The integration of the open source software in the EU competition law

Master program in international business law

Supervisor: professor Pierre Larouche

Student: Belma Cabaravdic (U1243671)

Tilburg, 31.1.2014.

Table of contents

Introduction

This paper aims to give a review of the issues arising out of the open source software in practice, in relation to the proprietary software. Taking into account its specific open nature, which is at the same time its advantage and its weak spot, I tried to present problems faced by the open source software developers and users.

The first part of the paper provides an overview of the relevant market, and the market structure concerning the desktop / PC operating system market and mobile / tablet software market. In order to explain specific issues, I have referred to different cases from US and EU case law. Also, I tried to point out the problems emanated from the case of Sun Oracle merger regarding the open source MySQL database, which was the most important part of the Commission decision and the topic that caused vigorous debate among the open source community members both before and after the merger.

Problems related to the interoperability in the context of the open source software is also an important point from the open source community standpoint. Creating a competitive software can be very difficult or even impossible without the relevant interoperability information. The EU made an important step in the field with its European Interoperability framework, however there are more topics to be discussed in the future. More specifically, I explained the issue related to Java technology in the context of the aforementioned case, and made the comparison of the anticipated consequences of the merger and the actual ones.

I have also included some of the less discussed issues concerning the open source software, such as the enforceability of FOSS licenses, the importance of the level of cooperation with the experts in field of the open source software while making the decisions concerning the open source software, and the relation between the open source software and openness

1. Market definition and the issues related to the open source software

Open source software includes some important features common with proprietary software. However, at the same time it is diametrically opposite with it on many different levels. For the sake of discussion, let us suppose that there is influence on trade among the Member states, and that we cannot claim with certainty that open source software and proprietary software really do belong to the same product market. The most important first step in identifying and determining competitive constraints upon companies is defining the relevant market. As we know, a relevant market analysis is comprised of two essential aspects: relevant product market and relevant geographic market.

1.1. Product market definition

From the judgments of Court of Justice, we see that the Court considers the market definition being a matter of interchangeability [1]. In other words, goods or products which are interchangeable consequently form part of the same product market. In the Notice on the Definition of the Relevant Market for the Purposes of EU Competition Law[2], the Commission pointed out that companies are subject to three forms of competitive constraint; demand substitutability, supply-side substitutability and potential competition, with demand-side substitutability being particularly emphasized.

Taking this into account, let us consider this aspect of product market definition in terms of the topic of discussion - open source and proprietary software. It is important to properly explain the actual distinction between the two. Not every proprietary software is completely closed source software (for example there is OSX from Apple, which is a proprietary software, however it contains pieces of the open source software (the kernel). There are some other examples, such as Fedora, CentOS, openSUSE etc.), therefore it would be more accurate to use "closed source software" instead of proprietary software. Furthermore, open source software does not mean it is

[1] For example in case United Brands Company and United Brands Continentaal BV v Commission of the European Communities, Case 27/76

[2] Commission Notice on the definition of relevant market for the purposes of Community competition law, Official Journal C 372 , 09/12/1997 P. 0005 – 0013 (http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31997Y1209(01):EN:HTML , accessed on 26.12.2013.)

free of charge.[3] This question is somewhat complicated. They do and, at the same time, they actually do not. The thing is, there is more to this distinction than was being said in different decisions made by the Commission. One of the major reasons why software users and/or developers use the open source software as opposed to the closed source software boils down to the fact that open source software provides for them the freedom of choice. As software is a specific product, it cannot be precisely compared to other products.

However, software developers who prefer (and use) the open source software can use, change and customize the software so that it completely fits their needs and wishes. Unlike the closed source software, OS software enables them to have control over the software and not the other way round. Closed source software compels the users to be controlled by the software. Although it does confer to them certain rights, in the exchange for renouncing some other rights (such as the right to access the source code and to change it), it compels them to accept specific terms of use. Users agreed upon these terms, however in some instances it even means the software can spy on users, even though this means violating users' privacy. Metaphorically speaking, one could compare the purchase of closed source software to the purchase of a flat under conditions that user cannot paint the walls or erect an additional wall if need be, or change the doors and windows.[4] Therefore, the user is limited and deprived of the rights that he or she should normally be able to exercise.

As the software developers are trying to design components with high cohesion and low coupling, freedom is of great importance for them. In case they use closed source software, they are unable to choose components while building certain systems, and they certainly cannot replace those components with better alternatives at a later date (e.g. should a person or a company behind a specific software component choose to stop further production or discard support for the component). Therefore, in case that some open source project doesn't

[3] In fact, there is a linguistic issue as regards the word “free”, so I intentionally avoid the notion “free software”. In English language word “free” refers to both free of charge, and free as in “freedom”. Meaning the open source community wanted to assign to word “free” is free like “freedom”; in French it is called simply “logiciel libre”. Many times the word is misinterpreted and as a result, some people draw an erroneous conclusion that open source software is available at zero price. Therefore, free software is always free, and open source software may not be always free, but available. Considering many common points of free and open source software, we will consider it conditionally equal (but only conditionally).

[4] I contacted several open source developers during the research for this paper, and this is the comparison one of them used.

fit the needs of a developer, he may contribute and/ or help others to come up with a solution more quickly. The result of this is that the OS software cannot "die", for it can always be forked[5] into something new.

For purpose of identifying interchangeability of OS software for its users, I conducted a small scale research regarding this (via social network)[6]. 187 users participated in the research. The sample is not representative as my research covered a small part of the community of open source software users, and did not include any of the proprietary software users whatsoever. The research questions were: "Is open source software substitutable for you?" and "Would you consider replacing OS software with proprietary (closed source) software?". Concerning the first question, 84% of users (157 users) said they would not use proprietary software instead of OS software. The rest of users answered that they may be able to use proprietary software instead, but only after examining all other possible options.

Regarding the second question, 74% of users answered that they would not consider replacing their OS software with proprietary software. Around 16% said they would do it only in extraordinary circumstances, such as for security reasons, and 10% said they would do it only in case the proprietary software offers considerably better solution. Should the large-scale research be conducted with similar objectives, it seems to me that would give somewhat different image of OS software interchangeability than the one the Commission regards as real.

Moreover, a customer lock-in (or vendor lock-in) is a strategy employed by the proprietary software vendors. In case of customer lock-in, a customer is dependent on the vendor as regards the products and services, and he / she would be exposed to considerable costs if he / she decides to switch to another vendor / product. These switching costs

[5] The most often used definition of fork is the one created by the Wikipedia. Since I have not found a generally accepted definition of the fork, I embraced this one: "In software engineering, a project fork happens when developers take a copy of source code from one software package and start independent development on it, creating a distinct piece of software. The term often implies not merely a development branch, but a split in the developer community, a form of schism. Free and open-source software is that which, by definition, may be forked from the original development team without prior permission without violating any copyright law. However, licensed forks of proprietary software (e.g. Unix) also happen."

[6] Namely Facebook

actually dissuade the customer from switching to another vendor and therefore force him or her to continue using that particular proprietary (closed source) product or service. Nevertheless, the lock-in is not an infinite one, and in exchange for it, the consumer obtains certain useful features. But some specific users, such as open source developers, find it flawed for the purposes of their specific requirements.

As regards to the SSNIP test, it would be quite impossible to conduct this type of test for the reason of different business models used by proprietary software vendors and OS software vendors. However, considering the reasons for which the OS software users choose to use the OS software, from my point of view, it is only logical to conclude that price does not affect their decision to a considerable extent. Since there are problems as regards the classic approach to the market share (amount of sales), for the reason of the specific business model adopted by the open source software distribution companies, we will give an overview of the data regarding the information on key statistics such as browser trends, search engine data and operating systems in the context of monthly usage by the users. As regards the operating systems, there are two major markets; desktop/PC operating system (laptops and personal computers) and mobile operating systems (mobile OS includes mobile and tablet OS). The data is collected from the web site www.netmarketshare.com[7]

[7] Net market share data collection methodology: "We collect data from the browsers of site visitors to our exclusive on demand network of HitsLink Analytics and SharePost clients. The network includes over 40,000 websites, and spans the globe. We 'count' unique visitors to our network sites, and only count one unique visit to each network site per day. This is part of our quality control process to prevent fraud, and ensure the most accurate portrayal of Internet usage market share. The data is compiled from approximately 160 million unique visits per month. The information published on www.netmarketshare.com is an aggregation of the data from this network of hosted website traffic statistics. In addition, we classify 430+ referral sources identified as search engines. Aggregate traffic referrals from these engines are summarized and reported monthly. The statistics for search engines include both organic and sponsored referrals. This data provides valuable insight into significant trends for internet usage. These statistics include monthly information on key statistics such as browser trends (e.g. Internet Explorer vs. Firefox market share), search engine referral data (e.g. Yahoo vs. Bing vs. Google traffic market share) and operating system share (Windows vs. Mac vs. Linux market share or even the iOS market share vs. Android) The data is made available free of charge on a monthly basis that includes monthly usage market share and trends for browsers, operating systems and search engines. An upgraded version is available that provides reports by geolocation, preview weekly data and other features.
Additional estimates about the website population:
76% participate in pay per click programs to drive traffic to their sites.
43% are commerce sites
18% are corporate sites
10% are content sites
29% classify themselves as other (includes gov, org, search engine marketers etc..)

With regard to the desktop/PC operating system market share, available data indicate that Windows (Microsoft) holds 90.66% of the market. Mac OS (Apple) holds 7.73% and Linux (the open source operating system) holds 1.61% of this market. The available data show irrefutable dominance of Microsoft on the desktop / PC OS market (although its market share has been slightly reduced from 93.70% in 2008. to 90.66% in 2013). This data does not include amount of sales of Microsoft's operating system, it depicts the statistics concerning hosted website traffic. I could not find the actual information on sales as regards the Microsoft's operating system – Windows, but the available information I could access through the search engines showed that Windows still seems to be installed on over 90% of the computers worldwide (more precisely 91.62%).[8]

According to the same method of the data collection, as regards the mobile / tablet operating system market share, the situation is significantly different. While absolutely dominant in desktop / PC software market, Microsoft holds negligible share of 0.50% (Windows Phone), while iOS (Apple) holds 55.39% of the market share. If added together, Android OS (open source) and iOS hold 85.97. The rest of the market is sliced among Java ME (Oracle) with 6.47%, Symbian (3.58%), Blackberry (2.55%), and Kindle, Windows Phone, Samsung, Bada, Windows Mobile, etc. Quite different state of affairs comparing to the desktop/PC OS market. Interestingly enough, there seems to be no issues related to interoperability on this market, probably since both of them are easily available to developers. Additionally, according IDC and Gartner 57.9% of all mobile devices worldwide in 2012 has been running the Android operating system. They are convinced that Google's Android will continue to lead the market through 2016.

Competing methodologies are not as accurate as using global analytics data. Competing methods include: Surveys or Panels: The results from a survey are based on a subset of the general internet population (those willing and able to take surveys). Also, surveys are generally not provided in all languages and for all regions. This can skew the results significantly. ISP data: While the amount of ISP data can be voluminous, ISPs are regional. So, unless the ISP data is an aggregation of all ISPs, there will be a built in regional bias to the market share reports. Toolbars or Other Tracking Components Installed on Computers: Since the components would need to be developed identically for every possible platform and language and distributed evenly across all platforms and regions, this collection method is inherently flawed." (http://www.netmarketshare.com/faq.aspx#Methodology, accessed on 11.11.2013.)

[8] This particular information I found on www.softpedia.com, but I have also noticed it on other websites that report on the operating system market share, and post the news regarding the matter (such as news.cnet.com, techpinions.com, etc.)

