be lessened post-transaction. Accordingly, the EC's findings that PostgreSQL and forks of MySQL would be able to maintain open-source competition in the database market following the merger were critical to its overall conclusion that the transaction was unlikely to significantly impede effective competition.[44]

The DOJ's review of the transaction was more abbreviated. The Government's public comments in the case suggested that the agency evaluated the acquisition of an open-source software vendor similarly to the acquisition of any other competitor in the relevant market—whether offering an open or closed solution. Besides the fact that a large community of developers and users can support MySQL or a derivative product with or without Sun, the DOJ, in declining to investigate the deal, noted that, "there are many open-source and proprietary database competitors" and that:

> "[C]onsumer harm [from the merger] is unlikely because customers would continue to have choices from a variety of well established and widely accepted database products."[45]

It appears that, at least in that case, the DOJ concluded that open-source software, if acquired, did not deserve any special protection simply because it was free.

The Seventh Circuit in *Wallace* analysed the issue similarly. In rejecting the plaintiff's predatory pricing argument, the court found that even if competition from open-source operating systems became obsolete, there was still substantial and growing competition among proprietary operating systems:

> "Software that is not maintained and improved eventually becomes obsolete, and the lack of reward may reduce the resources devoted to maintenance and improvement of Linux and other open-source projects. If that occurs, however, then proprietary software will enter or gain market share."

The *Wallace* court did not differentiate between proprietary and open-source software, and specifically contemplated that if an open-source product were unable to compete, proprietary software would step in and fill its shoes.

The *Wallace* court approach (and that of the DOJ) makes sense. The nature of the competing product, itself, is not necessarily important or dispositive. What is important is whether the product and firm offering the product offer a substantial or unique competitive threat to the acquiring firm, regardless of whether the product is free or not. There are some unique aspects of open-source competition, and if the elimination of a competitor offering such products would allow the acquiring firm to reduce competition, then the merger is problematic. If, on the other hand, the competition from the open-source software vendor is not unique, then the merger does not raise problems, even if the acquired firm sells an open-source product.

In the context of a merger, open-source is either included in the same product market as proprietary software, or it is not. If not, there is no overlap and, therefore, no risk of increased product market concentration as a result of the new combined entity. If open-source is part of the market, then it should be analysed just as any other competitor. The fact that companies utilising open-source have a different business model—i.e. selling products complementary to the software rather than the source code itself—does not mean that they should be considered any different than proprietary software companies ultimately selling the same end product. Indeed, given some of the problems associated with open-source software—for example, it sometimes is not kept up-to-date; it serves as the foundation for "research code" rather than "commercial code" and thus is not optimised for commercial deployment; it may or may not have commercial backing—the reviewing agency must carefully consider whether such software is sufficiently positioned to exert competitive influence on the acquiring firm.

Conclusion

Open-source presents some unique competition issues for the agencies to consider in merger analysis. It may be challenging both to determine if open-source is a relevant player in the market and, if so, to quantify its market share. In addition, determining whether entry by open-source competition is likely and, therefore, a constraint on post-merger price increases can be difficult and very much dependent on the functionality of the particular software. Some qualities of open-source tend to increase the probability of entry, such as the ability of a competitor to fork existing technology, while others, such as the presence of network effects, may deter entry in certain circumstances. Finally, the treatment of

[43] Decision 142/2010 para.661.

[44] Decision 142/2010 paras 756–59.

[45] Press release, European Commission, *Mergers: Commission Opens In-depth Investigation into Proposed Takeover of Sun Microsystems by Oracle* (September 3, 2009).

open-source when this form of competition is part of the transaction is a complex issue on which antitrust enforcers have disagreed. Undoubtedly, the competitive significance of open-source software will surface many times in future transactions, making the understanding of the issues surrounding open-source critical and well worth careful observation.