1.2. Geographic market definition

Geographic market refers to the area in which the product is being marketed and "where the conditions are sufficiently homogeneous for the effect of the economic power of the undertaking concerned to be able to be evaluated"[9] . Taking into consideration particularity of the field of information technology, it is relatively easy to affirm the geographic market definition. Since the software (both open source and closed source software) is considered as non-tangible goods that can be easily and quickly transferred among remote locations throughout the world, at none or considerably low cost, therefore we can conclude that the relevant geographic market is worldwide.

1.3. Competitive assessment

According to the aforementioned data, it seems that Microsoft has substantial market power, taking into account the opinion of the Commission that companies with 80% of market share are approaching "super dominance", and those with 90% of the market share, are approaching "quasi-monopoly", so the position of Microsoft, has been described by the Commission as "overwhelmingly dominant position" (single firm dominance). Apart from this, there are considerable barriers to enter the market.

Firstly, the users are used to the specific operating system, and introducing a new one would incur significant costs for them (in terms of time invested in learning about new operating system, and costs that the company would have to cover in order to train the personnel, etc.). The expenses the entering competitor would have to bear, are also considerable, and the venture altogether would carried an enormous amount of risk. In support of this claim stands the fact that there is a longtime dominance by a single company, which dates back to 1990s or even earlier. As regards the countervailing buyer power, as I already mentioned, the fact that most users are familiar with one type of the operating system, and that it has been so for quite a long time, makes it difficult to introduce a new operating system which would attract longtime users of one and the same operating system.

[9] United Brands Company and United Brands Continentaal BV v Commission of the European Communities. (http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:61976J0027:EN:HTML , accessed on 9.11.2013.)

As we know from the theory of competition law, dominance in itself is not prohibited, but the abuse thereof. It is interesting how Microsoft maintained its market position over time constituting a solid base for case law both in US and EU related to issues concerning abuse of dominance.

On the other hand, the market for mobile software and tablet is a bit more segmented, although there is a visible domination of Android operating system. Taking into account the fact that any company which would decide to enter the market for mobile / tablet OS, would be able to take advantage of the open source technology at almost no cost. That being said, the barriers to enter the market could be easily overcome and entrants would probably soon try their luck, since they do not have much to lose.

There has not been much case law as regards the competition on this market, except for the recent complaint by Microsoft, Nokia, Oracle and others concerning the embedded Google application on Android devices. It remains to be seen how this case would be developed, which would certainly provide new directions for future cases.

1.4. The problems concerning the relation between the open source software and proprietary software

Let us look at the previously mentioned issue of "refusal to supply" as a non-pricing abusive conduct of a dominant firm. In the case Sun Microsystems v. Microsoft (I took this example because I considered relevant, taking into account that Sun eventually started developing the open source projects and the issues it encountered can be easily linked to those of the open source software), Sun filed a complaint against Microsoft, contending that Microsoft has infringed Article 102 of TFEU[10] and thus has been abusing its dominant position by refusing to provide information necessary for developing a software that would be fully compatible and able to interoperate with Microsoft's PC operating systems. Microsoft has been found guilty as regards to the infringement of Article 102, reflected in refusal to supply and tying Windows Media Player.

[10] Treaty on the Functioning of the European Union

One of the most sensitive problems in this case is a definition of "interoperability information", including the ambit of its impact on Microsoft's intellectual property rights. Let us explain the definition of this and other technical terms used in the Commission decision (as its importance will be shown later on). The interoperability definition from Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs[11] has been embraced for the purpose of this decision. The Directive describes interoperability as the ability to exchange and mutually use that information, quite broad a definition. However, there is no uniform definition of interoperability that is actually provided by the IT experts and that is generally accepted in field of computer software and the law. At this point it is inevitable to induce some technical clarifications in order to explain why the remedies imposed by the Commission are not likely to prevent similar problems concerning (both proprietary and open source) software in the future.

A "protocol" has been defined by the Commission as "a set of rules of interconnection and interaction between various instances of Windows Work Group Server Operating Systems and Windows Client PC Operating Systems running on different computers in a Windows Work Group Network"[12]. A "data format" is a specific way of encoding information in order to store it and transmit. Computer storage system is able to store information in form of series of 0s and 1s, which is then converted into a structured information, according specific instructions. Application Programming Interface (API) is a set of standardized requests. Simply put, API is used as an intermediary between the two programs, where the asking program needs API to do some basic functions instead of it, such as accessing the file system. An interface can be defined as a formal description on how particular programs / software components should interact with each other.

The source code (also referred to as a source or a code) is a version of software as it is originally written (i.e., typed into a computer) by a human in plain text (i.e., human readable

[11] The Directive 91/250/EEC definition of the interoperability: "Whereas this functional interconnection and interaction is generally known as 'interoperability`; whereas such interoperability can be defined as the ability to exchange information and mutually to use the information which has been exchanged" , Council Directive 91/250/EEC on the Legal Protection of Computer Programs (http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31991L0250:EN:HTML , accessed on 13.11.2013.)

[12] T-201/04 Microsoft Corp. v. Commission of the European Communities (http://eurlex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:62004A0201:EN:HTML accessed on 13.11.2013.)

alphanumeric characters)[13]. It is translated into "binary code" (0s and 1s) which computer can understand and it is then executed by the computer. This translation is executed by another computer program called "compiler". The implementation of a protocol (an interface or a specific data format) is included in the source code, which then gives instructions to the computer program on how to act in case a specific interface is called upon by another computer program. In a specific set of interfaces, there are many ways certain computer program can implement specific operations, and they do not need to, nor have to be identical.

Therefore, it is highly questionable that two or more implementations of the same interface would be identical, save the possibility of previous agreement among the developers. Hence, there are issues of their compatibility, which can be created in a very subtle way, escaping the rules set by the law, or remedies set by the Commission for that matter. Accordingly, there are two ways to do it, namely "information hiding" and "encapsuling" methods, with a quite neat difference between the two. As disclosing an interface does not have much to do with disclosing its implementation, we can see how Microsoft's argument concerning copyright protection of its interface is weak.

"In computer science, information hiding is the principle of segregation of the design decisions in a computer program that are most likely to change, thus protecting other parts of the program from extensive modification if the design decision is changed. The protection involves providing a stable interface which protects the remainder of the program from the implementation (the details that are most likely to change)."[14]

Data formats, protocols and interfaces are specified in "standards", e.i. formal documents produced by "Standard Setting Organizations" (SSO). They may as well be provided by a particular software platform, without reference to a formal document. Their relevance is quite considerable, especially in ITC markets, which includes significant level of network effects. Hence, hiding or changing interfaces, protocols, data formats not on technical merits, but for the sole reason to compel competitors to waste time and resources to achieve and keep the compatibility with the dominant firm does not seem compatible with the principles

[13] The definition is adopted from http://www.linfo.org/source_code.html (accessed on 13.11.2013.)
[14] www.princeton.edu (accessed on 13.11.2013.)

of fair competition. Strategy known as “embrace, extend and extinguish” used by Microsoft is also a subtle way to create hurdles for the competition.

Firstly, they (dominant firm) embrace the software based on the clearly defined open standards, and make it fully interoperable with Microsoft’s software. At this point, the software is compatible with every available program, and users can freely choose among the programs they want to use. In this period, the dominant firm is actively advertising the program trying to attract as many users as possible, inducing them to accept its implementation. Afterwards, the dominant firm extends the software including proprietary features which are useful but not standardized. These features are not compatible with the other available software programs, so the communication with other programs is being impeded. As they lure more users into using the program, they continue changing the standards to the extent that it becomes impossible for other programs to communicate and interoperate with the program, whatsoever.

There are many more problems related to the dominant company (Microsoft), but I will not elaborate it as it is beyond the scope of this paper. I just wanted to describe how the open source software and its interoperability potential can be put in jeopardy as regards the legal remedies imposed by the law. Similar issues have been discussed overseas as well, and some aspects of it will also be discussed at a later stage, with respect to the points of contact with EU legal solutions / opinions.

As regards the second market observed, i.e. mobile / tablet software market, the concerns have been raised regarding tying its search engine into the software (in somewhat similar way Microsoft was accused of tying Windows Media Player 15 years ago), because Android OEMs[15] put Google apps on home screens as part of their contractual obligations. Therefore, the "FairSearch" initiative of 17 companies led by Microsoft which includes Nokia and Oracle filed a complaint with EC, contending that Google is acting unfairly as it gives away its Android operating system to mobile device companies for free, under the

[15] OEM stands for "Original Equipment Manufacturer." This refers to a company that produces hardware to be marketed under another company's brand name. (http://www.techterms.com/definition/oem , accessed on 13.11.2013.)

condition that its applications such as YouTube and Google Maps are incorporated and prominently displayed.

Google entered mobile / tablet OS market using the open source software, therefore it embraced the open source technology and was able to offer the operating system for free. This was an extension of the previous complaint by Microsoft, concerning its practices related to its dominance on the online search and advertising markets. In this case, Microsoft claims that Google tried to take advantage of the open source software in an anti-competitive manner, in order to dominate both search and advertising markets. This complaint is being considered by the Commission at present time, so we do not have Commission opinion on this matter, but we can roughly discern some features of the case even without it.

Although we can note certain similarities between the two cases, there is one fact that makes quite a significant difference. Windows Media Player has been bundled into the Windows OS and made it difficult for users to choose some other media player instead. However, according to my humble user experience, in case of Google application embedded in Android, Google search is incorporated in Android system, but users are not obliged to use it, they can download any other search engine they desire, with no pressure. Besides, it is not difficult for them to do so, and it should not give raise to the concerns from the competition law standpoint, since the consumers would not encounter difficulties when switching to some other search engine. And using the open source software may bother some members of the open source community, but it is not unlawful nor does it violate any of its licenses. Also, the use of the open source technology is available to any competitor who decides to embrace it, and it is also highly beneficial for the consumers. I will not go in deep analysis of this complaint as I do not have enough information on the matter, but in my humble opinion the state of competition and consumer welfare seems to be considerably better than in the previous market (desktop / PC OS), and it is difficult to point at certain anticompetitive “objects or effects”.

The openness of open source technology provided many possibilities for competitors and customers, making it easy to enter the market, compete on merits, and provide a good service to the customer at low price. According to available data, mobile software market

appears to be somewhat more competitive than the laptop/ PC OS market. Open source technology is proved to be a competitive advantage, which can be used by any of the competitors on the market.

2. Competition law concerns related to MySQL in light of Sun Oracle case

Generally speaking, in case that open source software is involved in a merger, it is necessary to stress some of its specificities in order to explain sources of potential issues. Firstly, it is necessary to determine market significance of the open source software, that is its market share. The open source programs are often not distributed through classic distribution channels. Since industry reports typically cover market share associated with the sale of commercial products, those open source programs that are also free, may not be included. As there is a number of ways to gain access to the source code, it would be problematic having to identify all the suppliers.

Additionally, it is often complicated to determine the actual number of downloads executed by the users. Apart from measuring the market share, another difficulty for competition authorities is determining whether and to what extent the open source solutions can be a usable alternative in the relevant market, that is whether the open source version of a product is able to adequately replace that of the proprietary software. The functionality of the product is crucial in that respect. If we compare for instance Libre Office, which is a free and open source software containing word processing tools, spreadsheets, databases, graphics, and presentations with Microsoft Office, it is not sufficient to identify the components with similar purposes within each of them in order to conclude that they are substitutable, or that Libre Office actually represents a serious competitive threat to Microsoft Office. This could happen, if Libre Office developers were able to access necessary information from Microsoft, which would make Libre Office absolutely compatible with Microsoft Office (and most probably highly competitive).

Furthermore, certain features derived from its open source provenance make it more or less appealing to the users, compared to the proprietary versions. As regards the commercial viability of the open source software, adding new features can cause certain

problems, if they are not regulated by the community of users, which consequently has negative effects on its performance. Besides, if the competition authorities establish that the OS software is in fact a viable substitute and an efficient competitor for the proprietary software, it is necessary to quantify its market share, which is quite complicated taking into consideration that their revenue is not generated from sale of the source code or software with the incorporated source code.

Along with this, the intertwining of the proprietary and the open source licenses can cause a number of problems, concerning the open source software and its licenses, which provokes reactions of the open source software community. Also, complex nature of the dual licensing challenges the Commission opinion on dual licensing as an amicable solution for both the open source and the commercial version of MySQL existing under the same roof. Additionally, protecting the consumers as one of the objectives of competition law, in a way purports protecting the open source software (including its users of course) because of its fragile nature caused by its openness. All of these factors should be taken into account when the companies distributing the proprietary and the open source software decide to execute an acquisition or a merger.

In Sun Oracle case, the Commission exhaustively scrutinized the ways the open source nature of MySQL impacts the analysis of the diverse issues of great importance to the merger assessment of non-coordinated effects under the SIEC test. There were many aspects to be considered, such as the open source nature of MySQL when determining its "competitive force", Oracle's aptitude and incentive to degrade MySQL after the transaction, the probability of sufficient and timely replacement entry after the merger, evaluating the acceptability of the Oracle's public announcement, etc.

We will see later how these issues were addressed by the Commission, and why the open source community objected to the merger. There were also other concerns raised, such as those related to Java about the future behaviour of Oracle in case of the merger, and possible foreclosure strategy to control Java to the detriment of its downstream competitors. Besides, potential degradation of Java licensing to its downstream competitors could relegate them, and also favouring the development of new Java specifications to the exclusive benefit of its own software could make them less efficient. Later on, we will also

give an overview of these issues related to Java, and discuss it in the light of the aforementioned case.

2.1. The open source licensing

Let us now throw a glance back to the fundamentals of the open source licensing. Richard M. Stallman[16] while working at Massachusetts Institute of Technology, realized that certain software licenses had been remarkably restrictive towards the users' rights, and hindered fixing, adapting and developing software by the programmers as they blocked access to the source code because the software vendors became concerned with intellectual property rights. In his view, these licenses thwart the ways the software was being developed and used by the hackers and programmers who work on its advancement. He considered that the programmers should continue to work freely with each other, to contribute fixes for the general public good, and saw development in a community context where people are able to benefit from the innovations and improvements created by the others, while still giving attribution and appreciation for the efforts of individual programmers.

He also believed that the proprietary development of the software would cause numerous problems related to the security, loss of innovation, incompatibilities and the like, partly because it decreased number of skilled, unconstrained programmers who are able to analyze and correct the source code. Attempting to resolve these problems, in the coordination with law professors, he published GPL (General Public License). The core of the GPL license was revealed in its preamble: "when we speak of free software, we are referring to freedom, not price. Our general public licenses are designed to make sure you have the freedom to distribute copies of free software (and charge for the service if you wish), that you receive source code or can get it if you want it, that you can change the software or pieces of it in new free programs; and that you know that you can do these things."[17]

Stallman and other software freedom activists reversed “copyright” into “copyleft”, since the proprietary software developers use copyright to take away the users' freedom; and

[16] The most famous software freedom activist (since 1983.) and computer programmer, founder of Free Software Movement and president of the Free Software Foundation. Studied at Harvard University and Massachusetts Institute of Technology, where he also worked.
[17] FSF, GNU General Public License, Preamble; http://www.gnu.org/copyleft/gpl.html (accessed 17.11.2013.)

"copyleft" is used to guarantee that freedom. So, the underlying purpose of this type of licensing is to deny anybody the right to exploit a code in an exclusive way. In order for the software to reach a broad audience of developers and users, they are required to surrender all or most of the rights accorded by copyright to those who are able to distribute and exploit it. It achieves the main purpose of the open source community – freedom. This freedom is opposite to the traditional copyright licensing, and that is a "point of collision", between the rights granted by an open source license and the rights granted by a traditional IP license.

In Sun Oracle case, the attempt to resolve the issue was "dual licensing" model, which enables MySQL to be available under commercial license, and at the same time under the open source (GPL) license. However, that solution is not straightforward as it seems. In the following part I will explain why the application of the dual licensing model is not such an easy solution.

2.2. Controversy of the MySQL dual licensing model

Dual licensing is somewhat problematic since it relies upon copyright assignment and the readiness of third-party developers to renounce ownership of the code and to cede it to a controlling company or an organisation. The advantage of this model is the perception that it stimulates companies to release code under a free software license. This type of licensing and copyright assignment has been considered as tolerable compromise to some of the developers. However, Brian Aker, the primary developer of Drizzle,[18] stresses that: "dual licensing forces any developer who wishes to contribute into a position of either giving up their rights and allowing their work to end up in commercial software, or creating a fork of the software with their changes. In essence it creates monopolies which can only be broken via forking the software," [19] and it is not certain whether the new versions of the code will stay free permanently. Aker also wrote that: "The GPL's approach is to provide a stick or carrot. If you are open source, then you don't pay, assuming that you are 'the right' sort of

[18] A fork of MySQL

[19] "MySQL – A Controversial Dual Licensing Model", by Richard Hillesley; "Linux User and Developer", Issue 132 – The monthly magazine for the GNU generation, 22.9.2013. (www.linuxuser.co.uk, accessed on 24.11.2013.)

open source. If you are closed source or pick a license which is not compatible with the GPL, you are forced into paying for use of a commercial license. When you 'pay' for open source, the freedom that was originally offered to the end user is removed. In the case of incompatible open source licenses, you too are forced into the position of removing the freedom granted and possibly the freedom you granted to your own work. Take any current distribution of a Linux distribution and do the research on licences within it. The tangled mess that is found will confuse anyone. MySQL itself was only able to solve some of this by offering a 'FLOSS Exception' to the portions of the code it owns."[20]

The problem of the dual licensing of MySQL reached a critical stage when Sun purchased MySQL in 2008, and during the latter Oracle's acquisition of Sun. This was especially emphasized after the Oracle/Sun acquisition, when Oracle decided to add certain extensions to the commercial version of MySQL by means of using previously available APIs, that were unavailable to users of the GPL'd[21] version of MySQL. Actually, MySQL AB[22] contends that the commercial version of MySQL which is equivalent to that of free software, can be added to proprietary embedded products with no requirements that manufacturers have to comply with as regards the releasing changes back to the community, and therefore this grants a commercial advantage to MySQL AB. Some developers were not really persuaded about this line of reasoning.

A developer of PostgreSQL, Josh Berkus asserted that most of the revenue of MySQL AB was generated by means of traditional mechanisms for selling the open source software, such as subscriptions, support and maintenance. He also said: "MySQL AB made the majority of its direct income from support contracts rather than from licensing. While these were termed 'subscriptions' and supposedly came with commercially licensed add-ons, customers were clear that what they wanted and were buying was MySQL's fanatical support team, not any of the proprietary add-ons, most of which were inferior to open source alternatives."[23]

[20] Ibid.
[21] Licensed under GPL
[22] Software company, acquired by Sun Mycrosistems in 2008.
[23] "MySQL – A Controversial Dual Licensing Model", by Richard Hillesley; "Linux User and Developer", Issue 132 – The monthly magazine for the GNU generation, 22.9.2013. (www.linuxuser.co.uk, accessed on 24.11.2013.)

The major problem as regards the dual licensing model is its dependability upon copyright assignment. It is difficult for a project to advance when there are issues related to the ownership of the code. Mechanisms such as copyright assignment, dual licensing model and contributor license agreements (CLA) can be interesting for participating companies, but they are an obstacle since they hinder the sense of community which is crucial for an open source project. Combination of dual licensing and copyright assignment generate lack of transparency as regards the upcoming direction and licensing related to the code. In case the right of possession of a piece of the code is allocated to a third party, they are typically granted right to release the code under a different license, sooner or later, and the new license can be different compared to that of the open source programs. A code that is being "owned" boils down to being an "asset" that can therefore be sold or reused. Hence, a project which represents a specific unit that is not owned by anyone is considerably more appealing to the developers, and it is more resistant to divisions as compared to a product that is being controlled by a particular firm. Therefore, in case the code for certain project is owned by one company, it is unlikely that other firms will step out and want to contribute to the project.

A proprietary software company is attracted by the open source, since it provides connection to the whole community of developers and users, as the benefits of such a connection purport reduced expenses, chances for prospective collaboration, software libraries and possibilities to access excellent input from different types of sources. Typical corporate culture purports maximizing returns on the investments as the major objective, without sentimental views about the open source elements. However, the open source is more than just a license and promise and from passed events we can see that companies often change, and "assets" are being purchased and sold.

There is no much use in a license when the ownership of a project has been ceded to someone else, with no warranty, or in case the code is intertwined with certain IP agreements which grant control to the holding firm, and not to a non-profit institution. Brian Aker explained the antagonism of dual licensing and the open source in the following way:

"At the heart of dual licensing is 'ownership rights', which inevitably come into conflict with the nature of open source.

The open source projects that preserve the ability to dual-license come into conflict with the developers who contribute the code. For the projects to continue, it requires the original developer to give up their rights to the code via copyright assignment (there is some debate on whether copyright can be held in joint, but this is often disputed by lawyers). Thus dual licensing forces any developer who wishes to contribute into a position of either giving up their rights and allowing their work to end up in commercial software, or creating a fork of the software with their changes. In essence it creates monopolies which can only be broken via forking the software…"[24]

2.3. Discussion on the Commission decision related to MySQL

Just after the announcement of intended merger, Oracle received Statement of Objections from the European Commission, expressing concerns related to the potential merger. The Commission dwelled on whether the merger would lead to "significant impediment of effective competition within the EEA". Given that database market was considerably concentrated and divided among the three major database vendors,[25] it is no wonder the Commission decided to raise certain questions. The merger fell within the framework of the European Merger Regulation, as the transaction in question constitutes a concentration within the meaning of Article 3(1)(b) of the Merger Regulation. So there are multiple problems to be thoroughly examined; would the competition be affected as the operation meets the requirements of Article 3(1) (b), and crosses the turnover thresholds contained in the Article 1(2) of the Merger Regulation.

The issues were related to MySQL, Java, middleware and IT stack. In this part we will thoroughly examine problems related to MySQL. MySQL was Sun's main database product which was the world's most popular database in 2008 according to Gartner Group[26] which was about to be integrated into Oracle's software products. Furthermore, another product of Sun's – Java "development environment" with many of its open source implementations

[24] Ibid.
[25] Oracle, IBM and Microsoft were the major players in database market (with total of 85% in terms of revenue)
[26] https://www.gartner.com/doc/580908 (accessed on 16.11.2013.)

widely accessible and the major intellectual property rights related therewith, held by Sun was another source of concern.

Besides, yet another important "development environment" is .Net, Microsoft's closed source environment which can be developed exclusively on Windows. As regards the middleware, Sun and Oracle were not close competitors on the market and therefore no special issues are related thereto. Apropos Solaris, the Commission considered it unlikely that it could be degraded by Oracle, or deprived of interoperability because this would not provide any benefits for the company whatsoever. However, Oracle inhibited development of Opens Solaris for some time and as a result, the operating system was "killed" by Oracle. However, this was just a parenthetical remark and will not be further developed here.

Since MySQL was major controversial issue in the case, most of the investigation and appraisal was dedicated to it. In appraising the market definition of MySQL, there was an issue whether embedded and non-embedded databases should constitute a single market, or each should represent a specific market, which was resolved in favor of single market. The Commission has confirmed that MySQL database was a leader in web segment, in spite of its small market share, so there were questions to be answered as regards restraining the competition. The Commission had to assess the competitive constraints exerted by MySQL before the transaction, the extent to which it would be removed after the merger, and how would other open source databases replace the competitive constraint of post merger.

The Commission found that MySQL has a potential to exert competitive constraint for Oracle in SME[27] segment, and parts of embedded segment, excluding high-end segment and that Oracle will face competitive pressure from the proprietary database vendors that imposed competitive constraint on it before the announcement of merger. It found that the incentives and ability to degrade MySQL post merger would be constrained by the open source nature of MySQL.

The Commission accepted Oracle's public pledges to enhance MySQL, and to continuously develop it along with its commercial version, and not to require commercial licenses from any third party storage engine vendors, since it considered that Oracle's

[27] Small and medium enterprises

reputation within the open source community would come into question in case of the breach. Consequently, this would have negative implications for Oracle. Besides, it concluded that PostGreSQL have potential to constrain Oracle to a considerable extent, and that it can "fill in" the gap left by MySQL, in timely and sufficient manner. It also concluded that that the forks of MySQL can develop to the extent to represent a competitive constraint for Oracle. In the following part I will reword the concerns of the open source community expressed in the letter [28] that was sent to the former Commissioner for Competition, Ms. Neelie Kroes, and reasons against the clearance of the merger, which will be discussed in the light of the Commission decision and factual situation at present time.

First problem they addressed referred to anticipated restrictions as regards the development of the functionality and performance of the MySQL software platform. Therefore, if Oracle acquires Sun, limiting the development of the functionality and performance of the MySQL is likely to happen, and if it does, it will certainly produce harm for those who use MySQL to power the applications. They also argued that Oracle being the leading seller of the proprietary database software designed for very large enterprises was in position of market dominance in this market segment, and this enabled it to charge high fees and obtain considerable earnings. Also, MySQL developed greater functionality, dependability and improved performance, and turned out to be an exceptionally significant part of the database market – much greater than can actually be measured by traditional market share analysis based on revenues. In addition, MySQL presented a significant competitive pressure on prices charged for the proprietary databases thus generating moderation or lowering licensing fees charged by Oracle or Microsoft, including turncoats of numerous enterprise database services to MySQL platform.

[28] The authors of the letter are: Richard Stallman, who is a software developer and software freedom activist and the main author of the GNU General Public License, the most widely used free software license. Stallman launched the Free Software Movement in 1983 and led the development of the GNU operating system (normally used together with the kernel Linux). KEI (Knowledge Ecology International) is a non-profit public interest organization, supporting work carried out earlier by the Consumer Project on Technology (CPTech), an organization that has in the past participated in a number of merger reviews, including those involving legal publishing, retail distribution, and media concentration and telecommunications regulation. KEI uses MySQL to power several different web page platforms, including those run by Free/Libre/ and Open Source (FLOSS) content management systems such as Joomla, Drupal and Wordpress. ORG (Open Rights Group) is a non-profit company founded in 2005 by 1,000 digital activists, is the UK's leading voice defending freedom of expression, privacy, innovation, consumer rights and creativity on the net.

As the behavior concerning the price lowering is ultimately favorable for the users, therefore this economic benefit should be preserved by means of prohibiting the merger. Also, since Oracle's database is considered to be dominant on the "old" database market, MySQL is dominant for the emerging database markets, therefore Oracle considers it to be its most important future competitor. Along with this, they stressed few reasons why the argument stating that Oracle does not have interest or incentive to harm or degrade MySQL is weak. Furthermore, in case that Oracle does not host properly the GPL version of the code, other companies and individual programmers can take over the future development, so they would be able to easily "fork" the GPL'd code into the new platform. Let as now look at the reasons why the open source community was convinced that Oracle had an actual interest to degrade the open source version of MySQL database.

Firstly, Oracle would be the only entity that can release the code except under the GPL, in case it acquires MySQL, considering that MySQL employs dual licensing approach to generate revenue. Therefore, Oracle would not be bound to diligently sell MySQL commercial licenses, or even to price in commercially reasonable manner. Besides, Oracle would not be bound to use the proceeds from the licenses for purposes of advancing MySQL. Hence, while making decisions on the matter, Oracle encounters clear conflict of interest; continuously developing a powerful, feature rich free alternative to its core product.

Since the exclusive right to sell commercial licenses belongs to the original rights holder only, new forked versions of the code could not practice the parallel licensing, and would not be able to induce resources in order to support MySQL platform development. This acquisition would hinder the development of a FLOSS[29] database platform and it was very likely that it would alienate and disperse MySQL's core community of developers. It is possible that only after several years another database could compete with current opportunities now available to MySQL, since it would take time for any other database platform to attract and cultivate sufficiently large number of developers and achieve a close customer base.

[29] Free/ Libre Open Source Software

Through evolution of the GNU GPL license, Oracle could be able to determine forking of MySQL. Since GPLv2 and GPLv3 are different licenses and they both entail that any modified program inherits the same license as the original one, there are many essential legal hurdles to combine code from programs that are licensed under different versions of GPL. At the present time, MySQL is available under GPLv2 only. Since numerous different FLOSS software projects are moving to GPLv3 and MySQL license lacks this clause, it will be available under GPLv2 only, so it will be impossible to combine the code of many GPLv3 covered projects with that of MySQL. As the forking will depend on the contributions of the FLOSS community, the absence of more flexible license for MySQL will constitute significant hurdles to a new forked development path of MySQL. These were concerns expressed in the letter sent to the former Commissioner for Competition, Ms. Neelie Kroes.

Apropos the first argument, in my humble opinion, degradation of MySQL escaped exceptionally detailed scrutiny of the Commission. Since the Commission has proven that MySQL is the leader for web databases, for one traditional software vendor the only economically rational solution is degrading the product which contains open source software and advancing the commercial version. The period that Oracle is committed to comply with the promises it made is set to five years as it was considered to be long enough to provide availability of an enhanced version of MySQL "until other open source database vendors possibly including forks of MySQL have further developed their market position."[30]

If we reflect on the open nature of MySQL, we can grasp that Oracle in fact had a similar product, and from an economic perspective, it would be unreasonable to expect that Oracle would actually invest its resources in advancing a product that is not generating any revenue whatsoever. Besides, it is a competing product, therefore it is just a matter of time when MySQL under the GPL license would vanish.

In addition, the fact that the commitments made by Oracle are not legally binding (except for Continued Availability of Storage Engine APIs, which purports that Oracle will

[30] Case No COMP/M.5529 – ORACLE/ SUN MICROSYSTEMS para. 653.

maintain and regularly enhance MySQL's Pluggable Storage Engine Architecture in order to provide users the choice between native and third party supplied storage engines, non-assertion; "As copyright holder, Oracle will change Sun's current policy and shall not assert or threaten to assert against anyone that a third party vendor's implementations of storage engines must be released under the GPL because they have implemented the application programming interfaces available as part of MySQL's Pluggable Storage Engine Architecture"[31], and license commitment; "Upon termination of their current MySQL OEM Agreement, Oracle shall offer storage vendors who at present have a commercial license with Sun an extension of their Agreement on the same terms and conditions for a term not exceeding December 10 2014. "[32]) shows that the Commission did not completely understand the danger that loomed over the open source version of MySQL.

In any event, the promises Oracle committed to adhere to during the five years after the transaction, have not been fulfilled. Throughout the year 2012, Oracle has been progressively working on degrading GPL'd MySQL, although the five years promise is to expire only in December 2014. In July and August 2012, as a result of continuous efforts to close up the open source version of MySQL, Oracle started with hindering development of MySQL by means of holding back the test cases in MySQL's latest release.

In a software development process, test cases are considered to be one of the most important part of the software testing. They are being used by the quality team, development team and the middle management executives. The importance of the test cases are reflected in the fact that they provide the conditions through which a developer can validate and verify a specific functionality or feature of the system. Therefore, the test cases provide the information about what should and can be improved in a specific system, and the steps that developers have to take, it also provides the values of input data and the results after executing a particular test case. Simply put, the test cases are "in charge" for the whole testing process. The issue stems back to first half of 2012, in relation to the discovery that the latest release of MySQL provides bug fixes, yet not a single one had any test cases associated with it. This gave rise to the wide range of problems among the

[31] Case No COMP/M.5529 – ORACLE/ SUN MICROSYSTEMS, p.42, footnote 154
[32] Ibid.

developers, as they were not able to get the feedback or any kind of guarantee that the problem is actually fixed. This made quite clear that the efforts of Oracle that caused difficulties for the use of MySQL, and that they were directed towards closing MySQL, in spite of the promises and commitments they made. This caused quite a turmoil within the open source community of developers, and gave rise to questions concerning Oracle's views on the open source, and closed source software.

Since 1999 MySQL used a testing framework named mysql-test. Over the time, new tests have been developed for the new features, so the regression tests assured that bug fixes are final. Some developers, for example from Twitter, actually rely on the testing framework. MySQL serves as "persistent storage technology behind most Twitter data: the interest graph, timelines, user data and the Tweets themselves."[33] at Twitter.

Besides, Oracle withdrew the revision history for MySQL. The revision history is a recording of all the changes made to the millions of lines of the source code, grouped in "change sets". A change set reveals all the changes for a specific feature. It contains on the bug fixes, such as who fixed a bug, when, why it had to be fixed, etc. Therefore, if the revision history is removed, the developers would have to guess what has been fixed, and what has not been fixed, whether it is necessary to fix certain bugs, etc. These information are of great practical importance for the developers, yet they are being deprived of it. But this is not the end of story. These missing test cases and revision history are converted to the closed source, by Oracle.

After the acquisition, within the next few years Oracle Corporation showed certain level of reluctance towards the open source software. As regards the Commission opinion "the notifying party will become the "owner" of other open source products, such as Java, OpenSolaris and OpenOffice. The notifying party already offers some open source products such as Oracle Enterprise Linux and Oracle VM. It will certainly have a continued interest in the success of a number of open source products..."[34], the time showed that this "continued interest" was quite short-winded. In the summer of 2010, "...the Oracle executives stated that the open source, community-driven OpenSolaris project as conceived

[33] Adrian Bridgwater, April 2012. http://www.computerweekly.com (accessed on 24.11.2013.)
[34] Case No COMP/M.5529 – ORACLE/ SUN MICROSYSTEMS para. 657.

and built by Sun Microsystems five years ago is dead. Get over it. Instead of OpenSolaris being coded well ahead of the commercial Solaris release that it will eventually become, Oracle is doing a 180-degree turn: now the only open source version of any future Solaris stack will come after the commercial product ships."[35] Practically, the "continued interest" of Oracle to advance the open source products, according to the Commission, is now gone. In the summer next year, Oracle renounced Open Office after key members of the Open Office.org community and some of the major corporate contributors forked Open Office.org in order to create a vendor-neutral alternative.

In November 2010, Oracle changed the pricing of standard support, which started at $2,000, as opposed to Sun Microsystems' basic and silver packages, which started at $599 and $1,999. At the high end, a cluster carrier grade edition priced at $10,000 has been introduced by Oracle. Previous top-level support package of Sun, platinum, was priced at $4,999. In my opinion, this fact speaks for itself. In November 2013, announced that it will no longer support commercial version of GlassFish. GlassFish is an open-source application server project initially created by Sun Microsystems for the Java EE[36] platform.

In addition, one other commitment made by Oracle to "...to continue to enhance MySQL and make subsequent versions of MySQL, including Version 6, available under the GPL. Oracle will not release any new, enhanced version of MySQL Enterprise Edition without contemporaneously releasing a new, also enhanced version of MySQL Community Edition licensed under the GPL. Oracle shall continue to make the source code of all versions of MySQL Community Edition publicly available at no charge."[37] has been violated by Oracle in September 2012, when Oracle released only three commercial extensions for MySQL, and no extensions for the open source version of MySQL. And this is perfectly reasonable from an economic perspective, as Oracle had to justify charging money for a non-open version of MySQL, therefore it had to add value to the commercial, proprietary version.

On the Internet, one could read tens of articles inspired by bitter disappointment of the open source community members. Their disappointment by Oracle's behavior is clearly

[35] Open Solaris axed by Ellison, 13.8.2010. (http://www.channelregister.co.uk, accessed on 24.11.2013.)
[36] Enterprise Edition
[37] Oracle's public announcement, Case No COMP/M.5529 – ORACLE/ SUN MICROSYSTEMS, footnote 154

visible, therefore its reputation among member of the open source community at the moment is quite poor. This did not happen at once, Oracle has been directing its actions against interests of the open source community for some time now, and its unfavorable actions directed against MySQL (such as adding commercial extensions to it) is just a link in the chain. After a range of the open source products that Oracle "killed" or tried to "kill", Oracle recently published a White Paper which they used to recommend to the Department of Defense of the United States not to opt for the open source software. This is obviously an offensive move in order to advertise their own products, but they did not promote the features of high quality, they instead informed the government that the source software is an expensive software (in terms of long-term costs) of a poor quality.

They mentioned as an example a failed project of developing a single electronic health record system, a project that took five years, and nearly $12 billion, but without explaining the reasons of the expensive delay of the project. Oracle warned Department of Defense that "oversimplifying the consequences of using open source software might be catastrophic", and implied that "taxes were wasted because of open source".

As we can see, behavior of Oracle corporation in the years following the merger has been contrary to the expectations of the Commission. Therefore the argument of "reputation" among members of the open source community does not really hold water. Also, the controversy related to the dual licensing model made it look less attractive as an amicable solution from the open source community standpoint. As regards to PostgreSQL, from this perspective we could say that it essentially met the expectations since it has been fiercely competing with MySQL, although some developers stress superiority of MySQL, in my humble opinion it seems that difference between the two boils down to the nuances. There is also a community fork of MySQL – Maria DB, which is another high quality alternative. In addition to this, an interesting fact is the recent lawsuit of Oracle against Google. The proceeding has been long and complex, and while Oracle was able to take the full advantage of the open source software, it has created quite a stir when Google have done the same. Since this is an ex-post analysis, and naturally the Commission could not anticipate the future behavior of Oracle, in my opinion the letter opposing the merger that was sent to the Commission, contained certain relevant points as regards the future behavior of Oracle.

3. Interoperability concerns regarding the open source software

We saw how and why it was unfavorable for the open source MySQL database to approve the merger between Sun and Oracle. In the following part, I will explain how the Commission tried to approach closer to the open source software and its interoperability by publishing the European Interoperability Framework, and what are the problems related to it.

3.1. Controversy over European Interoperability Framework

This is how IDABC [38] defines an interoperability framework: "An Interoperability Framework can be defined as the overarching set of policies, standards and guidelines which describe the way in which organisations have agreed, or should agree, to do business with each other. An Interoperability Framework is, therefore, not a static document and may have to be adapted over time as technologies, standards and administrative requirements change."[39]

European Interoperability Framework defines interoperability in the following way: 'Interoperability, within the context of European public service delivery, is the ability of disparate and diverse organisations to interact towards mutually beneficial and agreed common goals, involving the sharing of information and knowledge between the organisations, through the business processes they support, by means of the exchange of data between their respective ICT systems.'[40] EIF set up four levels of interoperability;

1. Legal interoperability

2. Organisational interoperability

3. Semantic interoperability

4. Technical interoperability

Technical interoperability covers the technical aspects of connecting the information systems.

[38] Interoperable Delivery of European eGovernment Services to public Administrations, Businesses and Citizens.
[39] EIF - European Interoperability Framework for pan-European eGovernment services (http://ec.europa.eu/idabc/en/document/2319/5644.html#what , accessed on 6.1.2014.)
[40] Article 2 of Decision No 922/2009/EC of the European Parliament and of the Council of 16 September 2009 on interoperability solutions for European public administrations (ISA) OJ L 260

This includes the interface specifications, data integration services, data presentation and exchange, etc. According to EIF, "technical interoperability should be ensured, whenever possible, via the use of formalised specifications, either standards pursuant to EU Directive 98/34 or specifications issued by ICT industry fora and consortia."[41]

Standardization has an essential role in the development of the software industry, taking into account that it represents a key factor as regards the innovation and interoperability. The standards facilitate ICT products to be able to share the information and to interoperate with one another. The text of EIF is broadly supportive towards the open standards, and the degree of openness of a formalized specification is considered as an important factor in determining the opportunities for reusing and sharing the software components which implement the specification. It goes further and states "Due to their positive effect on interoperability, the use of such open specifications, characterized by the features mentioned above as well as the sharing and reuse of software implementing such open specifications, has been promoted in many policy statements and is encouraged for European public service delivery. The positive effect of open specifications is also demonstrated by the Internet ecosystem."[42]

In the EIF, the Commission defines the openness as the willingness of persons, organizations and community members to share the knowledge and encourage debate within the community, with an ultimate objective of advancing the knowledge and its usage in order to resolve problems. They also consider that, if the principle of openness is fully applied, all the stakeholders would have the similar possibility to contribute to the development of the specification, it would be available to everyone. Also, the IP rights concerning the specification are licensed on FRAND[43] terms, or a loyalty free basis in a manner that permits the implementation in both the open source and the proprietary software. We could say that EIF tackles the interoperability in a proactive way, and takes a clear view towards it, since it also states that "When establishing European public services, public administrations should prefer open specifications, taking due account of the coverage of functional needs, maturity and market support."[44]

[41] EUROPEAN INTEROPERABILITY FRAMEWORK FOR EUROPEAN PUBLIC SERVICES, p 29
[42] Ibid. p31
[43] Fair, reasonable and non-discriminatory terms
[44] EIF, Recommendation 22

They committed to continue to support the National Interoperability Framework Observatory (NIFO), reasserting the significance of harmonizing such approaches across the borders. In addition, the Commission committed to coordinate its internal interoperability strategy with the EIS (European Interoperability Strategy) by means of the eCommission initiative. EIF seems to give due consideration to the use of the open standards related to the open source and closed source software.

The EIF introduced the requirement that the specifications have to be "mature and sufficiently supported by the market". Furthermore, it requires that IP rights related to the specification have to be licensed on FRAND terms, or on a royalty-free basis, as this could enable implementation in both proprietary and open source software. This would enable companies applying different business models to compete on equal footing in terms of providing answers for the public administrations, while at the same time the administrations that implement the standard into their software could share the software with others, under conditions of an open source license.

Few years ago, BSA[45] argued that the policy expressed in the EIF would promote giving away the patents by the companies that want to win the public sector contracts. However, the Commission stated that "public administrations may decide to use less open specifications, if open specifications do not exist or do not meet their interoperability needs. In all cases, specifications should be mature and sufficiently supported by the market, except if used in the context of creating innovative solutions."[46]

Although the governments are not able to migrate to the full openness promptly, the EIF probably gives too much leeway for the governments not to fully enforce the principle of openness. They adduced many reasons for this, but it could be just a challenge which is to be overcome. Besides, the notion of "open standards" as part of the internationally established terminology, was not used in the EIF. Instead, the notion of "formalized specification" was introduced. How are "formalized specifications" and "standards" exactly related? In accordance

[45] The Business Software Alliance is a trade group established in 1988 and represents a group of world's largest software producers. BSA is a member of International Intellectual Property Alliance (IIPA). The main activity of the BSA is preventing copyright infringements as regards the software produced by its members. The Alliance include SAP, IBM, Microsoft, HP and Dell.

[46] Controversial European Interoperability Framework Announced, Jennifer Baker (http://www.idgnews.net/ accessed on 11.12.2013.)

with FAQ provided; “The word "standard" has a specific meaning in Europe as defined by Directive 98/34/EC. Only technical specifications approved by a recognized standardization body can be called a standard. Many ICT systems rely on the use of specifications developed by other organizations such as a forum or consortium. The EIF introduces the notion of "formalized specification", which is either a standard pursuant to Directive 98/34/EC or a specification established by ICT fora and consortia. The term "open specification" used in the EIF, on the one hand, avoids terminological confusion with the Directive and, on the other, states the main features that comply with the basic principle of openness laid down in the EIF for European Public Services.”[47]

This may be true, but in fact Europe seems to be lagging behind as regards the terminology. The EIF does not provide the definition of open standard. In lieu of this definition, EIF describes “openness” in the following way: “All stakeholders have the same possibility of contributing to the development of the specification and public review is part of the decision-making process; the specification is available for everybody to study; intellectual property rights related to the specification are licensed on FRAND [(Fair) Reasonable and Non-Discriminatory] terms or on a royalty-free basis in a way that allows implementation in both proprietary an open source software.”[48] Apparently, this means that royalty-free is not more open than FRAND, even though royalty-free basis is actually a specially open subset of FRAND. Except in case of some special requirements, what is licensed on a royalty-free basis is usually considered as available under the FRAND terms.

Nevertheless, there is no generally accepted definition of FRAND terms. It is quite difficult to say what is exactly “fair,” “reasonable,” or even “non-discriminatory.” Standards organizations usually do not like to interfere in overseeing the license fees. Hence, the amount of the fees are agreed upon in the negotiations among the patent holders and the implementers. This way is quite convenient for the patent holders, since they have a great advantage as the implementer needs the license badly, so that he can enter the market and will accept the patent holder's opinion regarding “fair, reasonable and non-discriminatory” terms. However, Free Software

[47] MEMO/10/689 “Commission adopts Interoperability Strategy and Framework for public services - frequently asked questions”, Brussels, 2010 (europa.eu/rapid/press-release_MEMO-10-689_en.doc, accessed on 11.12.2013.)
[48] European Interoperability Framework, p 26

Foundation came up with the following definition of open standard, that originates from "Certified Open"[49]

"An Open Standard refers to a format or protocol that is

1. subject to full public assessment and use without constraints in a manner equally available to all parties;

2. without any components or extensions that have dependencies on formats or protocols that do not meet the definition of an Open Standard themselves;

3. free from legal or technical clauses that limit its utilization by any party or in any business model;

4. managed and further developed independently of any single vendor in a process open to the equal participation of competitors and third parties;

5. available in multiple complete implementations by competing vendors, or as a complete implementation equally available to all parties."[50]

There is also a definition of open standards by Bruce Perens[51], which is comprised of six principles;

1. Availability - Open Standards are available for all to read and implement.

2. Maximize End-User Choice - Open Standards create a fair, competitive market for implementations of the standard. They do not lock the customer in to a particular vendor or group.

[49] Certified Open® is designed to evaluate technical and commercial lock-in. It promotes fair competition and increases the ability of suppliers to compete effectively in the provision of software, hardware and services. The Certified Open Products & Services Framework is an industry-agreed, highly granular framework that defines the characteristics of products and services.

[50] Official website of Free Software Foundation Europe (http://fsfe.org/activities/os/def.en.html accessed on 18.12.2013.)

[51] Co-founder of Open Source Initiative and creator of Open Source Definition

3. No Royalty - Open Standards are free for all to implement, with no royalty or fee. Certification of compliance by the standards organization may involve a fee.

4. No Discrimination - Open Standards and the organizations that administer them do not favor one implementor over another for any reason other than the technical standards compliance of a vendor's implementation. Certification organizations must provide a path for low and zero-cost implementations to be validated, but may also provide enhanced certification services.

5. Extension or Subset - Implementations of Open Standards may be extended, or offered in subset form. However, certification organizations may decline to certify subset implementations, and may place requirements upon extensions.

6. Predatory Practices - Open Standards may employ license terms that protect against subversion of the standard by embrace-and-extend tactics. The licenses attached to the standard may require the publication of reference information for extensions, and a license for all others to create, distribute, and sell software that is compatible with the extensions. An Open Standard may not otherwise prohibit extensions.[52]

In addition, some of the adopted terms can cause certain problems for the open source software development model in the context of patents and royalty payments. The open source software has to be available under one of the wide range of the open source licenses. Some of the authors suggest that the most frequently used licenses disable the development of software that requires royalty payments (Oksanen and Valimaki). In the context of software, the standards have to purport the compatibility with the free and open source licensing terms, in order to allow all suppliers to have fair access to competition for the government contracts (Ghosh), hence the possible problems regarding the patents and royalty payments have to be taken into consideration. Even in the Regulation No 1025/2012, which should be one of the last steps of European Standardization Reform, the open standards are not precisely defined. Therefore, majority of Member States will "likely conclude that they will go on referencing and using standards beyond

[52] http://perens.com/OpenStandards/Definition.html (accessed on 18.12.2013.)

those created by the three European endorsed monopolists of standardization, CEN [53], CENELEC[54] and ETSI[55]".[56]

So if, on the one hand we do not have so clear a rule, and on the other hand, in practice there are standards originating from global standardization organizations, dominating in the IT industry, how do we understand this? Should the Member States continue using non- ESO standards, if they are called a different name? Besides, in the context of software, standards must be compatible with the free and open source software licensing terms to enable all the suppliers to have a fair access to competition for the government contracts (Ghosh 2005), therefore the potential issue with patents and royalty payments must be considered.

Some can argue that that royalty-free licenses quash patentees' incentives and divest a society of the useful technologies. However, the patent owners may want to license a patent on a royalty-free basis for many reasons, it can give them an advanced position regarding the product lead-time, or allow compatibility with the developing products, etc. The domain of this way of licensing is commonly limited to the extent which is essential to practice the standard, and all other uses of the patented technology would require the defrayal.

Taking into account all the controversy, interests at stake, and delays related to it, we could say that EIF is a positive step towards the interoperability. This is an important initiative to stimulate administrations throughout the European Union to augment the prospect of the

[53] The European Committee for Standardization (CEN) is a business facilitator in Europe, removing trade barriers for European industry and consumers. Its mission is to foster the European economy in global trading, the welfare of European citizens and the environment. Through its services it provides a platform for the development of European Standards and other technical specifications. CEN is a major provider of European Standards and technical specifications. It is the only recognized European organization according to Directive 98/34/EC for the planning, drafting and adoption of European Standards in all areas of economic activity with the exception of electrotechnology (CENELEC) and telecommunication (ETSI). (http://www.smartgrids.eu/CEN-CENELEC-ETSI accessed on 26.12.2013.)

[54] CENELEC is the European Committee for Electrotechnical Standardization and is responsible for standardization in the electrotechnical engineering field. CENELEC prepares voluntary standards, which help facilitate trade between countries, create new markets, cut compliance costs and support the development of a Single European Market. (Ibid.)

[55] ETSI is The European Telecommunications Standards Institute produces globally-applicable standards for Information and Communications Technologies (ICT), including fixed, mobile, radio, converged, broadcast and internet technologies. ETSI officially recognized by the European Union as a European Standards Organization. (Ibid.)

[56] European Interoperability Framework – a new beginning? (Trond Undheim, 2010. Source: https://blogs.oracle.com, accessed on 22.12.2013.)

information and communication technologies. But how will it work in the future? Will the aforementioned shortcomings make it partially inapplicable? Or the previous case law and Commission's guidelines may fill the blanks? As I already said, this is a significant step towards the interoperability, and the future will probably bring some refinements as regards the definition of open standards and other deficiencies of the EIF. In the following part I will explain the software interoperability and its importance from the view of the open source software developers.

3.2. Interoperability and software

We talked about the interoperability and its definition in general terms, extracted from text of the EIF, speaking in general terms. However, the interoperability definition concerning the computer software requires a few additional details. Interoperability can be broadly defined as a measure of the degree to which different organizations, systems and individuals have the ability to act synergistically in order to achieve a common objective. In context of the computer software, it can be defined in the sense of syntactic interoperability and semantic interoperability. Syntactic interoperability depends upon particular data formats, communication protocols and exchange of information. The system included in the procedure can process the data, but it is not certain whether the interpretation will be the same. Semantic interoperability is present when two systems are able to interpret the exchanged data in a meaningful and correct way, so that it can generate valuable results.

For the purpose of this discussion, I think this definition will be enough, without going down to the smallest detail. Firstly, it is important to point out that there is no definite, generally accepted definition of the interoperability among the software developers. Besides, this question is controversial, since most of the software producers wish their proprietary software technology become standard, so it can monitor the "ecosystem". This would enable them to charge royalties, or exclude the competitors who are able to produce an interoperable software. On the other side, there are the open source developers and users who wish that the standards have the open source, non-proprietary nature with no intellectual property protection whatsoever.

Therefore, in terms of the open source software, the question is: why is the interoperability such an issue? I mean, if the code is open, and the developers can easily make the desired changes in order to achieve interoperability, what is the problem? Besides, the developers typically use the open standards and build modular code, and what is wrong with that? The experience of the developers showed that this is true in case of very successful open source projects. However, it is not applicable to all open source projects. For instance, there is Drupal as one of the very successful open source projects, which attained its success because it obtained interoperability at an early stage. Modular design of the project enabled simultaneous development by individual developers throughout the world, and users were able to "plug and play" in a simple way, facilitating drive adoption. However, there are many excellent ideas that have been disregarded because the interoperability has been neglected. Hence, the open source industry encounters an important unfulfilled opportunity. This pertains to both development communities and commercial open source vendors. They are mostly small-scale vendors, and they put efforts into being focused on certain product properties in order to better compete among themselves, and with proprietary software vendors. The interoperability is required, but often it is underdeveloped due to the limited resources and time, so the vendors are prone to leave this question for later. However, the interoperability is quite important to be left behind, since it enhances a competitive potential of a product/ software. If the software lacks interoperability, a good deal of prospects cannot be adopted and this is consequently lowering the earning potential. What is more, competing on features continues over and over again.

In effect, the vendors need to obtain the interoperability in the very first version of their software. But that is not always easy when the proprietary software vendors are trying to prevent the access to the relevant information and keep the competitive advantage. How to prevent proprietary software technology to set the standards that are favorable to the closed source software, to the detriment of the open source software? There is the European Interoperability Framework, but as we already mentioned, it does not completely solve the problem. Besides, the European Standardization Reform and the Regulation, in spite of being supportive towards the open source software, does not offer a clear solution for standardization issues related to it.

3.3. The interoperability problems in Sun Oracle case

Here we will take a look at the interoperability problems arising out of Sun / Oracle merger. In this case, Sun controlled the most important IP rights, which concerned the companies developing the software using Java language. The main alternative to Java environment was .Net (closed and proprietary) environment (except for the fact that Java runs on all the operating systems, and .Net runs only on Windows). Sun proposed four mechanisms of licensing Java technology: open source licenses, commercial licenses, mix of royalty-free and commercial licenses, and royalty-free licenses for binary versions.

The concerns have been raised as regards the licensing Java to its downstream competitors. Oracle would be able to refuse to license the IP rights, and put the competitors in an unfavorable position. Oracle might also give priority to the development of new Java specifications in favor of its own software, which could make the downstream competitors less competitive. Oracle stated that the essential value of Java is contained in its shared and widespread standardization, so departing from the current model would be contrary to their interest. Taking into consideration the previous analysis regarding MySQL and the current time perspective, we can say that it was certainly not easy to make the decision on the matter. I mean, Oracle stated that the value of the project was in its open nature, and that they valued it for that reason, but then they have been gradually and discreetly working on its degradation. It seems to me that the Commission constantly strived to find a pretext to approve the merger, since their overseas counterpart approved it long time ago. Therefore, it would be logical to assume a similar line of reasoning as regards the situation related to Java. "The complainants did not provide any element on the basis of which it could be concluded that any of these potential products – which remain, in any case, unidentified – could constitute essential inputs to the competitors' products. Without such a conclusion, which constitutes a fundamental prerequisite for any possible analysis of vertical anti-competitive concerns, any abstract reasoning on what Sun/Oracle "could" develop outside the JCP remains pure speculation, and should not be taken into consideration in the Commission's assessment."[57]

There was no actual evidence, but a high possibility (pointed out and

[57] Case No Comp/m.5529 – Oracle/ Sun Microsystems, para 879

explained by the open source community) for Oracle to start degrading MySQL, yet they did it. I think I do not jump to conclusions if I say that it is quite difficult to provide a certain, conclusive evidence that Oracle will or will not jeopardize the open source software which would be under their control after the merger. However, it is important that they would have interest to do it and in my opinion, the Commission should have considered the possibility of this option and prevent the merger, because of fragile nature of the open source software.

Few years after the transaction, Java appears to "do well" under the auspices of Oracle, according to IDC analysts. The programme director of the application development software at IDC, Al Hilwa said that: "Oracle was thought by many, especially in the open-source software community, to be the antithesis of the type of company that should run Java. As a result, the Sun acquisition raised alarms in the Java community."[58] He also said that since Oracle split the prearranged improvements to Java SE 7 in two releases, it effectively "made more significant advancements after the Sun acquisition than in the two-and-a-half years prior"[59].

However, there have been some problems concerning Oracle's steering of Java. The project managers of the Apache Lucene search engine have pointed out that the release contained bugs that could have an effect on applications or crash Java virtual machines. Nevertheless, in 2010 the Apache Software Foundation decided to resign from the Java Process steering committee, as a consequence of approving Java SE 7, explaining that the widely-used scripting platform has become a classic proprietary technology fully controlled by Oracle. The problem was that the Java SE 7 voting was a way for Executive Committee to show their will to defend the JCP[60] as

[58] Oracle is "taking good care of Java post-Sun" Ben Woods, August 2012 (www.zdnet.com, accessed on 20.12.2012.)
[59] Ibid.
[60] Java Community Process –"Java is developed under the Java Community Process ("JCP") which is a process for developing and revising Java technology specifications. The JCP was established in 1998 and is a collection of bilateral contracts (Java Specification Participation Agreements ("JSPAs")) between Sun and the different members of the JCP (who may be individuals, corporations or other groups considered as "stakeholders" in relation to the Java development platform) and Sun. It is not an organisation, nor the steward of Java, but rather a participative process to define standards around Java. There are currently over 1 200 members. Currently, important competitors of both Sun and Oracle are represented in the JCP: IBM, SAP AG, Hewlett Packard, Oracle itself, Cisco Systems, Adobe Systems Inc., RedHat, as well as companies like Google, Motorola, Intel (a competitor of Sun for microprocessors) and Philips. " (Case No Comp/m.5529 – Oracle/ Sun Microsystems, para 811.

an open specification process, and to prove that it matters for them to fulfill obligations under JSPA[61].

The objective of JCP is to assure the downstream interoperability, and it is especially so that Java can be used by anyone in order to create wide array of applications that run on multiple platforms. Therefore, while Oracle failed to meet the obligations under the JSPA, it provided the Executive Committee with Java SE 7 specification request and the license consisted of contradictory provisions, In this way, they gravely restricted distribution of the independent specification implementation. The failure to meet the responsibilities under the JSPA refers to the fact that Oracle rejected to vest TCK[62] license for Java SE for the Apache Software Foundation's Harmony project, which was required under the JSPA.

It is important to mention that public promises made to the Java community by the officials of Sun Microsystems (acquired by Oracle) had been initiated by Sun Microsystems itself, before the merger. Hence, as a matter of fact, this is just a continued policy which was accepted by Oracle. Besides, prohibiting the distribution of the independent open source implementations of the specifications is quite disturbing as regards the open source community. Oracle did not address any of these concerns.

Therefore, Apache Software Foundation expected that Executive Committee would protect the rights of the implementers within their ability to do so, and that they would safeguard the integrity of the JCP licensing setup by means of securing that JCP specifications can be freely distributed and implemented. Interestingly enough, many members of the aforementioned Executive Committee publicly announced that limitations as regards the distribution similar to those found in the Java SE 7 license "have no place in the JCP". Two of the members of the Committee have resigned in protest over such a policy. Thus, when the Executive Committee approved Java SE 7, they failed to protect the rights of implementers, and when they accepted Oracle's TCK licensing terms for Java SE 7, they did not prevent shuttering of the JCP licensing structure. We can see once again that one of the major arguments - here the open standard, was intentionally misappropriated by Oracle.

[61] Java Specification Participation Agreement. These agreements were signed between Sun and the various members of JCP (stakeholders in the Java development platform).
[62] TCK - Technology Compatibility Kit ; compatibility tests are necessary to ensure that each implementation is compliant with the specification.

Taking into account that JCP was introduced as an Open Standards Body, it is inconceivable for it to inhibit implementations of the specification. Unsurprisingly, Oracle's promises to have open and transparent Java Specification Requests have been unfulfilled as well. These facts made the open source community members even more reluctant towards Oracle and its policy as regards the open source software and the interoperability. I assume that the Commission was aware of the possibility for this to happen, but it was not sufficiently significant to require imposing stricter limitations. In any event, the leniency of the Commission, whether it originates from the attempt to reach a standard imposed by the US anti-trust law, or from the lack of sense of sensitivity and vulnerability of the open source software (because of its specific nature), this decision resulted in a significant harm as regards the open source software and its numerous community. Although the Commission tried very hard to prove its inclination towards open standards through EIF, protecting the open source software in practice proved to be much more complicated task.

4. Other problems related to the open source software

Apart from the mentioned problems related to the open source software, there are also some other concerns which are not frequently discussed, even though some of them tend to become actual topics in the foreseeable future. In the following part I will discuss the current software protection in EU, and the impact of the possible adoption of the EU Software Patent Directive on the open source software.

4.1. Open source software and patents

While the United States the protection of software define very broadly, European Union is quite careful, and hardly makes the concessions. The question of patent protection for computer software in European Union has been a politically and legally difficult one for a number of years. There are many reasons for this, such as alleged "lack of inventive step", or lack of "technical effect" as its segment, examples of abuse of US software patent system, balance of power between the European Parliament and European Commission, etc. States that are the members of

the European Patent Convention[63] insofar forbid the patenting of computer programs as such. Nevertheless, the present state of affairs provide certain leeway for EU Member States to decide on the matter which is probably overriding reason, as the legal discussions on the issue slowly but steadily move towards the US insight.[64] Taking into account the stance of US law on the matter, it is interesting to mention that in May 2007., Microsoft claimed that free and open source software infringed more than 235 of their patents.

Brad Smith, the general counsel of Microsoft, stated that Linux kernel violated 42 patents belonging to Microsoft. In addition, he said that Linux's user interface and design infringed additional 65 patents. The former Microsoft CEO, Steve Ballmer said that the open source competitors of Microsoft are to play "by the same rules as the rest of the business". Interestingly enough, the action has never been brought before the court and there had been no litigation as regards the matter.

However, even in case this happens, in my opinion it would be extremely difficult for them to prove the patent violation, since the developers within the open source community did not unlawfully acquired the interoperability information, but by means of reverse engineering. Reverse engineering means that the developers were able to take apart pieces of a specific software and then put it back together, in order to understand how it is made, and what are the necessary features that their future product has to have, in order to be compatible with it. This would be similar to a certain production technology, where one company produces output based on the technology it developed, and the team of technicians of the other company take it apart and comprehends what technology has been used to produce it. There is no illegal information obtainment and therefore no legal liability. This is seemingly irrelevant as regards the software

[63] Albania, Austria, Belgium, Bulgaria, Switzerland, Cyprus, Czech Republic, Germany, Denmark, Estonia, Spain, Finland, France, UK, Greece, Croatia, Hungary, Ireland, Iceland, Italy, Liechtenstein, Lithuania, Luxembourg, Latvia, Monaco, Portugal, FYR Macedonia, Malta, Netherlands, Norway, Poland, Portugal, Romania, Serbia, Sweden, Slovakia, San Marino, Turkey. Bosnia and Herzegovina and Montenegro recognise European patents upon request.
[64] "Comparative analysis of the software protection in EU and US", Belma Cabaravdic, Université de Strasbourg, August 2012

protection in Europe, since the EU Software Patent Directive[65] raised fervent discussions and a great deal of ink has been spilled on its account, but the answer was a resounding no.

However, based on the aforementioned case law, we can see that EU is trying to bring closer its legal policy as regards to software, to that of US. Therefore, the adoption and the enforcement of the Directive will not be unexpected. Besides, even though Art. 52 of EPC[66] excludes software from patent protection, even now it is possible to obtain patent protect of the software, if it meets certain requirements. So called "computer-implemented inventions" can be patented within European Patent Office. In this context, "computer-implemented inventions" relates to those implementations that includes the use of a computer, a computer network or an alternative programmable apparatus with one or several features actualized through a computer program. Hence, the patentability cannot be denied in this case only because it includes the computer use, and one can require patent protection if the subject matter of the invention as a unit possess a technical character which is present in all versions covered by the patent request. And how would the potential software patentability in Europe affect the open source software?

If the proprietary or closed source software obtains the patent protection, would the dissemination of the interoperability information necessary to make the open source software interoperable with its proprietary counterpart, represent a violation of the patent rights? And then, how to treat the open source software? The alternative here can be "open patent license", which is still under development. In essence, this license represents "a mutual non-aggression pact", and it would "Allow companies to be mutually non-aggressive with respect to:

• only a specific set of patents,

• all their software patents, or

• all their patents;

• contain language such that submitted patents are available for Open Source/Free Software use; and

[65] EU Directive on Software Patentability or the EU Software Patent Directive was a proposal by the Commission in order to adopt patent protection of the software in EU. It has been rejected several times but it continues to be an actual and controversial topic.
[66] European Patent Convention

• require that companies wishing to obtain the full advantages of the license with respect to software patents and more, not attempt to make an end-run around the license by using forms of IP other than patents that restrict the reimplementations of works. This would include things such as look and feel copyrights, and restrictions on reverse engineering. (The license would be useless if participants could still restrict each other from whole areas of the market using dubious non-patent forms of IP.)"[67]

Another issue we mentioned before, was incompatibility of the FOSS and FRAND terms. Are they really incompatible, or it is just ostensibly so? Some commentators argue that if both FOSS and patent advocates are sufficiently reasonable, the synergy of the two would not be so difficult to achieve. However, if FOSS advocates act in an unreasonable manner than it is easy for them to make the projects incompatible with patents. On the other hand, if the patent rights holders use IP rights for foreclosing the FOSS based competitors, that would generate quite an issue. And in case the patents contribute to a standard on FRAND terms, the situation does not have to necessarily be problematic.

Let us look back at the software patentability in Europe, and what would it mean for open source software. It is possible for the aforementioned Directive to be adopted, but it is uncertain when, and under what conditions. There are several potential consequences of this trait. Firstly, certain software patents risk would always be present and therefore all persons using the software would be potential target of the patent lawsuit. Simple use of the software would constitute a patent law infringement. What is more, introducing patentability into software would mean hindering the development of useful software by means of obstructing compatibility and interoperability. Also, patents are not compatible with software since it is quite complex. Large number of ideas had been used to create a software, and discerning all of them to check against the patents that are already there would be quite difficult, and it would incur significant costs and would be time consuming. In addition, it is important to mention that one of the most important opponents of the Software Patent Directive proposal are the members of the open source community, and the question of this Directive's impact on the open source software is partially answered. Besides, patenting software would represent an issue for companies, irrespective of their size, for individual software developers, users, etc.

[67] http://www.openpatents.org/ (accessed on 30.12.2013.)

For instance, the companies would have to bear the costs of registering patents, negotiating patent crosslicensing, and costs of defense in case of patent claims. Sometimes big companies use software patents as means of preventing new competitors in the market, but they also confront with companies that bring action based on the patent claims against other companies, but never actually develop any software by themselves. The respondent company would be in a very difficult position. As regards the software developers, they would face difficulties as regards the legal uncertainty. A developer while engaged in programming would risk violating the patent law, but he will not know for sure whether or not he violates the patent law. And ultimately, the users would be compelled to pay for everything. There are some estimates that the costs related to smartphones are around 20% of the price the customer paid. Ultimately, we can say that maintaining the current system of software protection under EU law would be favorable towards the open source software, even though there is a certain tendency towards introducing the patent protection of the software.

4.2. Enforceability of the FOSS licenses

We gave a short review of the open source licensing and GPL license, but the question remains: are these licenses actually legally enforceable? This is important because if its specific licenses would not be enforceable in practice, then the open source software would be at disadvantage compared to the proprietary software and its copyrights. Therefore, the open source software would not be able to compete on equal footing with the proprietary software, and users / developers who use open source software would be deprived of the rights conferred by the GPL license. That is, the rights to share, improve and distribute the software. In case of the infringement of GPL license, a specific software usually does not include notice about GPL software being included in the software product.

Therefore, there is no any text referring to the license, or the source code provided whatsoever. In some cases, there is a notice regarding the GPL, but the text is not readable or not available. Sometimes the license notice directs to a website including necessary information, or not even that. Even when the necessary information are provided on a specific website, the District Court of Munich made a decision that a mere link to a website with the necessary information is not

sufficient to conform with section 3 of the GPL terms and conditions. More frequent situation is when firmware updates are available but with no “complete corresponding source code”. Also, sometimes the source code that is available, is not complete since the complete source code would enable installation and compilation of the software. The works derived from GPL'd software have to be GPL'd as well. There was no court cases regarding the derived works of GPL'd software, but it is not excluded to happen in the future. At the moment, there is incertitude concerning the definition of “derivative work”, and how could it be discerned from an independent piece of software.

The FOSS licenses are not tried in court very often. The very first lawsuit concerning the enforceability of GPL license was filed in 2004. A WLAn router manufacturer had GPL'd software included in its firmware and declined to declaration to “cease and desist” from the distribution which was inconsistent with the GPL. The copyright holder asked for a preliminary injunction, which was granted by the District Court of Munich, only a day later. After an objection has been raised by the manufacturer, the Court confirmed the preliminary injunction and delivered a written explanation. The most significant part the judgment is the conlusion that the infringement of GPL has the effect of a copyright infringement, in the context of the automatic termination clause of section 4 of GPLv2.

The GPL is regarded as a license agreement containing the requirements which, if not met, entail reversal of the rights conferred. Consequently, GPL distribution incompatible with the GPL entails also a copyright infringement, not just infringement of a contract. Although there are not many cases on the matter of the GPL enforceability, from this decision we can see that the courts are not lenient towards the infringements of GPL licenses. Taking into account the specific nature of open source software, and even more specific its model of licensing, it gained the due attention and protection. It is fair to mention here Mr. Harald Welte, who founded gpl-violations.org. It is a non-profit project, supported by a large number of the open source software developers. The purpose of this project was raising the awareness about past and present infringements of the GNU GPL. It gathers, maintain and distribute information concerning people and organisations who use and distribute GPL'd free software without enclosing the license terms. Hence, based on these few cases, we could say that those licenses, although specific, seem to be legally enforceable. Although, the German courts appear to be inclined to

protect the open source software by fully enforcing its licenses, we are to see how would other European courts treat these licenses. In the following part, I will explain the relation between the innovations and openness, why is it important to promote the open use of IP rights, and how this ultimately increases the consumer welfare.

4.3. The innovations and "openness"

In the case Microsoft Corp. v. Commission of the European Communities we had the situation of "refusal to grant a licence for the use of a product covered by intellectual property rights". This case shows the extent of a conflict between competition law and IP law, but its decision raise a question: does it point in the direction of "opening" the IP law towards the competition? If the intellectual property rights are created to encourage new innovations and not to block the competitors, maybe "open use" of IP rights, promoted by FOSS licensing could be a new, better way to promote innovation, improve the competition and ultimately, increase the consumer welfare? This would certainly make sense.

However, even though this decision showed that even longtime dominant firm is to be held accountable for refusing to provide certain interoperability information to (former) Sun Microsystems Inc., there has been many other software products that the open source community never managed to obtain the necessary interoperability information (for instance the Microsoft Office), but there has never been any litigations concerning the matter.

There is also an interesting, though unsuccessful case in US, Wallace v. International Business Machines Corp. et al., where the Free Software Foundation was accused of price fixing. In this case, the plaintiff claimed that the fact the Linux was available free of charge, he was not able to make profits by selling his own operating system. He also stated that the requirements under the GPL to provide copies of software licensed under it, and making it freely available and even at no charge is equal to price fixing. The case was dismissed, and the Court found that "The GPL encourages, rather than discourages, free competition and the distribution of computer operating systems, the benefits of which directly pass to consumers. These benefits include lower

prices, better access and more innovation."[68] Although the first case took place in Europe, and the other one in US, and taking into consideration all other facts concerning proprietary and open source software, I think it is fair to conclude that the open source software is ultimately more likely to contribute to the overall consumer welfare by sharing knowledge and innovations. Nevertheless, the consumers benefit from the proprietary software as well, but the cases related to the dominance in field of proprietary software distribution points to pursuing the interests which are not necessarily the ones of the consumers.

Besides, some analysts consider that longtime dominance of Microsoft in field of desktop / PC operating system hinder innovation. The fact that Microsoft is almost absent from the market for mobile and smartphone software, talks about its lack of innovation incentives. Also, many IT experts consider that personal computers would be barely employed about 5-6 years from now, which means if Microsoft does not start innovating and try to keep up with the latest technologies, it could as well be extinct.

4.4. Limited understanding of the open source software on the part of the legal profession

A problem regarding the lack of knowledge on open source software by the lawyers has been emphasized by the open source community every now and then. Legislators not considering all aspects of the open source software in great depth entails their decisions / opinions being based on the incomplete picture. Many of the open source community members and developers complained about this problem, suggesting that it would be more convenient to gather together the experts in field of the open source software, the developers and lawyers who are trying to resolve an issue from a legal standpoint instead of distributing all the work to the lawyers. In other words, the issue of fate of the open source software should be equally distributed among the developers and lawyers. In this way, each group would be able to contribute in the field it is specialized in, so the decisions made would be based on the better information, and would not omit the ostensibly irrelevant facts. For instance, in the Sun Oracle case, there was a warning

[68] Wallace v. FSF

made by the open source community and Mr. Richard Stallman about the anticipated fate of MySQL, which eventually became true. And the open source community

members are embittered because in their view, the lack of deep understanding of the issue from the technical aspect have lead to the result. It seems to me that a higher level of the cooperation with the IT experts when deciding about the similar cases could be a good idea.

Conclusion

Does the open source software require a special treatment from the competition law authorities? Is its integration in the competition law perfectly aligned with its specific nature? The answer to this question is quite complicated, and goes both ways. There has been significant steps taken by the Commission, directed towards approving "openness", and facilitating adoption and use of the open source software. They perceived the importance of interoperability within the European Interoperability framework, when creating (open source) software, and gave it the due attention. However, there are still unresolved issues of "open standards" definition, which calls for further actions in order to specify the requirements, which would enable verification whether the standards meet the conditions or not.

Imposing the remedies which are not legally binding proved to be quite futile as we could see from the case of Sun Oracle merger. The three legally binding remedies were not concerned with the most important part of the Commission decision, and that is the future of the open source version of MySQL database. The other seven remedies which were not legally binding, were violated by Oracle, and naturally the Commission did not have grounds for penalizing the company. Therefore, the whole investigation and over 150 pages of the decision and reasoning was reversed in practice, with no possibility of correcting the mistake. Besides, the issue related to the Java technology was ostensibly resolved, but in effect the resignation of the Apache Software Foundation from the Java Process steering committee, pointed to the opposite direction as regards the downstream interoperability.

Would involving more of the IT experts in the decision making regarding the open source software take into consideration ostensibly irrelevant facts that are in fact important for the open source software? There is only one way to find it out; to include them in the process. However, it is quite doubtful whether the Commission would be inclined to accept the suggestion. As regards the GPL licenses, although there is no much case law on the matter, but the few cases (most of which come from Germany) of the GPL enforcement, showed a satisfactory level of the legal enforcement of the GPL licenses, even though they are rather specific compared to the classic

licenses. Great deal of work has been done, but as we can see, there is also a lot of work to be done in the future.

This is especially so because of the nature of ever changing IT sector, and the fact that the open source software in 1998 is not the same as the open source software in 2005, or 2013 for that matter. The users are also becoming more skilled over time, and there are fewer and fewer IT problems regarding which users rely upon experts. Hence, hopefully the Commission would take into consideration the past mistakes and learn from it, in order to make better and more informed decisions regarding the open source software (and its future) in the future.

Bibliography

Articles

1. Baker J, Controversial European Interoperability Framework Announced <http://www.idgnews.net/>, accessed on 11.12.2013.

2. Bitzer J & P. J. H. Schröder, 'Open Source Software, Competition and Innovation', University of Applied Sciences OOW at Emden, Germany, Aarhus School of Business, University of Aarhus, Denmark, 2007

3. Bridgwater A, April 2012, <http://www.computerweekly.com>, accessed on 24.11.2013.

4. Buhr C.C. , Crome S, Lübbert A, Pozzato V, Simon Y, Thomas R, 'Oracle/Sun Microsystems: The challenge of reviewing a merger involving open source software', Competition Policy Newsletter, No 2 – 2010 (Mergers)

5. Cabaravdic B, 'Comparative analysis of the software protection in EU and US', Université de Strasbourg, 2012

6. Glorioso A, 'An interoperable world: the European Commission vs Microsoft Corporation and the value of open interfaces', Media Innovation Unit – Firenze Tecnologia, April, 2005, Queen's University of Belfast

7. Hillesley R, 'MySQL – 'A controversial dual licensing model', Linux User & Developer Issue 132 (The monthly magazine for the GNU generation)

8. Jaeger T., Enforcement of the GNU GPL in Germany and Europe, 1 (2010) JIPITEC

9. Kahin B, 'Open standards and the royalty problem', Jan 2011

10. Kennedy Dennis M, 'A Primer on Open Source Licensing Legal Issues: Copyright, Copyleft and Copyfuture'

11. Moglen E, General Comments on GPL and Competition, November 2009, Software Freedom Law Center, New York

12. Perrin C, 'Is Oracle poised to effectively end open source software?' <www.techrepublic.com/blog/linux-and-open-source> September 13, 2010, accessed on 17.11.2013.

13. Petrunia S, 'Disappearing test cases or did another part of MySQL just become closed source?' ´https://blog.mariadb.org/disappearing-test-cases/, accessed on 24.12.2013.

14. Sartorio D, 'Open Source Interoperability: It's More than Technology', January 2008 <http://timreview.ca/article/112#sthash.TFkAamtr.dpu>, accessed on 28.12.2013.

15. Stallman R, 'Viewpoint: Why "Open Source" Misses the Point of Free Software' (2009) 52-6 Communications of the ACM 31

16. Stallman R., KEI, ORG, Letter to the EC opposing Oracle's acquisition of MySQL, <http://keionline.org/ec-mysql>, accessed on 3.12.2013.

17. Whish-Bailey, 'Competition Law', Oxford University Press, 2013

18. Wildenius M, 'Oracle Adding Closed Source Extensions to MySQL' (Monty Says, 19 September 2011) <http://monty-says.blogspot.nl/search?q=MySQL, accessed on 18.11.2013.

19. Udenheim T, ´ European Interoperability Framework - a new beginning?´ https://blogs.oracle.com/trond/entry/european_interoperability_fram, accessed on 29.12.2013.

20. Välimäki M, 'A Practical Approach to the Problem of Open Source and Software Patents' (2004) 26-12 E.I.P.R. 523

21. Välimäki M, 'Software Interoperability and Intellectual Property Policy in Europe', Helsinki University of Technology

22. Vaughan-Nichols S, ´ It's not Apache vs. Oracle; it's Oracle vs. open source`http://blogs.computerworld.com/17334/its_not_apache_vs_oracle_its_oracle_vs_open_source, November 2010, accessed on 18.11.2013.

23. Vezzoso S, ´Open Source and Merger Policy: Insights from the EU Oracle/Sun Decision´, University of Trento, 2010

24. Woods B, ´Oracle is taking good care of Java post-Sun´ <http://www.zdnet.com/oracle-is-taking-good-care-of-java-post-sun-7000002459/> accessed on 27.12.2013.)

25. Young R, 'New Commercial Extensions for MySQL Enterprise Edition' (Oracle's MySQL Blog), accessed on 1.12.2013.

Cases

- United Brands Company and United Brands Continentaal BV v Commission of the European Communities

- Case No COMP/M.5529 – ORACLE/ SUN MICROSYSTEMS

- T-201/04 Microsoft Corp. v. Commission of the European Communities

- Wallace v. FSF (US)

- Harald Welte v. FANTEC GmbH

Public documents and legal sources

- Commission Notice on the definition of relevant market for the purposes of Community competition law, Official Journal C 372 , 09/12/1997 P. 0005 – 0013

- Council Directive 91/250/EEC

- Decision No 922/2009/EC of the European Parliament and of the Council of 16 September 2009 on interoperability solutions for European public administrations (ISA) OJ L 260

- European Interoperability Framework

- MEMO/10/689 "Commission adopts Interoperability Strategy and Framework for public services - frequently asked questions", Brussels, 2010

- Oracle press release, ´ Oracle Makes Commitments to Customers, Developers and Users of MySQL´, December 2009

- Oracle White Paper, 'The Department of Defense (DoD) and Open Source Software', Oracle Corporation, Baum D et al., September 2013

- Statement of Objections from the European Commission regarding Sun Oracle merger

- Statement of Oracle Corporation <http://www.oracle.com/us/corporate/press/039824>, accessed on 17.11.2013.

- The Commission Communication, 'Towards Interoperability for European Public Services' COM (2010) 744 final

Additional web sources used for the statistical data and certain definitions

<http://perens.com/OpenStandards/Definition.html>, accessed on 29.12.2013.

<http://www.smartgrids.eu/CEN-CENELEC-ETSI>, accessed on 26.12.2013.

<http://www.openpatents.org/>, accessed on 27.12.2013.

<www.netmarketshare.com>, accessed on 18.1.2013.

<www.princeton.edu>, accessed on 20.11.2013.

<http://www.techterms.com>, accessed on 22.22.2013.

<http://www.gnu.org/copyleft/gpl.html>, accessed on 10.11.2013.

<http://fsfe.org/activities/os/def.en.html>, accessed on 18.12.2013.

<http://www.gnu.org/licenses/>, accessed on 12.12.2013.